



HAL
open science

Evaluation des simulations de feux de forêts

Bahaa Nader Hannah Milad

► **To cite this version:**

Bahaa Nader Hannah Milad. Evaluation des simulations de feux de forêts. Modélisation et simulation. Université Pascal Paoli, 2015. Français. NNT : 2015CORT0005 . tel-01403922

HAL Id: tel-01403922

<https://theses.hal.science/tel-01403922>

Submitted on 28 Nov 2016

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

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



UNIVERSITÉ DE CORSE-PASCAL PAOLI
ÉCOLE DOCTORALE ENVIRONNEMENT ET SOCIÉTÉ



**Projet de thèse présenté pour l'obtention du grade de
DOCTEUR EN INFORMATIQUE**

**Soutenance publique par
Bahaa NADER**

Le 19 Juin 2015 à 14h, Amphithéâtre, faculté des sciences

Évaluation des Simulations de Feux de Forêt

Directeurs :

M. Paul-Antoine BISGAMBIGLIA, Professeur, Università di Corsica
M. Jean-Baptiste FILIPPI, CR, CNRS / Università di Corsica

Rapporteurs :

M. Orhan FEYZIOGLU, Professor / Galatasaray University
M. Alexander FILKOV, Associate Professor / Tomsk State University

Jury :

M. Dominique LAMBERT, Professeur / Université de Toulouse
M. Orhan FEYZIOGLU, Professor / Galatasaray University
M. Christophe PAOLI, Professor / Galatasaray University
M. Alexander FILKOV, Associate Professor / Tomsk State University
M. Paul-Antoine SANTONI, Professeur, Università di Corsica
M. Paul-Antoine BISGAMBIGLIA, Professeur, Università di Corsica
M. Jean-Baptiste FILIPPI, CR, CNRS / Università di Corsica

A JC,

A mes parents,

RESUME

L'évaluation de la performance de modèles est une étape fondamentale de leur développement et de leur amélioration. Le travail de recherche présenté dans ce manuscrit est consacré à l'évaluation des modèles de simulation de propagation des incendies des forêts. Une revue des travaux a montré que, si de nombreux éléments étaient disponibles, aucune solution normalisée et automatisable n'était proposée dans ce champ applicatif. Une solution à ce problème est proposée en déclinant une approche formelle développée dans le cadre de la théorie de la modélisation et de la simulation. Cette étape a permis de déterminer conceptuellement quels composants devaient être développés et comment les interconnecter.

La réalisation de ce cadre a requis premièrement la normalisation de données disponibles pour les incendies de forêts, aucun standard de fichier ni même de nomenclature n'étant disponible et/ou utilisé par les modélisateurs ou ingénieurs (observations ou simulation). Un ensemble de noms, notations et formats d'encodage des données dans un conteneur NetCDF a pour cela été proposé.

Une seconde étape a consisté à déterminer les métriques nécessaires pour quantifier les erreurs de simulation (score de simulation). Si quatre méthodes standard ont pu être identifiées dans la littérature, nous avons pu montrer qu'elles se limitaient à la comparaison à un instant donné, ne pouvant donc rendre compte de la performance de la dynamique d'une simulation incendie. Cette problématique a été traitée en proposant deux nouvelles méthodes de calcul de score spécifiques. Ces différentes méthodes d'évaluation ont été implantées au sein d'une bibliothèque de calcul.

Enfin, la réalisation d'une évaluation de modèles a été réalisée à l'aide d'une implantation du cadre défini précédemment. Cette évaluation a consisté à confronter quatre formulations de modèles de vitesse de front de flammes effectuées sur 80 simulations

d'incendies réels de manière complètement automatique. L'automatisation et le non-ajustement de paramètres ont ainsi permis de se rapprocher au plus près du contexte opérationnel où peu d'informations locales sont disponibles, peu de temps après l'alerte de l'éclosion d'un incendie. Les résultats ont démontré que cette approche est de nature à laisser apparaître une hiérarchie des performances de paramétrisation ou de formulation relativement à une autre, sans toutefois être en mesure de donner une mesure absolue et objective de l'erreur modèle.

MOTS CLES

Évaluation – Modèles de simulation de feux de forêt – Cadre expérimental – Métriques – Erreurs de simulation.

ABSTRACT

Performance evaluation of models is a fundamental step towards an efficient development and improvement. The research work presented in this manuscript is devoted to the evaluation of forest fire simulation models. Review of current work showed that if many elements were available, there were not any standardized and automated solutions proposed in this field. A solution for this problem is thus proposed, built upon a formal approach from the theory of modelling and simulation. This formal framework allowed to identify conceptually which components should be developed and how they would be interconnected.

Realization of this frame required to start with a normalization of available wildfires data, as no standardized file or even nomenclature were available and/or used by all modellers and engineers (observations or simulation). A set of standard notation and name as well as a standard encoding data format in a scientific container NetCDF is proposed along with associated software.

A second step is devoted to the identification of the scoring methods required to quantify the simulation error. If four standard methods have been defined in the literature, we have shown that these methods were limited for the comparison at specific time, not reporting clearly the performance of the simulation dynamics. This issue has been solved by proposing two new specific score calculation. These different evaluations methods are implanted in an open source computation library.

Eventually, the realization of models evaluation was performed using an implementation of the proposed experimental frame. This evaluation consisted to confront four formulations of flame front velocity models on 80 real wildfire simulations in a fully automatic way. Because it was automatic, it implied that no parameters adjustments could be performed by an operator after

the fire was observed; being more representative of an operational context with little information available immediately after a fire has been reported.

The results have shown that this approach, while unable to provide an absolute measure of the model error is capable to reveal a hierarchy of performances between parameterizations or formulations.

KEYWORDS

Evaluation – Forest fire Simulation Models – Experimental frame – Metrics – Simulation errors.

ACRONYMES

CE	Cadre Expérimental
DEVS	Discrete Event System Specification
EF	Experimental Frame
ONF	Office National des Forêts
ANR	Agence Nationale de la Recherche
SPE	Sciences Pour l'Environnement de l'université de Corse - UMR 634
CERFACS	Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique
M2P2	Laboratoire d'énergétique Moléculaire et Macroscopique, Combustion de l'École Centrale de Paris - UPR288
EM2C	Laboratoire de Mécanique, Modélisation et Procédés Propres de l'Université Paul Cézanne - Aix Marseille III - UMR-68
INRIA	Institut National de Recherche en Informatique et en Automatique
CNRM	Centre National de Recherches Météorologiques
ETLA	Laboratoire d'Aérodynamique - UMR 5560
SIG	Format de données SIG est une représentation informatique des données géographiques
SDIS 2A	Service Départemental d'Incendie et de Secours Corse du sud
SDIS 2B	Service Départemental d'Incendie et de Secours Haute Corse
DDAF 2A	Direction Départementale de l'Agriculture et de la Forêt Corse du Sud
INSEE	Institut National de la Statistique et des Études Économiques
DFCI	Défense de la Forêt Contre l'Incendie
TIFF	Tagged Image File Format

REMERCIEMENTS

Les travaux présentés dans ce mémoire de synthèse ont été réalisés au sein de l'équipe *Feux de Forêts* de l'Université de Corse. Cette équipe est intégrée au laboratoire Sciences Pour l'Environnement (SPE), Unité Mixte de Recherche CNRS 6134.

J'adresse également mes remerciements aux institutions qui ont soutenu techniquement et financièrement cette étude : *l'Université de Corse* et la *Collectivité Territoriale de Corse*.

Je tiens tout particulièrement à exprimer ma gratitude à *Monsieur Paul Antoine BISGAMBIGLIA*, Professeur de l'Université de Corse, pour avoir accepté de diriger ces travaux, pour sa disponibilité, son patience et ses conseils qui m'ont été indispensables tout au long de ces années. Ce manuscrit ne pourrait pas être apparu sur cette forme ni scientifique ni orthographe sans ses corrections qui étaient plus que des corrections et des remarques.

Je tiens également à remercier *Monsieur Jean-Baptiste FILIPPI*, Chargé de Recherche au CNRS, pour avoir co-encadré ces travaux depuis mon stage en Master 2. Il était très patient avec moi ainsi il était une grande source d'inspiration non seulement dans mon travail, mais aussi dans ma vie. Ma thèse est inspirée de ses travaux, ses idées, l'ensemble des discussions que nous avons eu et les remarques qu'il m'a conseillées. Grâce à ses corrections, nous avons pu présenter deux publications dans les premiers forums internationaux de simulation (The Winter Simulation Conference – Etats Unies 2011, The Summer Simulation Conference – Canada 2010) ; ainsi grâce à ses travaux principaux avec *Monsieur Vivien MALLET*, nous avons pu publier deux articles dans des journaux internationaux (The International Journal of Wildland Fire – 2013, Natural Hazards and Earth System Sciences Discussions – 2014). Je lui remercie pour son

disponibilité afin de m'expliquer, de me rappeler et de me corriger les notions scientifiques. Sans lui, cette thèse ne serait jamais terminée ni cette mémoire a été rédigée.

Je souhaite exprimer toute ma gratitude à **Monsieur Vivien MALLET**, Chercheur scientifique (INRIA) que je ne saurais comment décrire sa participation à ce travail tant ses expertises dans les Mathématiques et Informatique Appliqués. Je remercie également **Madame Hajer AYED**, Stagiaire (INRIA) pour sa participation avec son excellente étude sur le simulateur ForeFire.

Je remercie les rapporteurs, **Monsieur Orhan FEYZIOGLU**, Professeur de l'Université de Galatasaray et **Monsieur Alexander FILKOV**, Professeur adjoint de l'Université de Tomsk State, ainsi que les autres membres du jury, dont **Monsieur Dominique LAMBERT**, Professeur de l'Université de Toulouse, **Monsieur Christophe PAOLI**, Professeur de l'Université de Galatasaray, Professeur de l'Université de Galatasaray et **Monsieur Paul-Antoine SANTONI**, Professeur de l'Université de Corse, pour avoir évalué mon travail.

Je voudrais également exprimer toute ma reconnaissance à **Madame Evelyne VITTORI**, Maître de conférences à l'Université de Corse, pour sa grande gentillesse et pour sa disponibilité.

Je remercie enfin **Monsieur Frédéric BOSSEUR**, Ingénieur de Recherche CNRS, pour sa participation à ce travail en offrant les données sources des feux de forêts d'ONF.

Ma gratitude éternelle va à **mes parents** et **ma sœur** pour leur soutien et leur capacité à supporter sans faille durant toutes ces années d'études.

Je tiens également à remercier chaleureusement la famille **BOGDAN**, **Joëlle**, **Christiane** et **Roger** pour leurs encouragements, leur soutien et leurs aides qui m'ont été indispensables tout au long de ces années.

Je remercie tous les doctorants, chercheurs et personnels de l'Université de Corse pour leur grande gentillesse et leur contribution à l'élaboration de ce travail, avec une pensée particulière et surtout à *Marie GARRIDO* et *Souhila SEHILI* pour leur soutien et leurs aides, *Lea LECOMTE* et *Candice OBRON-VATTAIRE* pour leurs corrections de mes rapports, également à *Olivia BASSILY* et *Vladimir VORONITSKI* pour leurs corrections des discours de ma soutenance orale aussi *Samuel TOMA*, *Gauthier LAPA*, *Tom TOULOUSE* et *Jean-José FILIPPI*.

Je tiens à remercier mon ex-femme *Marina* pour son soutien.

Finalement, un grand merci à tous mes amis et surtout *Toni*, *Fadi*, *Amir* et *Amir* qui m'ont été d'un soutien indispensable dans des moments quelquefois difficiles.

TABLE DES MATIERES

Introduction Générale	1
1. <i>Cadre de la thèse</i>	2
2. <i>La problématique</i>	3
3. <i>Cadre expérimental</i>	5
4. <i>Évaluation des modèles</i>	6
5. <i>Plan de la thèse</i>	8
Chapitre 1 État De L’art	10
1. <i>Modélisation et Simulation</i>	11
1.1. <i>Modélisation à événements discrets</i>	12
1.1.1. <i>Modèles atomiques</i>	12
1.1.2. <i>Modèles couplés</i>	14
1.2. <i>Simulation à événements discrets</i>	16
2. <i>Cadre expérimental</i>	19
2.1. <i>Développement d’un cadre expérimental</i>	20
2.2. <i>Application dans le cas des feux de forêt</i>	22
2.3. <i>Mise en œuvre</i>	24
3. <i>Outils utilisés</i>	27
3.1. <i>Le système source</i>	27

3.2. Visualisation des résultats.....	29
4. Représentation des feux.....	30
4.1. Les observations de terrain	30
4.1.1. Représentation graphique.....	31
4.1.2. Présentation textuelle.....	33
4.1.3. Présentation spatiale normalisée.....	33
4.2. Le résultat des simulations	35
4.2.1. Farsite	35
4.2.1. ForeFire	37
5. Méthodes de calcul de similarité.....	39
5.1. L'indice de similarité de Sorensen	40
5.2. Le coefficient de similarité de Jaccard.....	40
5.3. L'indice de Kappa de Cohen	41
5.4. Rapport des aires et rapport des périmètres	43
6. Conclusion.....	44
Chapitre 2 Calcul d'erreurs De Simulations	45
1. Calcul des erreurs	46
2. Développement des scénarios idéaux.....	48
2.1. Formes régulières.....	49
2.2. Formes irrégulières	49
3. Dynamique des phénomènes	52

3.1. <i>L'Accord de Temps d'Arrivée (ATA)</i>	53
3.2. <i>L'Accord des Formes (SA)</i>	54
4. <i>Publication</i>	54
5. <i>Conclusion</i>	67
Chapitre 3 Architecture Logicielle	68
1. <i>Formalisation du cadre expérimental proposé</i>	68
1.1. <i>Représentation du CE suivant l'approche DEVS</i>	70
1.1.1. <i>Générateur</i>	71
1.1.2. <i>Accepteur</i>	72
1.1.3. <i>Transducteur</i>	74
1.2. <i>Les conditions d'arrêt des simulations</i>	75
2. <i>Cas d'utilisation</i>	75
3. <i>Base de données des feux observés</i>	77
3.1. <i>Lexique (Glossary)</i>	78
3.2. <i>La configuration du conteneur NetCDF</i>	79
3.3. <i>Les traitements et les filtres</i>	82
3.4. <i>API</i>	87
3.5. <i>Publication</i>	88
4. <i>La base de simulations</i>	102
4.1. <i>Algorithme de la simulation systématique</i>	102

4.1.1. Côté générateur du CE	102
4.1.2. Côté accepteur du CE.....	104
4.1.3. Côté transducteur du CE	104
4.2. Le champ « vent »	104
4.3. La simulation systématique	105
5. Conclusion.....	106
Chapitre 4	107
Résultats D'évaluation Des Modèles	107
1. Publication	108
Conclusion Générale.....	124
<i>Perspectives</i>	126
Annexe.....	129
1. Formats de fichiers pour la simulation.....	135

TABLE DES FIGURES ET DES TABLEAUX

Figure 1. Cadre générale de la thèse	6
Figure 2. Évaluation des modèles	7
Figure 1.1. Représentation d'un modèle atomique	13
Figure 1.2. Représentation d'un modèle couplé	15
Figure 1.3. Cadre expérimental dans le processus de modélisation/simulation	20
Figure 1.4. Démarche de développement d'un cadre expérimental.....	21
Figure 1.5. Cadre expérimental.....	25
Figure 1.6. Structure du fichier KML.....	30
Figure 1.7. Exemple de représentation correspondant au contour d'un feu	30
Figure 1.8. Représentation graphique du feu d'Aullène à l'été 2009	32
Figure 1.9. Représentation du feu de l'Ortolo à l'été 2009	32
Figure 1.10. Représentation spatiale normalisée du feu d'Aullène à l'été 2009	34
Figure 1.11. Résultat d'une simulation avec le logiciel Farsite.....	38
Figure 1.12. Observation et simulation avec ForeFire [Ayad H., 2011]	39
Tableau 1.1. : Interprétation des valeurs de Kappa.....	43
Figure 2.1. Représentation matricielle d'un feu.....	46
Figure 2.2. Représentation dynamique des feux.....	47
Figure 2.3. Processus de calcul des scores.....	48
Tableau 2.1. Scores d'erreurs pour les formes régulières.....	50
Tableau 2.2. Scores d'erreurs pour les formes irrégulières	51
Figure 3. 1. Cadre expérimental : cheminement des données.....	70

Figure 3.2. Diagramme final des cas d'utilisation	76
Tableau 3.1. Lexique des données des feux de forêt	80
Tableau 3.2. Les dimensions qui forment le cadre expérimental	81
Tableau 3.3. Les variables du cadre expérimental	82
Figure 3.3. Le cas Porto-Vecchio	84
Figure 3.4. Le cas Santo-Pietro-di-Tenda.....	85
Figure 3.5. Le cas Corte.....	85
Figure 3.6. Le cas Propriano avant la modification	86
Figure 3.7. Le cas Propriano après la modification	87
Figure 3. 8. L'algorithme de la construction de la base de simulations.....	103
Figure 3. Capture d'écran des fichiers qui sont créés pour ForeFire	139

INTRODUCTION GENERALE

La compréhension d'un système naturel passe généralement par notre capacité à pouvoir recréer son comportement à travers des techniques de modélisation et de simulation. La plupart des systèmes naturels sont complexes, c'est-à-dire composés d'un ensemble de sous-systèmes qui possèdent une dynamique propre et dont le résultat final n'est pas prédictible au premier abord [Fishwick, 1995]. Les démarches de modélisation et de simulation sont généralement les techniques utilisées pour analyser, comprendre ou prévoir le comportement de tels systèmes. La modélisation consiste à identifier les différents sous-systèmes, à en donner une représentation plus ou moins fidèle et à décrire leurs interactions. La simulation est un processus qui réalise la communication d'information, et l'évolution dans le temps du système dans un environnement donné.

Les systèmes à interfaces évolutives (propagation de feu de forêt ou de polluant en milieu marin...) peuvent être considérés comme étant de tels systèmes naturels. Du fait de leur complexité et de la taille des domaines sur lesquels ils évoluent, il est improbable qu'une personne seule puisse traiter l'ensemble des problèmes et exploiter les résultats. Leur résolution nécessite une étroite collaboration entre des chercheurs en informatique pour la simulation et des spécialistes des domaines étudiés pour la modélisation. Si l'on souhaite étudier ou évaluer des modèles développés dans ce contexte, le travail consiste à standardiser les communications entre les simulateurs et à définir une architecture logicielle permettant d'interconnecter à moindre coûts des modèles hétérogènes existants pour résoudre un problème plus complexe.

La validation du processus de modélisation consiste à comparer les résultats de la simulation avec les données expérimentales ou avec les observations de terrain. L'un des

éléments de base de n'importe quelle approche de modélisation consiste donc à définir un cadre expérimental permettant de comparer facilement les observations avec les résultats de la simulation et de donner du sens à ces résultats d'analyse.

1. Cadre de la thèse

Le travail présenté dans ce document a été réalisé au sein du laboratoire **CNRS SPE - UMR 6134** de l'université de Corse, et plus particulièrement dans l'équipe « **Feux de forêt** ». Cette équipe étudie ces systèmes selon deux points de vue : la modélisation et l'expérimentation des feux de végétation. Ces travaux reposent sur le développement théorique et pratique d'outils de lutte et de prévision des incendies en zone méditerranéenne.

L'objectif principal de ce projet est de proposer un ensemble de méthodes à l'attention des professionnels du feu permettant de modéliser, de simuler et de quantifier le risque incendie. Les activités menées concernent aussi bien la problématique internationale du risque incendie que les enjeux d'aménagement du territoire de la région Corse.

Notre étude s'inscrit dans la partie expérimentation numérique. Nous nous sommes intéressés plus particulièrement à l'évaluation des modèles et à l'outil de simulation de propagation d'incendies de forêt développé au sein du laboratoire. Les problématiques soulevées par les modèles de simulation sont nombreux. L'une d'entre elles concerne la comparaison des simulations avec les données relevées sur le terrain par les professionnels du feu. Nous développons un cadre expérimental pour comparer et évaluer les feux simulés par rapport à ceux qui ont été observés selon des méthodes d'erreurs. Nous proposons également un format de données commun pour décrire les feux ainsi qu'une base d'observations comprenant un grand nombre de cas à simuler.

Une partie des travaux réalisés dans cette thèse ont été développés au sein du projet IDEA (ANR - 09-COSI-006-01). L'objectif final de cette étude est donc de fournir un outil d'évaluation des modèles simulés afin de les améliorer, et de définir un cadre logiciel pour la simulation et l'évaluation de feux de forêt à l'échelle d'une vallée.

2. La problématique

Chaque année, les incendies ravagent des milliers d'hectares de végétation, causant aussi ponctuellement des dégâts humains importants. Le développement de modèles de simulation de ces incendies est destiné à prévoir ces phénomènes, pour pouvoir mieux les maîtriser, mitiger et mesurer leurs conséquences éventuelles.

Pour cela, de nombreuses approches ont été, et continuent à être développées, avec deux objectifs principaux. Le premier, scientifique, est d'améliorer la connaissance du phénomène et les processus physiques. Le second, plus opérationnel, veut appliquer les connaissances obtenues afin de proposer un outil opérationnel de prévision de comportement du feu. C'est dans ce second cas d'application que l'évaluation de modèles est d'une importance capitale, car elle permet de déterminer si une évolution, une paramétrisation ou un jeu de données est en mesure d'offrir une amélioration des performances de l'outil. Dans ce contexte, cette évaluation doit être réalisée de manière objective, sans opérateur humain pouvant biaiser les résultats, requérant des formats de données bien définis pour un processus complètement automatisable.

Pour les feux observés, la plupart des informations disponibles existent sous forme textuelle. Sur le même territoire, deux professionnels (par exemple un sapeur-pompier et un modélisateur) peuvent enregistrer les données du même feu sous différentes formes. Par conséquent, les formats obtenus sont tous différents, très hétérogènes. De même, les informations

contenues ne sont pas compatibles. Aucune ontologie ni format de fichier n'existe pour décrire les données d'observation des incendies. Cette façon de présenter les données est très souple, mais elle n'est pas pratique pour le traitement automatique, ce qui rend ce format inexploitable dans un contexte de comparaison des feux réels/simulés. De plus, il faut souvent analyser plusieurs fichiers différents pour obtenir toutes les informations utiles à l'exploitation d'un feu observé.

Pour les feux simulés, chaque logiciel utilise son propre format de sauvegarde des résultats obtenus. L'exploitation des résultats dépend des logiciels de simulation. De plus, il n'y a aucune compatibilité entre les formats des feux observés et des feux simulés, ce qui rend difficiles les échanges de données entre les experts du domaine et ne permet pas d'obtenir une méthode générique pour évaluer les logiciels de simulation.

Nous devons également définir un Cadre Expérimental (CE) pour valider l'approche de simulation proposée sous certaines conditions et certaines contraintes. Le cadre expérimental permet de générer les données initiales nécessaires pour lancer une simulation. Il gère également le processus de simulation et contrôle les conditions d'arrêt. Enfin, ce cadre est responsable de l'analyse et de la sauvegarde des résultats obtenus lors d'une simulation sous un même format.

Une fois le cadre expérimental modélisé et le format des données défini, nous proposons des méthodes de calcul d'erreurs pour comparer les données sauvegardées. Ces méthodes seront appliquées sur les résultats de simulations et serviront de conditions d'arrêts. Grâce à ces méthodes, nous pourrions en déduire, pour chaque modèle de simulation, les points forts et les points faibles et procéder à des améliorations éventuelles.

3. Cadre expérimental

La première partie de notre travail a consisté à récupérer des données sur les feux observés à partir des bases Prométhée et de l'ONF. Prométhée est une banque de données sur les incendies de forêt en région méditerranéenne et en France, publiée sur Internet (www.Promethee.com). Ces données ont été analysées et mises en forme afin d'être utilisées et comparées avec les résultats des simulations. À partir des données observées, nous sommes en mesure de générer les conditions initiales et d'effectuer plusieurs jeux de simulation en utilisant le logiciel développé au laboratoire. Les résultats obtenus peuvent ensuite être comparés avec les données observées afin d'estimer les erreurs commises. Nous avons défini un cadre expérimental (Figure 1) nous permettant de comparer les résultats de la simulation avec les données observées sur le terrain.

Le cadre expérimental permet la spécification des conditions (scénarios) dans lesquelles le système (le simulateur des feux de forêt) évoluera. Un ensemble de scénarios d'expérimentation basés sur la perturbation de paramètres d'entrée a été appliqué aux modèles pour les tester. Nous avons défini un format standard pour tous les fichiers manipulés par l'application, qu'ils proviennent de la simulation ou de l'observation. Un parseur permet de transformer les jeux de données des feux observés afin qu'ils soient compatibles et utilisables au sein de notre cadre expérimental. Les résultats obtenus sont, quant à eux, sauvegardés au même format, ce qui permet entre autres de pouvoir les comparer avec les feux observés, et de les réutiliser comme conditions initiales pour d'autres simulations. Il est également possible de lancer un ensemble de simulations systématiques et de construire une base de données facilement exploitable.

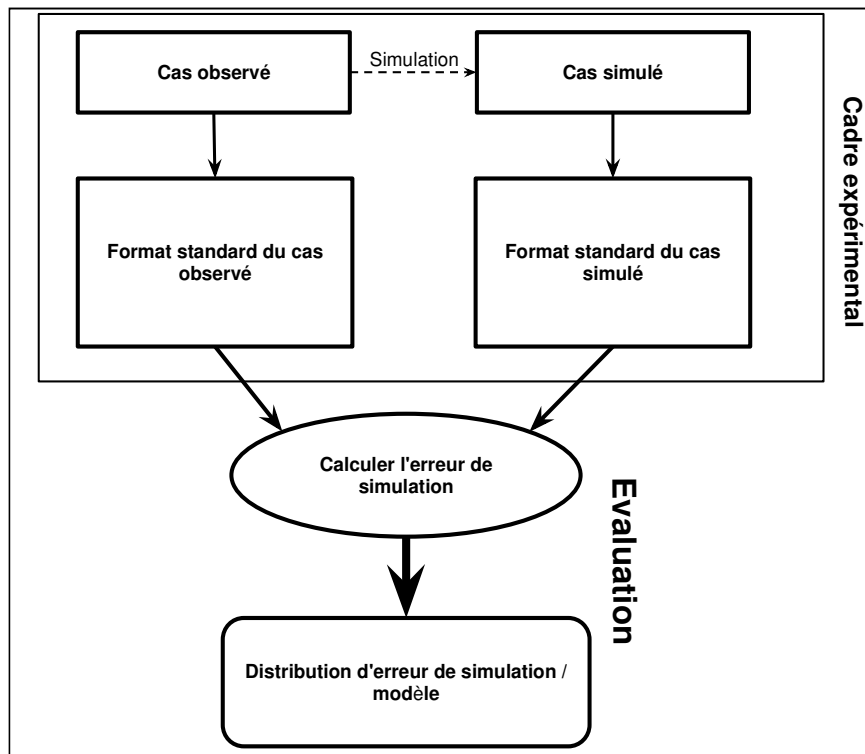


Figure 1. Cadre générale de la thèse

4. *Évaluation des modèles*

Il est possible de contrôler la qualité des modèles au sein du cadre expérimental. Nous avons choisi de tester les simulations en fonction de deux conditions d'arrêt, à savoir l'égalité entre les surfaces totales brûlées issues des fichiers d'observation et de simulation et l'égalité entre le temps calculé et le temps mesuré. Afin de mesurer la fiabilité d'un modèle, nous proposons plusieurs méthodes de calculs d'erreurs issues du domaine des statistiques. Ces méthodes sont utilisées pour comparer la similarité et la diversité entre les échantillons.

Ces méthodes permettent d'obtenir un « score », c'est-à-dire un pourcentage de similarité. Plus ce pourcentage est élevé, plus nous avons une bonne similarité entre les modèles de simulation et les données réelles, plus cette valeur tend vers 0 (celle-ci pouvant être négative

pour certaines métriques) et plus la différence entre les résultats de la simulation et les données réelles sont mauvaises, ce qui implique que les modèles doivent être améliorés.

La Figure 2 détaille les étapes de l'évaluation des modèles. Une base de N échantillons est construite à partir des observations et de données collectées par les professionnels. Cette base est mise au format standard proposé. Il est ensuite possible d'extraire les conditions initiales des feux observés pour pouvoir générer les simulations correspondantes, qui sont ensuite incorporées à la base de données. Il est également possible de modifier les conditions initiales en adaptant ou en modifiant certains paramètres, ou de régénérer de nouvelles conditions initiales. Ce processus itératif permet d'agrandir la base de données et d'avoir, pour un même feu observé, plusieurs simulations.

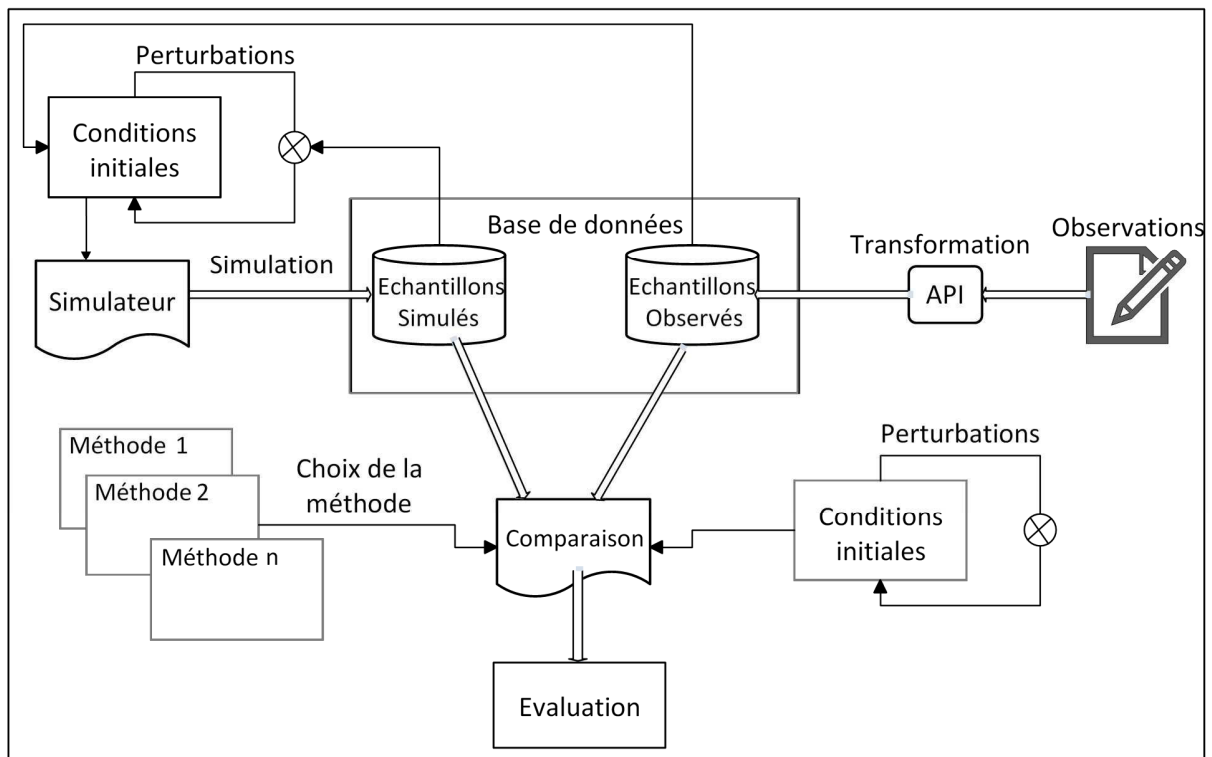


Figure 2. Évaluation des modèles

La comparaison des fichiers d'observation et de simulation permet de mesurer la qualité d'un simulateur. Nous utilisons pour cela des méthodes de concordance pour calculer l'erreur du couple observation/simulation.

S'il est possible d'utiliser n'importe quel simulateur, notamment Farsite [Finney, 2004] ou ForeFire [Filippi et al., 2010], pour tester les méthodes de concordance, cela nécessite d'avoir une connaissance fine de ces simulateurs. En effet, comme ces simulateurs ne proposent pas d'ontologie pour la sauvegarde des résultats, il est nécessaire de post-traiter les données pour les enregistrer au format que nous proposons. Il en est de même pour les données d'entrée.

Dans le cadre de cette thèse, nous utiliserons le logiciel ForeFire développé au sein de notre laboratoire. ForeFire est un logiciel de propagation d'incendie utilisant un modèle de propagation simplifié [Filippi et al., 2010] développé dans le laboratoire du CNRS UMR 6134.

5. Plan de la thèse

Le manuscrit de cette thèse est divisé en quatre chapitres.

Nous commençons notre exposé par une introduction sur les concepts généraux de la méthodologie de modélisation et de simulation utilisée. Parmi ces concepts clés, nous détaillerons plus particulièrement ceux de : système, modèle, simulation et validation. Nous présentons plus particulièrement le formalisme de modélisation et simulation à événements discrets DEVS [Zeigler, 1976] qui est une spécification pour le développement et le couplage de modèles hétérogènes. Nous porterons également notre attention sur les étapes de la construction d'un cadre expérimental afin d'examiner un modèle conceptuel appliqué, en montrant sa formalisation et celle des entités qui le compose. La troisième partie est consacrée aux environnements techniques utilisés. Nous présentons tout d'abord le conteneur de données scientifiques NetCDF,

la base de données des feux de Méditerranée Prométhée qui contient les fichiers d'observation, et enfin le format du fichier KML pour la visualisation. Nous détaillons ensuite comment sont enregistrés les feux observés qui ne sont pas dans la base Prométhée ainsi que les différents formats utilisés dans les simulateurs de propagation des incendies. Le chapitre se termine par l'incertitude et l'étude bibliographique des méthodes statistiques pour calculer les similarités entre les échantillons.

Le second chapitre décrit quatre méthodes de calcul d'erreurs utilisées dans le cadre de cette thèse. Ces méthodes dépendent seulement du contour final du feu et non pas de sa dynamique. Nous avons pu constater que les méthodes classiques de calcul des erreurs n'étaient pas suffisantes dans le cas de la propagation des incendies et pouvaient aboutir à de mauvais résultats. Nous proposons deux nouvelles méthodes plus adaptées aux feux de forêt qui prennent en compte le temps d'arrivée et l'incohérence des formes.

Dans le troisième chapitre, nous présentons tout ce qui concerne le cadre expérimental des feux de forêt proposé. Dans une première partie, nous décrivons les spécifications du cadre expérimental, les données d'entrée/sortie, les scénarios et les conditions d'arrêt de l'expérimentation. Nous détaillons ensuite la structure de la base des données des feux de forêt observés qui est le support principal de cette étude. Cette construction se déroule en trois phases : la collecte des données du domaine et la construction du lexique afin de définir les formats d'enregistrement des données. Nous présentons ensuite comment les données collectées sont mises en forme pour être configurées au format NetCDF. Pour cela, nous avons développé une API en Python « Fire2NetCDF ».

À la fin de ces quatre chapitres, la conclusion générale discutera des points abordés lors de cette étude et détaillera les perspectives et les démarches d'amélioration.

CHAPITRE 1

ÉTAT DE L'ART

Ce chapitre présente l'état de l'art et l'étude bibliographique qui nous ont aidés à réaliser le travail dans le cadre de cette thèse. La première partie est une introduction générale à la théorie de la Modélisation et de la Simulation dans laquelle nous présentons, plus particulièrement, le formalisme de modélisation à événements discrets DEVS. Nous abordons ensuite la notion de cadre expérimental ainsi que le processus de construction associé. À la fin de cette première partie, nous montrons comment réaliser une application spécifique appliquée à la simulation des feux de forêt.

La seconde partie est consacrée aux différents formats de fichiers proposés dans cette thèse pour enregistrer ou sauvegarder les données observées et simulées. Très utilisé dans le cadre d'un usage scientifique, le format NetCDF, qui permet de stocker de manière structurée des données volumineuses et hétérogènes, a été choisi et est présenté au début de cette partie. Nous décrivons ensuite la base de données Prométhée qui sert de support pour les observations et permet d'obtenir les conditions initiales injectées dans le simulateur. En plus du simulateur, nous proposons également un outil pour le stockage des caractéristiques géographiques permettant un affichage sur des plateformes telles que Google Earth ou Google Maps.

La troisième section est dédiée au phénomène des feux de forêt. Elle commence par les différents types de présentation des incendies existants. De même, cette section détaille les formats des fichiers qui sont utilisés par deux des outils de simulation que nous avons utilisés dans le cas de cette thèse, ainsi que le format des fichiers de sortie qu'ils produisent.

Enfin, la dernière section de cet état de l'art concerne les méthodes de calculs d'erreurs ainsi qu'une analyse critique de ces méthodes classiques.

1. Modélisation et Simulation

L'étude des systèmes complexes et la prédiction de leur comportement peuvent être facilitées en utilisant des formalismes de modélisation et de simulation adaptés, combinés à la performance des ordinateurs.

La modélisation est un processus d'identification d'un système et d'interprétation des phénomènes afin de concevoir des modèles correspondants. La simulation est un processus de développement d'algorithmes informatiques en vue d'exécuter les modèles. Dans la pratique, les modèles sont associés à une structure de contrôle appelée simulateur, de manière à produire un comportement possible du système sous certaines conditions et de générer des résultats. Les résultats obtenus sont ensuite comparés aux résultats observés sur le système afin de pouvoir être validés.

La réalisation d'une simulation dans un domaine demande de nombreuses connaissances et compétences, souvent bien éloignées de celles d'un des spécialistes du domaine. Pour aider les scientifiques, il est peut être souhaitable de séparer les processus de modélisation et de simulation. La simulation est de la responsabilité des informaticiens, le processus de modélisation peut, sous certaines conditions de normalisation, être découpé en sous-modèles, qui seront à la charge des différents modélisateurs. Ce découpage permet de faire intervenir à différents niveaux plusieurs personnes en fonction de leurs compétences et de leurs connaissances du domaine.

Un formalisme spécifique ne peut prendre en compte toutes les caractéristiques nécessaires à l'analyse des systèmes complexes. Nous avons choisi comme support un multi-

formalisme DEVS [Zeigler, 1976] qui offre la possibilité d'intégrer, pour un même système, des modèles décrits à partir de différents formalismes (équations différentielles, réseaux de neurones, réseaux de Pétri...). De plus, ce formalisme propose une séparation explicite entre les aspects de modélisation et de simulation, ce qui permet au modélisateur de s'abstraire totalement de la partie simulation et de sa mise en œuvre.

Plus précisément, ce formalisme présente la particularité de se situer au niveau de la simulation modulaire, hiérarchique et centré sur la notion d'états du système. Au niveau de la simulation, le temps est vu de manière continue et agit directement sur les états du système. DEVS est donc un formalisme à événements discrets qui s'adapte bien aux simulations de systèmes naturels complexes.

1.1. Modélisation à événements discrets

Le formalisme DEVS repose sur la définition de deux types de modèles : les modèles atomiques et les modèles couplés. Les modèles atomiques permettent de représenter le comportement de base du système. Les modèles couplés, quant à eux, sont définis par un ensemble de sous-modèles atomiques et/ou couplés. Ils permettent de représenter la structure interne du système grâce à la définition de couplages entre modèles.

1.1.1. Modèles atomiques

Un modèle atomique permet de rendre compte des comportements d'entrée/sortie et des changements d'états du sous-système. C'est à ce niveau qu'intervient le spécialiste du domaine et que se fait le choix du formalisme qui va décrire le comportement d'une partie du système. Le modèle atomique est centré sur les transitions d'états, ces états évoluent en fonction de deux types d'événements reçus par le modèle : les événements internes et les événements externes (Figure 1.1).

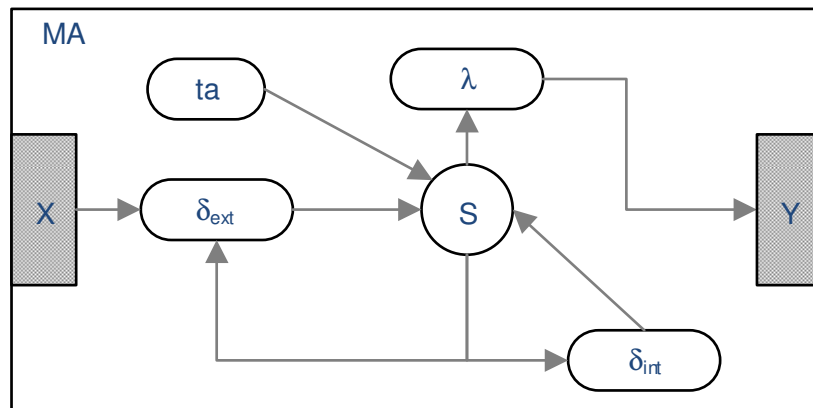


Figure 1.1. Représentation d'un modèle atomique

Un modèle atomique est défini par la structure suivante :

$$MA = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$$

- X : l'ensemble des ports d'entrée, à travers lesquels les événements externes, correspondant aux sollicitations du système, sont reçus par le modèle ;
- Y : l'ensemble des ports de sortie, à travers lesquels les réponses du modèle sont envoyées vers l'extérieur ;
- S : l'ensemble des états qui caractérisent le modèle atomique ;
- $\delta_{int} : S \rightarrow S$, la fonction de transition interne qui spécifie vers quel nouvel état le modèle évoluera, quand un événement interne survient ;
- $\delta_{ext} : Q \times X \rightarrow S$, la fonction de transition externe qui spécifie vers quel nouvel état le modèle évoluera, quand un événement externe survient ;
- $\lambda : Q \rightarrow Y$, la fonction de sortie qui génère une réponse du modèle suite au traitement d'un événement interne ;

- $ta : S \rightarrow R+$, la fonction d'avancement du temps qui retourne le temps auquel le modèle exécutera un événement interne si aucun événement externe ne survient.

Tel que :

- $Q = \{(s,e) \mid (s \in S \text{ et } 0 < e < ta(s))\}$ est l'ensemble total des états du modèle.
- e est le temps écoulé depuis la dernière transition et s l'ensemble partiel d'états pour la durée de $ta(s)$ en absence d'activation externe.

1.1.2. Modèles couplés

La représentation de systèmes complexes est généralement difficile à mettre en œuvre. En effet, ceci implique la prise en compte d'une multitude d'éléments, reliés les uns aux autres par de nombreuses connexions. Pour pallier ce problème, le formalisme DEVS propose une hiérarchie de descriptions basée sur la notion de modèles couplés, qui permet d'introduire graduellement les composants du système étudié.

Un modèle couplé permet la construction de modèles à partir d'un ensemble de sous-modèles et de trois relations de couplage. Les sous-modèles peuvent être des modèles atomiques et/ou couplés, les sous-modèles couplés sont, à leur tour, décrits comme l'interconnexion d'un ensemble de sous-modèles. Un modèle couplé est défini par la structure suivante :

$$MC = \langle X, Y, D, \{M_d \in D\}, EIC, EOC, IC \rangle$$

- X : ports d'entrée, sur lesquels les événements externes sont reçus ;
- Y : ports de sortie, sur lesquels les réponses du modèle sont envoyées ;
- D : l'ensemble des noms des modèles intervenant dans le modèle couplé ;
- $\{M_d\}$ est l'ensemble des modèles, où pour chaque $d \in D$, M_d peut être soit un modèle atomique DEVS, soit un modèle couplé DEVS.

- *EIC* : l'ensemble des liens d'entrée qui connectent les entrées du modèle couplé à une ou plusieurs entrées des sous-modèles ;
- *EOC* : l'ensemble des liens de sortie qui connectent les sorties d'un ou plusieurs sous-modèles au modèle couplé ;
- *IC* : l'ensemble des liens internes qui connectent les sous-modèles entre eux.

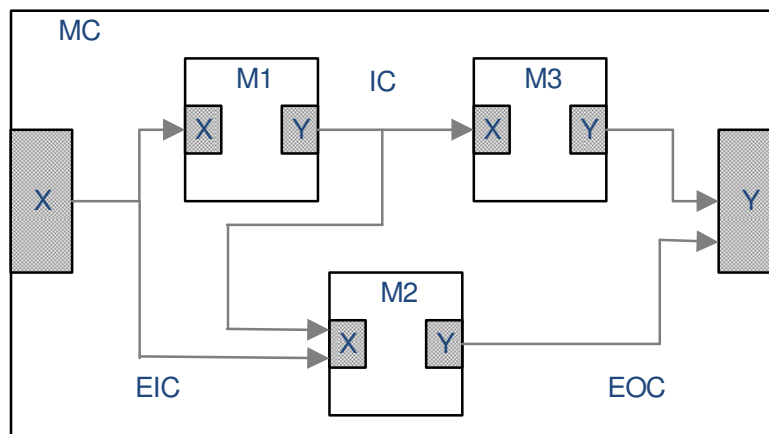


Figure 1.2. Représentation d'un modèle couplé

La Figure 1.2 représente un modèle couplé DEVS composé de trois sous-modèles. Tous ces modèles disposent de ports d'entrée et de sortie à travers lesquels les échanges d'information sont réalisés. Le modèle couplé MC correspond à une description structurelle du modèle et est décrit par la liste des sous-modèles qui le composent et par les trois types de couplage. Les couplages EIC sont des liaisons de ports d'entrée à ports d'entrée, les couplages IC représentent des liaisons de ports de sortie à ports d'entrée, et enfin les liaisons EOC sont des couplages de ports de sortie à ports de sortie. Les sous-modèles M1, M2 et M3 pouvant à leur tour être décrits comme des modèles couplés ou comme des modèles atomiques.

1.2. Simulation à événements discrets

Parallèlement à l'élaboration des modèles de description, B. P. Zeigler a développé le concept de simulateur abstrait [Zeigler, 1976] qui permet d'effectuer la simulation d'un modèle décrit suivant le formalisme DEVS. Un simulateur abstrait représente une description algorithmique permettant de mettre en œuvre les instructions des modèles du formalisme, afin de générer leur comportement. Outre le fait que la simulation soit dirigée par les événements, c'est-à-dire que les transitions d'état interviennent à intervalles de temps irréguliers en fonction des événements programmés, l'originalité de cette démarche réside dans le fait que la construction du simulateur est indépendante du modèle et du formalisme utilisé pour le décrire.

Un simulateur abstrait représente une description algorithmique permettant de mettre en œuvre les instructions implicites des modèles du formalisme DEVS afin de générer leur comportement. Un tel simulateur est obtenu en faisant correspondre à chaque élément du modèle un composant de simulation de type processeur. L'originalité de cette démarche réside dans le fait que la construction du simulateur est indépendante du modèle.

La simulation de modèles a pour objectif la génération d'événements de sortie, à partir d'événements d'entrée. Un simulateur abstrait est représenté par une structure de type graphe, nommé arbre de simulation et composé de différents processeurs. Chaque modèle (atomique ou couplé) est géré par un composant spécifique dérivé du type processeur (simulateur ou coordinateur). La communication entre les composants s'établit au travers d'un échange de messages qui représentent les événements qui influencent les modèles.

L'arbre de simulation est construit à partir de trois types de nœuds : le root, les simulateurs et les coordinateurs. Le root est le composant, qui dérive également du processeur, et qui se trouve à la racine de l'arbre de simulation. Ce composant de simulation n'est associé à

aucun modèle et n'existe dans l'arbre de simulation qu'au niveau de la racine. Les coordinateurs sont les « branches de l'arbre » et correspondent à des modèles couplés. Les simulateurs sont les « feuilles de l'arbre » et correspondent aux modèles atomiques.

La simulation est exécutée à l'aide de différents types de messages échangés entre pilotes. Ces messages contiennent les informations décrivant les événements à traiter au cours de la simulation. Six types de messages permettent de réaliser la simulation :

- *i-message* assure l'initialisation des variables et des paramètres des modèles ;
- **-message* indique qu'un événement interne est prévu pour un simulateur ;
- *x-message* indique l'arrivée d'un événement externe à destination d'un processeur ;
- *y-message* correspond à un message de sortie ;
- *d-message* indique qu'une transition d'état a été exécutée par un simulateur.

Le processeur de type root permet de lire des données d'entrée sur un support d'enregistrement (base de données, fichier ...) et de les transformer en messages de type *x* à destination du modèle couplé de plus haut niveau. Le composant de simulation root réceptionne également les événements de sortie sous la forme de *y-messages* et enregistre les résultats de la simulation. Avant chaque simulation, un message de type *i* est généré par le root à destination de tous les nœuds de l'arbre de simulation afin de permettre l'initialisation de tous les paramètres et de toutes les variables des processeurs et des modèles sous-jacents. Le root synchronise les différents processeurs en faisant évoluer le temps global de la simulation.

Les coordinateurs ont pour mission principale de dispatcher les événements vers les sous-modèles ou à destination du nœud supérieur dans le cas d'un événement de sortie. Ils disposent également d'un échéancier dans lequel ils peuvent conserver les messages avant de les router

dans l'arbre de simulation en utilisant les différents couplages du modèle couplé associé. Une autre tâche importante dédiée aux coordinateurs est la gestion d'une variable qui indique à quelle date sera traité le prochain événement par le coordinateur ou l'un de ses enfants. À la réception d'un *x-message*, le coordinateur recherche dans sa liste *EIC* le destinataire et l'enregistre dans son échéancier. À la réception d'un *y-message*, les listes *IC* puis *EOC* sont consultées et le message est transformé en *x-message* si le destinataire est dans *IC* et en *y-message* si le destinataire est dans *EOC*. Un *d-message* sert à la synchronisation des processeurs, enfin lorsque le temps de vie d'un simulateur est à 0, le coordinateur génère un **-message* à destination de ce simulateur.

Les simulateurs sont les feuilles de l'arbre de simulation et doivent exécuter les fonctions de transition des modèles atomiques en fonction des types de messages reçus. Si un *x-message* est reçu sur l'un des ports d'entrée du modèle, la fonction de δ_{ext} est exécutée puis la fonction *ta* est évaluée permettant de calculer le temps de vie du composant et donc de connaître la date du prochain événement interne. La réception d'un **-message* provoque l'exécution des fonctions λ et δ_{ext} . Enfin, *ta* est à nouveau évalué et un *d-message* est transmis au coordinateur parent.

Dans son approche théorique sur la modélisation et la simulation de systèmes, B.P. Zeigler introduit une notion de cadre expérimental associé [Zeigler, 2000] à l'utilisation des modèles. Le cadre expérimental décrit comment seront exécutées les simulations dans le cadre de l'utilisation qu'on souhaite proposer pour les modèles. Ce concept est présenté en détail dans la section suivante.

2. Cadre expérimental

Un cadre expérimental (ou Experimental Frame) est la spécification des conditions dans lesquelles un système est observé ou dans lesquelles nous l'expérimentons (Figure 1.3). Ce cadre expérimental est la partie qui nous intéresse le plus dans cette étude comme nous le présenterons dans la suite de ce document. Il est réalisé comme un système qui interagit avec le système source pour obtenir les résultats intéressants dans des conditions spécifiées.

Ces conditions spécifiées sont nommées les « contraintes » qui ont été soumises pour l'expérimentation ou l'observation d'un système. Ces contraintes doivent être en accord avec les objectifs et les questions que le modélisateur ou expérimentateur a à l'esprit. En d'autres termes, c'est une formulation opérationnelle des objectifs qui motive le développement de l'application de la Modélisation et de la Simulation.

Le système source est l'environnement réel ou virtuel que nous souhaitons modéliser. Il est vu comme une source de données observables sous forme de trajectoires de variables indexées par le temps. Il est alors considéré comme une boîte noire sur laquelle l'environnement agit et réagit grâce à des stimuli. Les données qui ont été recueillies après l'observation ou l'expérimentation du système constituent la base de données du comportement de ce système. Il faut se rappeler que ces données sont vues ou générées lorsque le système est plongé dans un contexte particulier du monde réel. Ces données sont affichées ou acquises par les cadres expérimentaux choisis par le modélisateur.

Un modèle est la spécification du système réel ou virtuel. Les modèles peuvent être exprimés dans une variété de formalismes qui peuvent être compris comme un moyen de spécifier les sous-classes d'un système. Un formalisme utilise des abstractions comme les équations différentielles, les réseaux de neurones, les réseaux de Pétri, etc. La complexité des

systemes nécessite d'avoir différents formalismes et différentes formes d'abstraction, parfois successivement, parfois simultanément.

Un **simulateur** est un système de calcul (comme un seul processeur, ou un réseau de processeurs ou, plus abstraitement, un processeur logique ou un algorithme) qui est capable d'exécuter un modèle pour produire son comportement sous forme de vecteurs de sortie. L'originalité de cette démarche réside dans le fait que la construction du simulateur est indépendante du modèle et du formalisme utilisé pour le décrire.

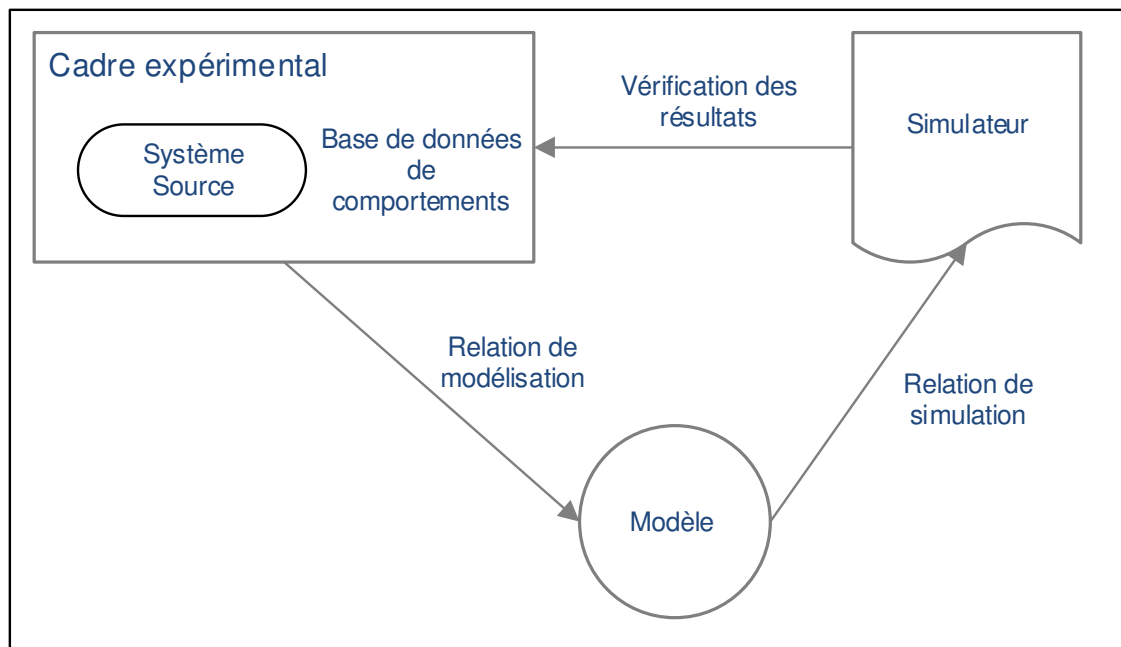


Figure 1.3. Cadre expérimental dans le processus de modélisation/simulation

2.1. Développement d'un cadre expérimental

La notion de cadre expérimental permet de spécifier les conditions d'expérimentation ou d'observation d'un système. En d'autres termes, c'est la spécification des conditions dans lesquelles un système est observé ou dans lesquelles nous l'expérimentons. Aussi, le cadre expérimental permet d'établir un jeu de données pertinentes et est donc toujours lié au modèle

simulé. Les données du cadre expérimental sont les données utilisées pour décrire le contexte dans lequel un modèle donné doit être valide. Ces données sont présentées aux utilisateurs comme une partie de l'information descriptive d'un modèle pour les aider à prendre une décision concernant ce modèle.

Le processus de construction d'un cadre expérimental est réalisé pour analyser et comparer les résultats obtenus par rapport à ceux qui sont désirés. Le développement d'un cadre expérimental, illustré en Figure 1.4, est inspiré par le diagramme « Object Driven Methodology for Experimental Frame » extrait de la thèse de Saehoon Cheon [Cheon, 2007].

Ce développement commence avec la conception de n'importe quel modèle à modéliser. Nous devons, tout d'abord, nous poser des questions précises sur le comportement d'un système et celles-ci nous conduisent à définir les objectifs de modélisation. Cette étape nous permet de sélectionner les variables qui ont un intérêt pour le modélisateur puis de les transmettre au cadre expérimental, qui gère l'expérimentation du modèle construit afin de le valider et le vérifier.

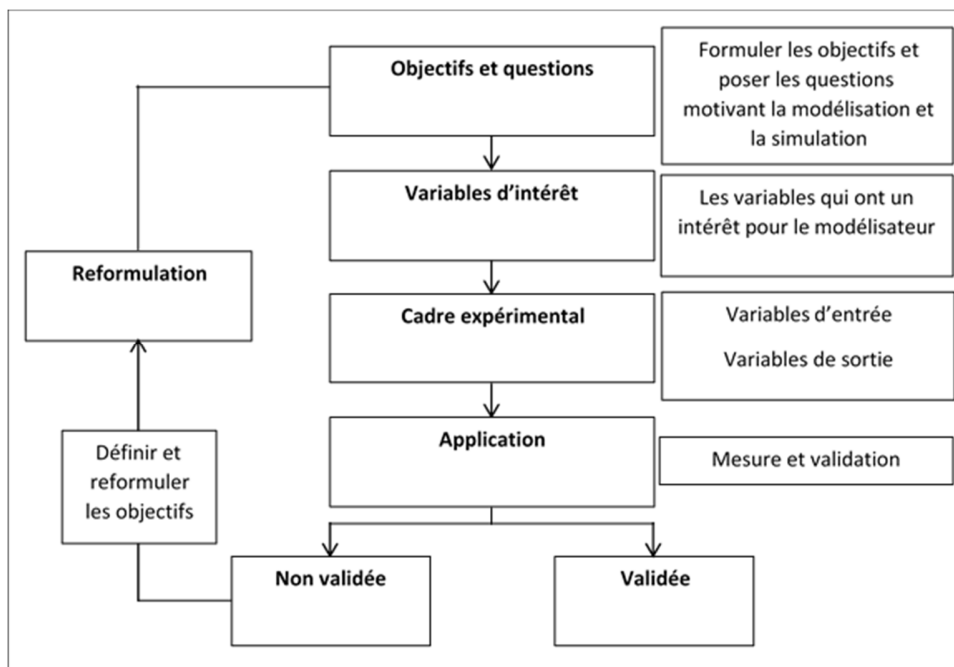


Figure 1.4. Démarche de développement d'un cadre expérimental

Le cadre expérimental se compose de quatre segments. D'abord, les variables d'entrée sont les variables que l'expérimentateur souhaite traiter et qui influencent le système ; elles sont sous son contrôle ou sous le contrôle de l'environnement. Les variables de sortie sont des variables qui sont influencées par le système et sont mesurables par l'observateur. Les variables de contrôle d'exécution présentent les contraintes sous lesquelles le CE du système réel est soumis à l'expérimentation. Dans notre cas, ce sont les conditions d'arrêt des simulations qui sont les variables de contrôle d'exécution. La synthèse des correspondances d'un cadre décrit les correspondances entre les états du système. Elle représente un résumé des résultats des expérimentations obtenues en fonction des objectifs formulés au début.

L'étape suivante valide l'étape du développement du cadre expérimental. Si l'application n'est pas validée, nous devons reformuler les objectifs et définir de nouveau les questions sur les comportements du système.

2.2. Application dans le cas des feux de forêt

En développant le cadre expérimental de cette étude, nous devons formuler les objectifs et nous poser les bonnes questions comme indiqué dans la Figure 1.4.

Étape 1 – Objectifs et questions. Nous posons la question : comment étudier les comportements d'un feu ? Comment avoir une action de lutte et de prévention contre les feux de forêt ? Quels sont les objectifs qui conduisent à développer un outil de simulation des feux de forêt ? Ces questions nous conduisent à définir deux grands objectifs, à savoir :

Créer un outil de simulation des feux de forêt à l'échelle territoriale, nationale et internationale en vue d'aider les experts du domaine à mener des actions de lutte et de prévention.

Cet outil doit prendre en compte les paramètres environnementaux qui influent sur le comportement d'un feu, comme le climat, la topographie, l'humidité, les conditions atmosphériques et les combustibles de terrain.

Étape 2 – Variables d'intérêt. De la formulation des objectifs, nous pouvons identifier deux groupes de variables d'intérêt. Les variables d'intérêt qui décrivent le comportement du feu et les variables environnementales qui ont une influence sur son comportement.

Les variables d'intérêt qui décrivent le comportement du feu :

- Date de début et de fin du feu
- Points d'éclosion
- Surface brûlée
- Contour final du feu
- Vitesse de propagation du feu
- Intensité du feu
- Présence de sautes de feu

Les variables d'intérêt qui ont une influence sur le comportement du feu :

- Données climatiques (vitesse et direction du feu, humidité, température...)
- Données géographiques (topographie, élévation)
- Combustible

Après cette étape, nous sommes en mesure de définir le cadre expérimental qui nous permettra de tester et de valider l'outil de simulation développé. Cette étape correspond à la partie la plus importante de cette thèse.

Étape 3 – Cadre expérimental. Un cadre expérimental est nécessaire pour expérimenter un modèle, afin que sa relation de modélisation soit validée et que sa relation de simulation soit

vérifiée. Pour cela, nous devons définir les entrées et les sorties du cadre expérimental ainsi que les variables de contrôle.

Les variables d'entrée de ce cadre sont :

- L'état final du feu simulé (date de fin, contour final)
- Les variables de sortie sont : l'état initial du feu à simuler (date de début, points d'éclosion)

Les variables de contrôle d'exécution du cadre (ce sont les conditions d'arrêt de la simulation) sont :

- Surface totale brûlée
- Durée du feu simulé

Étape 4 – Application. Enfin, l'outil de simulation de la propagation des feux de forêt est l'application qui sera mesurée et validée. Si cet outil est validé, alors le modèle et son cadre expérimental sont validés, sinon nous devons revoir les données et éventuellement les objectifs. La mise en œuvre d'un cadre expérimental s'appuie sur une base de données qui contient les résultats des simulations, les données observées et toutes les informations ou paramètres utiles à leur traitement. Dans la section suivante, nous présentons différents formats et outils utilisés dans le cadre de cette thèse pour manipuler les données.

2.3. Mise en œuvre

Nous donnons une description de la notion de cadre expérimental telle que nous l'avons mis en œuvre dans le cadre du calcul de l'incertitude lors des simulations des feux de forêt. Nous avons choisi de représenter le cadre expérimental en le décrivant comme un modèle couplé composé de trois modèles atomiques, le Générateur, l'Accepteur et le Transducteur. Le cadre expérimental

proposé permet de mettre en relation les données observées, de sélectionner les variables d'intérêt en fonction des objectifs que nous nous sommes fixés, d'exécuter les simulations en respectant nos contraintes et, enfin, de traiter les résultats obtenus afin de les valider ou de les rejeter. La Figure 1.5 présente de manière synthétique le cheminement des données et les interactions entre les composants.

Au départ, nous disposons d'un entrepôt de données brutes, correspondant à des fichiers issus des bases de données d'observation comme Prométhée ou à des fichiers de texte fournis par des services tels que l'ONF. La première étape de notre travail a consisté à sérialiser toutes ces données pour les rendre exploitables et à choisir un format et une description utilisables par le cadre expérimental. Un outil spécifique a été développé pour « parser » et mettre en forme les observations dans un format de données auto-documenté, NetCDF.

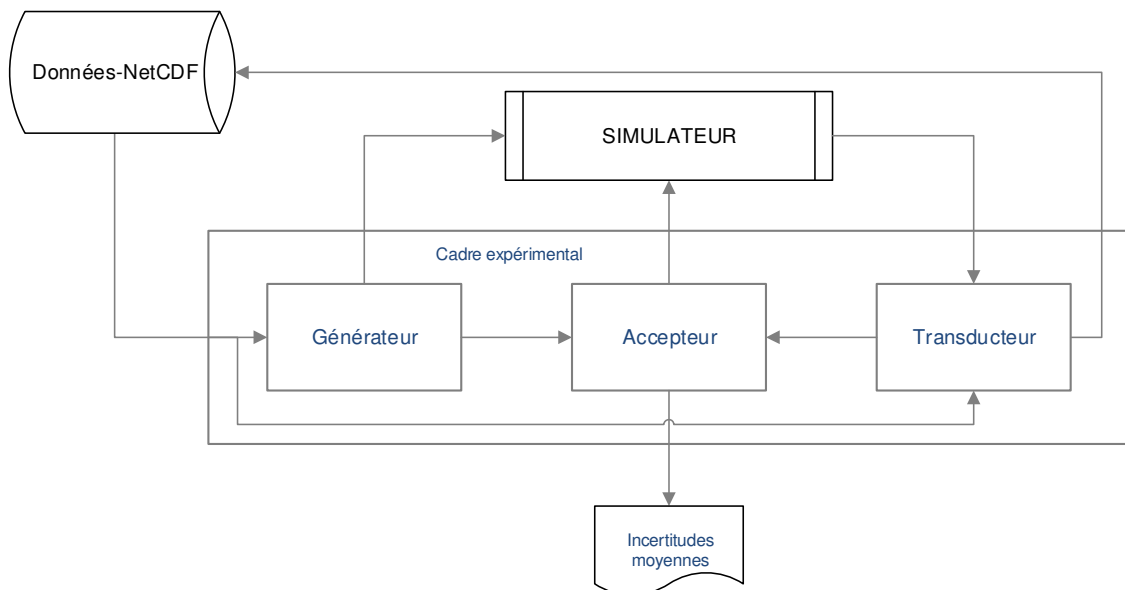


Figure 1.5. Cadre expérimental

Le Générateur du cadre expérimental est un composant qui doit pouvoir générer les événements d'entrée à envoyer au modèle de propagation. Pour cela, il accepte en entrée des

fichiers au format NetCDF correspondant à l'état d'un feu à l'instant t ainsi que les données environnementales telles que le vent, le combustible, l'état du terrain... La principale tâche du générateur consiste à extraire des données d'entrée, afin de construire un plan d'expérimentation, les conditions initiales et finales du feu observé. Les variables d'intérêt sont transmises à l'Accepteur, tandis que les conditions initiales et finales du feu sont, elles, envoyées au simulateur.

Le Générateur traite une seule observation à la fois en extrayant les données nécessaires à l'exécution d'une simulation, comme les points de déclenchement du feu, la date et l'heure de déclenchement. À la fin de la génération, le Générateur stimule le modèle à expérimenter par ces données en communiquant sur la sortie du cadre expérimental les conditions initiales du feu.

L'Accepteur détermine quelles sont les variables d'intérêt du modèle à traiter et les méthodes de calcul des erreurs qui seront utilisées. En fonction des variables d'intérêt réceptionnées du Générateur et du Transducteur, ce modèle sera en mesure de calculer les erreurs commises. L'Accepteur est également en mesure de contrôler la simulation. Enfin la simulation génère des résultats au format NetCDF compatible avec le format des données initiales ; ceci nous permettra d'utiliser ces résultats comme données d'entrée pour lancer d'autres simulations. Les résultats sont post-traités par le Transducteur qui pourra en extraire les variables d'intérêts et les transmettre vers l'Accepteur.

Les entrées du cadre expérimental sont donc des fichiers (NetCDF) aussi bien pour les feux observés que pour les résultats de la simulation. La cohérence de ces formats permet au cadre expérimental de calculer les scores pour les erreurs. Enfin, à partir d'un échantillon significatif de scores, il est possible d'en déduire l'erreur moyenne pour chaque modèle simulant.

3. Outils utilisés

Dans cette partie, nous allons présenter les lexiques et grammaires des données utilisées. La définition de ces formats est une étape fondamentale permettant de donner du sens aux observations et aux simulations. Sans format correctement défini, il est en effet impossible d'effectuer les comparaisons nécessaires à la vérification de modèles. Les observations sont, dans le cadre de cette étude, récupérées dans la base de données en ligne Prométhée [Arnould P. et al., 2008]. Cette base de données est alimentée par les différents services qui travaillent dans le domaine de la lutte ou de la prévention des incendies (ONF, SDIS, DDTM...) dans la région méditerranéenne. Extraits de cette base, les fichiers sont formatés et complétés avec des données environnementales afin d'être enregistrés dans une nouvelle base de données, au format NetCDF, qui offre un moyen pratique pour compiler les observations et les résultats de simulation sous un format standard général et indépendant de l'architecture matérielle.

Afin de pouvoir visualiser les résultats des observations et des simulations avec des navigateurs tels que Google Earth, nous avons développé un outil permettant de générer de manière automatique des fichiers KML correctement formatés à partir des données issues de fichiers NetCDF.

3.1. Le système source

Conçue et lancée en 1973, la base de données Prométhée couvre 15 départements dans le Sud-Est de la France et notamment la Corse qui est notre domaine d'étude. Les critères de recherche de cette base en ligne sont multiples ; on peut par exemple citer des critères comme l'origine, la surface parcourue ou la zone.

Les données extraites de Prométhée sont disponibles au format ASCII :

- Date et temps du départ et de fin du feu

- Coordonnées du point d'allumage
- Nom et code de la commune
- Département
- Surface parcourue
- Contour final de l'incendie
- Contours intermédiaires avec leurs temps d'observation (s'ils existent).

C'est à partir de ces données, associées aux données environnementales sur le vent et type de combustible, que nous allons constituer la base de données du comportement de notre système source. Nous avons choisi de formater ces données au format NetCDF. Un conteneur NetCDF est un dispositif informatique qui sauvegarde des informations regroupées sous forme de collections ou de tableaux quelle que soit la quantité d'information. Aussi, il nous permet d'organiser et de structurer ces informations de manière à pouvoir facilement manipuler le contenu.

NetCDF (Network Common Data Format) [Rew and Davis, 1990] est géré comme un conteneur par une bibliothèque logicielle, ou API, avec pour but la lecture et la création de fichiers. Un fichier NetCDF est composé de tableaux multidimensionnels qui contiennent eux-mêmes les données proprement dites. Une autre caractéristique de ce format est qu'il est auto-documenté : c'est-à-dire qu'il existe un en-tête qui décrit le nom et la disposition des données dans le reste du fichier, ce qui permet la lecture des données en accès direct. Un fichier NetCDF est composé de dimensions, de variables et d'attributs.

Une dimension est utilisée pour présenter un paramètre physique réel comme le temps, la latitude, la longitude, ou la hauteur dans les simulations climatiques. Chaque dimension est définie par un nom et une longueur qui est un entier positif. Dans le NetCDF-4, n'importe quel

nombre de dimensions illimitées peut être utilisé. L'intérêt est d'avoir un nombre illimité de tableaux qui ont une taille dynamique qui n'est pas définie à l'avance. Les anciennes versions de NetCDF n'ont qu'un seul tableau illimité.

Afin de sauvegarder un lot de données, le format NetCDF utilise des variables qui sont des tableaux de dimension N contenant des données de même type. Une variable a un nom, un type de donnée et une forme qui est en fonction de/des dimension(s). Les variables dans les fichiers NetCDF peuvent être d'un des six types de base (char, byte, short, int, float, double).

Les attributs du format NetCDF sont utilisés pour stocker des informations liées aux variables (données auxiliaires ou métadonnées). Elles ressemblent aux informations stockées dans les dictionnaires de données ou le schéma des systèmes de base de données conventionnelles. La plupart des attributs fournissent des informations sur une variable spécifique. Un attribut a un nom et est associé à une variable. On peut ainsi, par exemple, donner à une variable une valeur par défaut, une description ou un sigle, consultables à tout moment.

3.2. Visualisation des résultats

Nous utilisons deux interfaces de sortie pour visualiser les résultats incendies et ainsi vérifier qualitativement la cohérence globale des résultats. Le premier utilise la bibliothèque Python Matplotlib à partir des matrices incendies, le second est un export au format KML. Le format KML, basé sur XML, est interprété au sein d'un SIG ou de logiciels spécifiques tels que Google Earth ou WorldWind (NASA). La Figure 1.6 présente un front de flamme encodé en KML. Les contours de ce front contenant une date (balise "when") et le contour (balise "coordinates"). La représentation de cette structure KML est présentée en Figure 1.7.

```
<when>2015-03-16T09:52:24Z</when>  
<LineString><coordinates>  
  9.0349985193352,42.326252779396,0 9.0351953228149,42.326323084351,0 (...)  
</LineString></coordinates>
```

Figure 1.6. Structure du fichier KML



Figure 1.7. Exemple de représentation correspondant au contour d'un feu

4. Représentation des feux

Dans cette section, nous présentons la manière dont sont généralement et historiquement enregistrées les données des feux de forêt afin de déterminer l'ontologie nécessaire. Nous nous sommes intéressés à trois types de présentations : graphique, textuelle et spatiale normalisée. Nous présentons également les formats de fichiers utilisés par les principaux outils de simulation.

4.1. Les observations de terrain

Historiquement, les incendies des forêt ont été documentés sous forme de rapports permettant d'analyser les réussites et les échecs des opérations de lutte. Au cours du temps, les informations sauvegardées ont été de plus en plus nombreuses et précises à mesure que les besoins évoluaient. Trois types de présentation des données existent dans les rapports rédigés : une présentation graphique, une présentation textuelle et une présentation spatiale normalisée.

4.1.1. Représentation graphique

En général, la présentation graphique prend la forme d'une carte qui est une représentation partielle d'un feu réel. Différentes informations sont représentées généralement dans un mode plan à l'aide d'un ensemble de symboles ou d'icônes colorées. Par exemple, un feu pourrait être représenté sur une carte par :

- le point d'allumage "X" ;
- différents contours qui peuvent être des "lignes de petits carrés", des "lignes pointées" ou des "polygones" ;
- un ensemble d'icônes ou de symboles caractérisant le feu.

La légende de la carte affiche les données correspondant à chaque symbole. Les icônes et les symboles réels sont généralement normalisés au niveau des pays. Les sinistres peuvent être délimités dans l'espace suivant des formes linéaires ou surfaciques.

Voici deux exemples de représentations graphiques dites SITUations TACTiques (SITAC) fournies par des organismes différents. La Figure 1.8 correspond au Feu de Valle Male, à Aullène, qui a démarré le 25 juillet 2009. Les informations sont affichées sous forme de symboles : le polygone en rouge correspond au contour final du feu, la flèche noire indique la direction du vent et les textes dans les encadrés blancs sont les notes d'observations. Les petits rectangles colorés décrivent les actions de lutte contre ce feu. Les informations de cette carte sont fournies par le SDIS 2A le SDIS 2B, l'ONF et le DDAF 2A. Source : SDIS 2A.

La Figure 1.9 correspond au feu de l'Ortolo qui a débuté le 23 juillet 2009. Le polygone en rouge est le contour du feu. La surface totale brûlée est délimitée par le polygone rayé. Les informations de cette carte sont fournies par le DDAF 2A et le Service Eau Environnement Forêt. Source : IGN/DDEA.

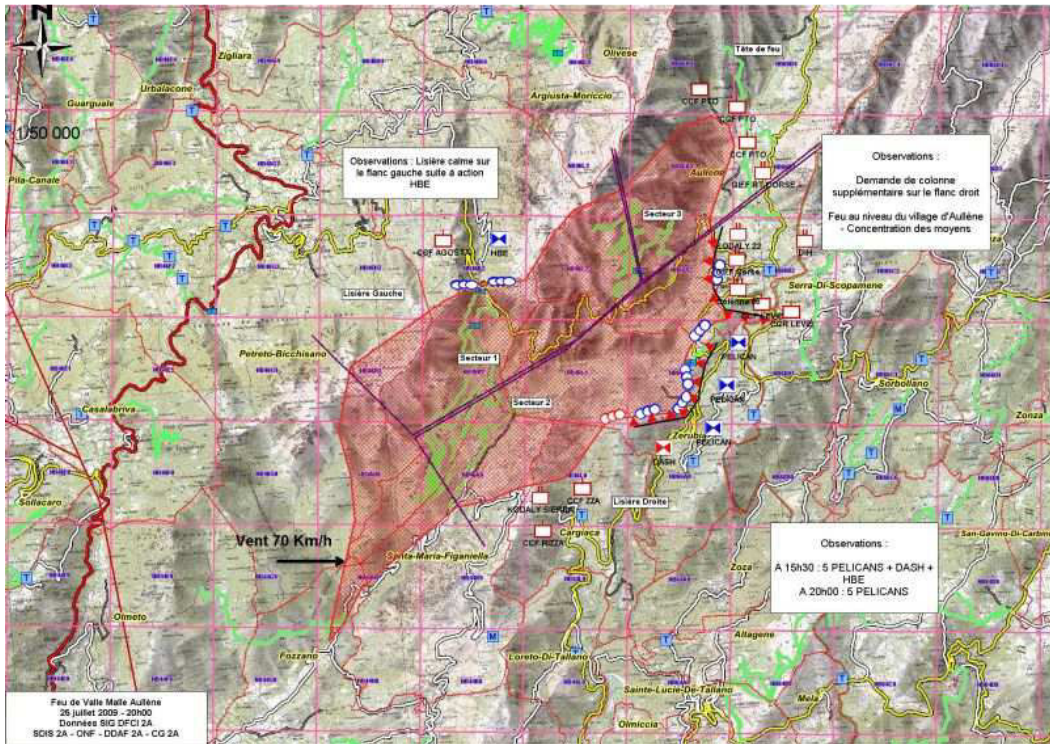


Figure 1.8. Représentation graphique du feu d'Aullène à l'été 2009

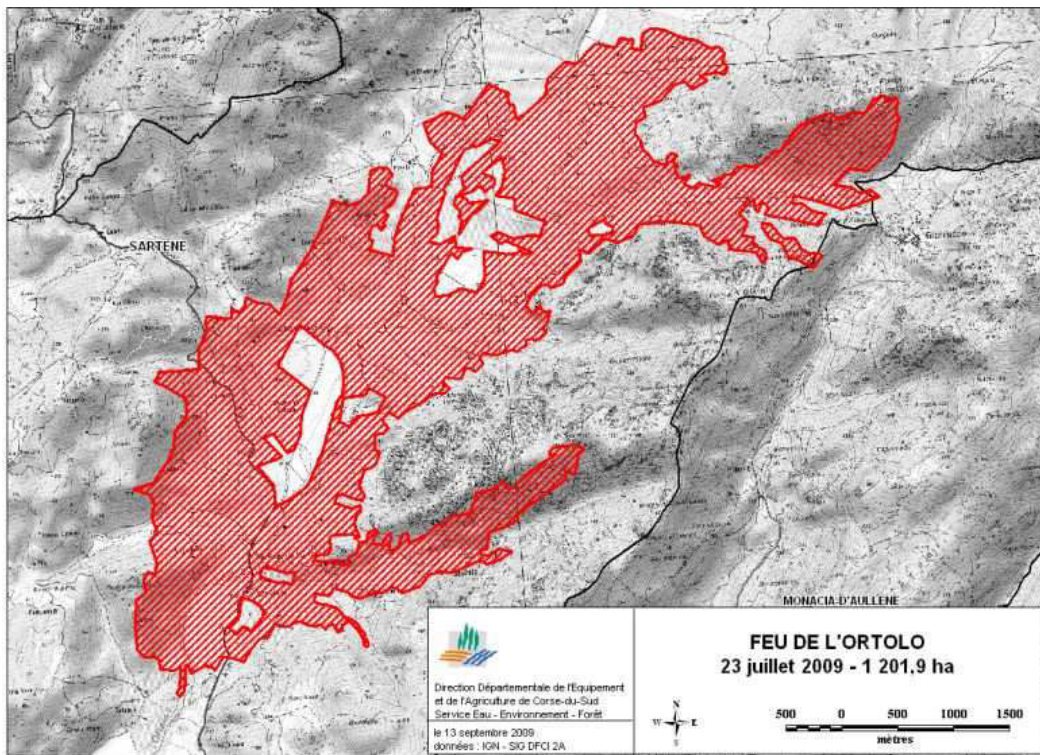


Figure 1.9. Représentation du feu de l'Ortolo à l'été 2009

Ces outils permettent de schématiser de façon simple les principales informations opérationnelles relatives au feu et sont utilisées par les services d'incendie pour connaître l'état final du feu. Certaines cartes incluent également des informations sur les moyens d'intervention mobilisés et les actions effectuées. En revanche, ces types de représentations correspondent à des informations élémentaires et généralement incomplètes ou erronées. De plus, leur niveau de précision est nettement insuffisant. Enfin, il est impossible d'utiliser des outils génériques permettant d'extraire de ces cartes des données nécessaires dans le cadre de nos travaux.

4.1.2. Présentation textuelle

La présentation textuelle est un rapport qui se compose des observations de terrain pour les paramètres tels que la vitesse de propagation, l'intensité, la forme, la localisation finale du feu, la hauteur de la flamme et la hauteur des fumées avec un temps d'observation pour chacune. La plupart des informations disponibles pour les feux observés existe dans cette présentation.

Cette façon de présenter les données est très flexible et s'avère très cohérente pour les pompiers sur le terrain de lutte. Ce type de représentation se prête beaucoup plus au développement de « parseurs » pour en extraire les informations utiles à une simulation. Malheureusement, dans la pratique, on constate que, contrairement au format graphique, il n'existe aucun standard entre les organismes. Même au sein d'un même organisme il existe de grandes disparités entre les fichiers. En fait, chaque fichier dépend de la personne qui a eu la charge d'enregistrer les informations sur l'incendie. Tout ceci ne nous permet pas de généraliser une démarche pour un traitement automatique.

4.1.3. Présentation spatiale normalisée

Un autre moyen plus récemment utilisé pour représenter les données du feu correspond aux images satellite normalisées. Le SERTIT (Service Régional de Traitement d'Image et de

Téledétection) est un service d'un laboratoire de recherches au sein de l'université de Strasbourg qui offre la possibilité d'extraction et de mise en forme de l'information à partir des données des systèmes d'observation de la terre. Le SERTIT traite les données issues des satellites opérationnels : SPOT, LANDSAT, Ikonos, Envisat et Radarsat.

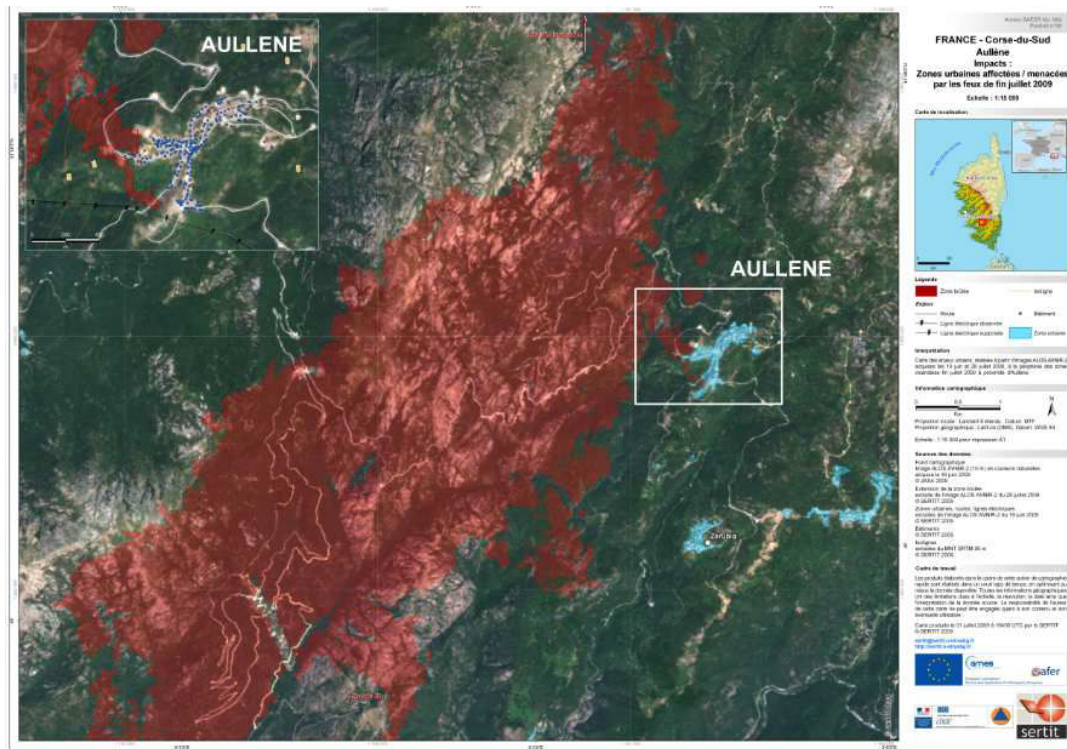


Figure 1.10. Représentation spatiale normalisée du feu d'Aullène à l'été 2009

À l'aide des données spatiales et territoriales collectées, la Figure 1.10 est une image satellite correspondant à une représentation spatiale normalisée du feu d'Aullène que nous avons présenté sous sa forme textuelle au paragraphe précédent. Comme pour les figures précédentes, le polygone rouge représente le contour final du feu. Sur la droite de l'image, on a accès à toutes les informations disponibles sur cet incendie.

L'analyse de ces sources, si elles ne sont pas exploitables, nous permet toutefois de déterminer l'ontologie (soit la base de nom et le type d'information) récurrente utilisée au sein des incendies. Cette ontologie servira à construire le format où seront encodées les informations

provenant de bases de données aux formats disparates mais numérisées telles que Prométhée, ou provenant de services incendies.

4.2. Le résultat des simulations

Nous présentons maintenant quelques exemples de logiciels de simulation ainsi le format des données qu'ils utilisent et qu'ils produisent. L'analyse comportementale des feux de forêt passe par le développement d'outils de simulation performants et efficaces. Pour que les résultats soient conformes aux situations réelles, ces outils ont besoin de pouvoir accéder à des données spatiales normalisées. Ces données sont, par exemple, le point d'allumage du feu, la vitesse de propagation, son intensité, la surface totale brûlée, les contours et les hauteurs de la flamme et des fumées.

D'autres données environnementales comme la direction et la vitesse du vent, les données géographiques de l'endroit, les distributions de combustibles existants (fuels), le climat et le taux d'humidité sont indispensables pour initialiser une simulation. Des formats de données spécifiques ont été développés pour s'adapter aux logiciels de simulation. Voici quelques exemples de logiciels avec les formats de données nécessaires pour lancer les simulations et les formats de sortie obtenus.

4.2.1. Farsite

FARSITE est un système de modélisation déterministe, ce qui signifie que les résultats des simulations peuvent être comparés directement aux feux observés pour avoir un indice d'incertitude. Par conséquent, il est largement utilisé par le service forestier des États-Unis « U.S. Forest Service (USFS) » [Finney, 2004].

FARSITE (Fire Area Simulator) est un modèle de propagation du feu et de ses comportements sur le terrain. Il utilise des données spatiales obtenues à partir d'un système

d'information géographique (SIG) ainsi que des données météorologiques et de distribution du vent pour simuler la propagation du feu sous forme d'un front.

FARSITE peut être utilisé sur un ordinateur personnel sous l'environnement d'exploitation WINDOWS. Il est prévu pour aider à la gestion et à la lutte contre les feux et à la planification de la lutte contre des feux potentiels [Finney, 1995].

Le modèle du FARSITE intègre cinq sous-modèles qui décrivent le comportement du feu :

1. Feu de la surface ([Rothermel, R.C. ,1972], [Albini, F.A., 1976]),
2. Feu de cime ([Rothermel, R.C. ,1991], [Van Wagner, C.E., 1977], [Van Wagner, C.E., 1993]),
3. Accélération du feu [Alexander et al. 1992],
4. Humidité du combustible ([Rothermel et al. 1986]),
5. Sautes de feu ([Albini, F.A., 1979b]).

Afin de pouvoir lancer une simulation, les fichiers suivants sont indispensables :

1. Landscape (.LCP): il s'agit d'un fichier de métadonnées, qui contient les liens vers les données matricielles de rasters d'altitude, de pente, d'aspect de modèle de combustible et du couvert forestier.
2. Au moins un fichier de climat (.WTR) au format texte délimité, qui contient les taux d'humidité, les températures et les précipitations à plusieurs heures de la journée.
3. Au moins un fichier de vent (.WND) au format texte délimité, qui contient la vitesse et la direction du vent à plusieurs heures de la journée.
4. Un fichier « adjustment » (.ADJ) au format texte délimité, qui contient des paramètres personnalisés définis par l'utilisateur pour la propagation du feu.

5. Et un fichier « initial fuel moisture » (.FMS) au format texte délimité, qui définit les taux d'humidité de chaque type de combustible au début de la simulation.

FARSITE produit des sorties sous forme des fichiers SIG. Ces fichiers sont des cartes matricielles (Raster Output Maps) qui contiennent différentes informations sur la dynamique du feu comme les temps d'arrivée, l'intensité du feu, ou des polygones qui décrivent les contours du front de flamme [S. Kanga et al., 2014].

Comme nous pouvons le constater, tous les fichiers qui sont utilisés par le simulateur FARSITE sont soit au format SIG, soit dans un format de texte. C'est également le cas de la plupart des outils de simulation disponibles.

Sur la Figure 1.11, on dispose de sorties utilisables sur les plateformes telles que Google Earth. Ces résultats font intervenir la dynamique du feu sous la forme de plusieurs fronts de flamme en fonction du temps. Dans cet exemple, la surface en vert donne accès au front du feu une heure après son déclenchement, celle en rouge correspond au front après la deuxième heure.

Source: <http://www.redfish.com/wildfire/fireScenarios/HydeParkFire.jpg>.

4.2.1. ForeFire

ForeFire est un outil de simulation développé et testé par l'équipe « Feux de forêt » à l'Université de Corse [Laboratoire UMR CNRS SPE 6134] [Filippi J.-B., 2003]. C'est cet outil que nous utiliserons dans le cadre de la présente thèse.

Comme dans la plupart des simulateurs de feux de forêt, une simulation avec ForeFire nécessite des données pour l'initialisation telles que la date d'allumage du feu, la durée de simulation, les points ou le segment d'allumage [Balbi et al., 2009]. Ces données sont enregistrées dans un fichier de configuration au format XML qui permet également de référencer tous les fichiers nécessaires pour lancer une simulation [Filippi et al. 2010].

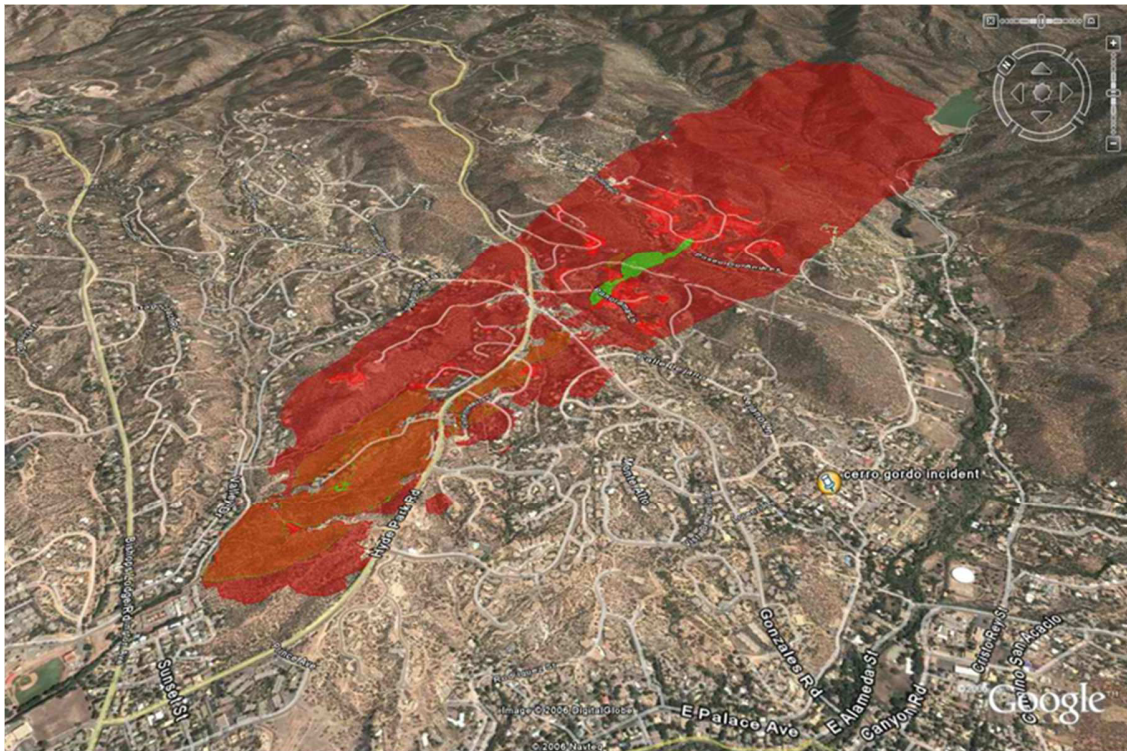


Figure 1.11. Résultat d'une simulation avec le logiciel Farsite

Ces fichiers sont :

- Le fichier Altitude (.asc) au format ascii, qui représente l'élévation de la zone.
- Le fichier (.shp) au format GIS (images vectorielles), qui décrit la distribution du combustible.
- Des fichiers de vent (.asc) de type texte, qui représentent la distribution locale du vent.

Les résultats de la simulation sont enregistrés dans des fichiers au format NetCDF ainsi que dans un fichier au format KML qui permet la visualisation et la vérification des résultats correspondant au dernier état du feu. À partir des données initiales, le logiciel est également en mesure de fournir des fichiers KML permettant de visualiser le contour final du feu observé afin

d'ajuster les comparaisons. L'environnement permet également de visualiser les différents types de combustibles.

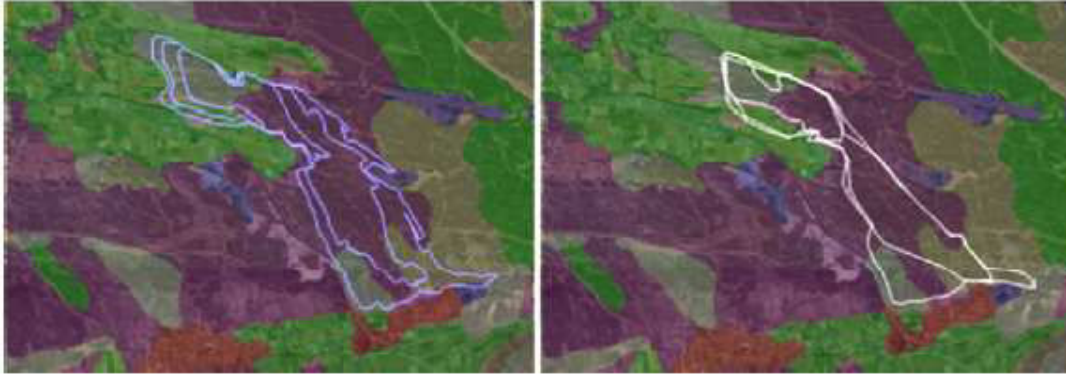


Figure 1.12. Observation et simulation avec ForeFire [Ayad H., 2011]

La Figure 1.12 présente les résultats obtenus dans le cas du feu qui s'est produit en 2005 à Lançon de Provence. Sur l'image de droite, nous présentons les résultats dus aux observations, sur l'image de gauche, les résultats de la simulation. Les différentes couleurs des images correspondent aux différents types de combustibles rencontrés par le feu au cours de la propagation. Comme dans le cas de FARSITE, nous pouvons avoir accès aux différents fronts en fonction du temps.

5. Méthodes de calcul de similarité

Dans cette section, nous présentons les principales méthodes statistiques du calcul de la similarité. La plupart de ces méthodes s'appliquent assez facilement à des échantillons mais doivent être adaptées lorsque, comme dans le cas des feux de forêt, il n'est pas possible d'obtenir directement des grandeurs physiques caractéristiques.

Les méthodes de calcul d'erreurs sont utilisées pour comparer la similarité entre des échantillons de données. Nous présentons quatre méthodes de calcul d'erreurs : l'indice de

similarité de Sorensen [Sorensen, 1948], le coefficient de similarité de Jaccard [Jaccard, 1901], l'indice de Kappa de Cohen [Cohen, 1960] et les rapports des aires et des périmètres [Fujioka 2002]. Dans la première partie, nous présentons ces différentes méthodes ; dans la seconde, nous donnons un exemple d'application de ces méthodes sur un ensemble de cas d'étude. Enfin, nous discutons des avantages et des inconvénients de ces méthodes.

5.1. L'indice de similarité de Sorensen

L'indice de similarité de Sorensen (ou coefficient de similarité de Sorensen) est une mesure statistique, développée par le botaniste Thorvald Sorensen et publiée en 1948 [Sorensen, 1948]. Cet indice calcule la valeur (le pourcentage) de la similarité entre deux échantillons. Il a été utilisé plus tard pour calculer l'erreur de simulation des incendies [Perry, 1999].

L'indice de similarité est considéré comme une méthode de calcul de concordance entre deux échantillons (ou ensembles). L'indice obtenu est lié au rapport entre le cardinal de l'intersection par le cardinal de la somme de deux ensembles. Le résultat est toujours une valeur entre zéro (0) et un (1). La valeur 1 représente une similarité parfaite (cas idéal), alors que la valeur zéro indique qu'il n'y a aucune similarité entre les deux échantillons.

L'équation pour calculer l'indice de similarité de Sorensen est :

$$Ind_{Sorensen}(t) = \frac{2|S^0(t) \cap S(t)|}{|S^0(t)| + |S(t)|}$$

Dans laquelle, t est le temps et $S^0(t)$ la surface observée au temps t et $S(t)$ la surface simulée au temps t.

5.2. Le coefficient de similarité de Jaccard

Le coefficient de similarité de Jaccard a été développé à l'origine par Paul Jaccard [Jaccard, 1901]. Ce coefficient est aussi une méthode statistique permettant de calculer la

concordance en deux échantillons comme l'indice de similarité de Sorensen. Le coefficient de Jaccard est très utilisé pour assumer le pourcentage de similitude entre deux ensembles.

L'indice de Jaccard est une simple comparaison. Il est calculé en divisant la valeur absolue de l'intersection entre les deux ensembles par leur union. Cela donne une valeur de similarité selon l'équation suivante :

$$Ind_{Jaccard}(t) = \frac{|S^0(t) \cap S(t)|}{|S^0(t) \cup S(t)|}$$

L'indice de Jaccard est interprété comme celui de Sorensen.

5.3. L'indice de Kappa de Cohen

Kappa de Cohen est une mesure statistique de l'accord, qui a été développée par Jacob Cohen et est parue dans *Educational and Psychological Measurement* en 1960 [Cohen, 1960]. Kappa mesure l'accord entre deux variables qualitatives pour des données appariées [Banerjee et al., 1999]. Elle est largement utilisée pour la simulation de modèles de distribution de végétation, par exemple dans [Prentice et al, 1992] [Prentice et al, 1993] et dans [Diffenbaugh et al., 2003].

Dans le domaine des incendies, le coefficient Kappa de Cohen a été utilisé par Bachisio Arca [Arca B. et al., 2007] pour l'estimation de l'erreur à partir du simulateur Farsite [Finney, 2004] pour des feux de végétation en Méditerranée.

Ce coefficient permet d'évaluer le degré d'accord entre n échantillons (dans la littérature, on trouve également les termes de juges ou d'observations ...) fait sur un échantillon de taille m. Les éléments d'une élévation peuvent appartenir à q catégories.

A partir de la matrice des résultats R[i,j], on obtient sans difficulté une matrice de proportion P[i,j].

Si l'on suppose que l'on a 2 évaluations notées A et B, q catégories notés $\{C_1 \dots C_q\}$, la matrice des résultats s'écrit :

$$\begin{array}{c} \text{Évaluation B} \\ \text{Évaluation A} \end{array} \begin{bmatrix} R_{1,1} & \dots & R_{1,q} \\ \vdots & \ddots & \vdots \\ R_{q,1} & \dots & R_{q,q} \end{bmatrix}$$

Dans cette matrice, $R_{i,j}$ correspond au nombre d'éléments de l'échantillon qui ont été classés dans la catégorie C_i par l'évaluation A, et dans la catégorie C_j par l'évaluation B. La taille n de l'échantillon est obtenue en sommant tous les éléments de la matrice.

L'indice de Kappa pour deux évaluations est donné par l'équation :

$$K(t) = \frac{P_a(t) - P_e(t)}{1 - P_e(t)}$$

$|\Omega|$ avec P_a la concordance observée donnée par l'équation :

$$P_a(t) = \frac{|S^0(t) \cap S(t)|}{|\Omega|} + \frac{|\Omega/S^0(t) \cup S(t)|}{|\Omega|}$$

Et P_e la concordance aléatoire donnée par l'équation :

$$P_e(t) = \frac{|S^0(t)| |S(t)|}{|\Omega|^2} + \frac{|\Omega/S^0(t)| |\Omega/S(t)|}{|\Omega|^2}$$

Si les évaluations sont totalement en accord, le coefficient de Kappa est égal à 1. Si elles sont totalement en désaccord (ou en accord dû uniquement au hasard), le coefficient de Kappa peut devenir inférieur ou égal à 0.

Les valeurs négatives se produisent lorsque l'accord est plus faible que prévu, mais cela arrive rarement.

Landis et Koch [Landis et al., 1977] ont proposé le Tableau 1.1 pour interpréter les valeurs de Kappa :

Valeur de Kappa	Interprétation
<0	Désaccord
0.0 — 0.20	Accord très faible
0.21 — 0.40	Accord faible
0.41 — 0.60	Accord modéré
0.61 — 0.80	Accord fort
0.81 — 0.99	Accord presque parfait
1.00	Accord parfait

Tableau 1.1. : Interprétation des valeurs de Kappa

5.4. Rapport des aires et rapport des périmètres

Ces deux méthodes sont renommées comme étant un moyen d'exprimer le pourcentage de similarité. Francis M. Fujioka est la première personne à avoir utilisé ces deux rapports entre les polygones (contours) des feux observés et simulés pour calculer un indice de similarité entre la simulation et l'observation. Il les a appliqués sur les résultats des simulations obtenues par Farsite [Fujioka, 2002].

Francis M. Fujioka a utilisé le rapport des aires (périmètres), en divisant l'aire de la surface brûlée sur la carte du feu simulé par celle du feu observé correspondant.

Mais dans certains cas, le rapport est supérieur à 1; surtout lorsque l'aire d'observation est plus grande que l'aire de la simulation. Aussi, pour obtenir une valeur normalisée entre 0 et 1, l'auteur propose de diviser l'aire d'intersection des feux observés et simulés avec l'aire de leur union.

Si le ratio est égal à 1, l'accord est parfait ; si le ratio est égal à 0, l'accord est mauvais.

Lorsque les formes sont irrégulières (cas le plus répandu), la méthode consiste à :

1. Diviser le feu de forme irrégulière en des formes régulières
2. Calculer l'aire de chaque partie régulière
3. Faire la somme des aires des parties régulières.

6. Conclusion

Ce chapitre a présenté une introduction simplifiée à la Théorie de la Modélisation et de la Simulation. Il a également détaillé les principes théoriques permettant de créer un cadre expérimental afin d'évaluer la modélisation d'un système en s'appuyant sur notre choix d'utiliser le formalisme DEVS. Ce dernier facilite la création et le développement des cadres expérimentaux pour les logiciels ainsi que l'intégration de modèles et leur réutilisabilité. L'intérêt de la formalisation et du développement des cadres expérimentaux est d'évaluer et de valider un modèle de simulation.

Nous avons présenté, dans la deuxième section, les méthodes généralement utilisées pour formaliser les observations incendies. Nous avons commencé par NetCDF qui est un conteneur particulièrement bien adapté au traitement de données scientifiques. Nous avons ensuite présenté la base de données Prométhée qui est utilisée pour générer les conditions initiales des feux simulés et obtenir les observations. Nous avons par la suite présenté deux logiciels existants permettant de simuler les incendies ainsi que les formats des données (entrées/ sorties) utilisées par ces logiciels. Enfin, nous avons présenté une bibliographie des méthodes de calcul de concordance qui permettent d'évaluer un feu simulé par rapport à un feu observé. Dans le chapitre suivant, nous ajouterons à ces méthodes deux nouvelles méthodes afin de calculer les erreurs de simulation de feux de forêt en prenant en compte leur dynamique temporelle.

CHAPITRE 2

CALCUL D'ERREURS DE SIMULATIONS

Dans le cadre de cette étude, l'évaluation des simulations des feux de forêt est réalisée en comparant la base de données d'un ensemble de feux observés avec la base de données des simulations. Cette évaluation est obtenue à partir de méthodes statistiques permettant d'évaluer un modèle en comparant les erreurs commises entre les résultats de la simulation et des observations. Les méthodes statistiques sont appliquées pour calculer la similarité ou la précision entre deux échantillons, mais elles ont un inconvénient majeur, car elles ne prennent pas en compte la dynamique des phénomènes étudiés. Or dans le cas d'un feu, il est non seulement primordial de savoir où se trouvera le front de flamme à un instant donné, ce que les méthodes statistiques font convenablement, mais également de savoir comment ce front se propage dans l'espace et dans le temps. Compte tenu de ces limitations, nous proposons deux nouvelles méthodes permettant de comparer des résultats avec les observations en y intégrant une notion qui prend en compte la dynamique des feux.

Avant de tester ces méthodes sur un simulateur tel que ForeFire, il est indispensable de comparer ces méthodes sur des cas idéaux afin de savoir dans quelles situations elles peuvent être appliquées. Nous décrivons, dans un premier temps, comment sont enregistrés les échantillons et quelle est la démarche que nous proposons pour obtenir les scores. Nous présentons ensuite les deux méthodes nommées Accord de Temps d'Arrivée (ATA) et Accord des Formes (SA) pour lesquelles l'aspect dynamique des feux est pris en compte. Les résultats et l'évaluation de ces méthodes sont présentés dans la publication parue dans *International Journal of Wildland Fire*, indexée au chapitre.

1. Calcul des erreurs

Afin de pouvoir comparer les échantillons, nous utilisons deux types d'enregistrement. Les échantillons à comparer sont composés soit de matrices contenant des uns (1) et des zéros (0), soit des images au format PNG composées de nuances de gris. Un code en Java a été développé pour traiter ces échantillons et leur appliquer les différentes méthodes statistiques présentées dans l'état de l'art.

Le type d'enregistrement est utilisé pour des scénarios dans lesquels n'intervient pas la dynamique du système, ou dans lesquels celle-ci n'est pas connue. Une matrice binaire correspond à la zone étudiée, chaque pixel correspondant à une cellule de la zone. Les (1) sont les cellules brûlées, les (0) celles non encore atteintes par le feu. La Figure 2.1 représente une matrice binaire d'un feu idéal ; le front de flamme peut être aisément repéré par les cellules qui contiennent une valeur 1 et ont un contact avec une cellule non brûlée ou ayant une valeur 0.

0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	1.00	0.00	0.00
0.00	0.00	1.00	1.00	1.00	1.00	0.00
0.00	0.00	1.00	1.00	1.00	1.00	0.00
0.00	0.00	1.00	1.00	1.00	1.00	0.00
0.00	0.00	0.00	1.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 2.1. Représentation matricielle d'un feu

Afin de pouvoir prendre en compte la dynamique des feux, nous utilisons un second type d'enregistrement à partir d'images au format PNG composées de différentes nuances de gris. Les différents niveaux de gris décrivent l'évolution du feu dans le temps. Le noir est utilisé pour le

front de flamme, le gris clair pour le point d'ignition. L'exemple de la Figure 2.2 correspond à un feu « idéal » qui s'est propagé depuis son point d'allumage en gris clair vers le contour final en noir. Si l'on connaît la date t_0 de départ du feu et la date t_f à laquelle la dernière cellule a été brûlée et que l'on normalise les niveaux de gris (1 pour le noir et 0 pour le blanc), alors si n est le niveau de gris normalisé d'une cellule C , on en déduit la date t_C à laquelle celle-ci a été brûlée.

$$t_C = t_0 + n * (t_f - t_0)$$

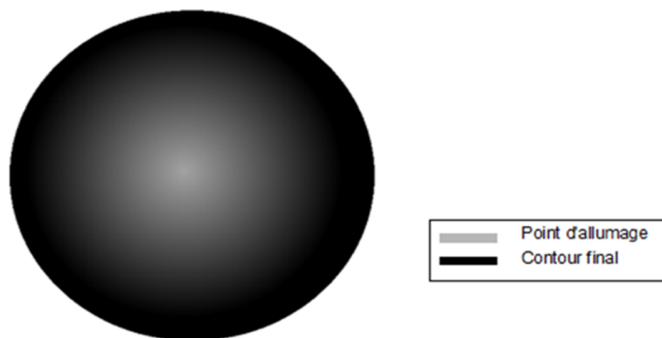


Figure 2.2. Représentation dynamique des feux

Les scénarios idéaux sont obtenus à partir de matrices binaires ou par leur expression analytique. Nous présentons sur la Figure 2.3 la démarche permettant de créer les scénarios idéaux et d'obtenir les scores d'erreurs. Deux scripts ont été développés : le premier script, **ScenarioCreator**, permet de produire les images qui seront ensuite utilisées par le second script, **PyFirescore**, pour calculer l'erreur.

À partir d'une matrice, nous appliquons un filtre gaussien pour obtenir l'image d'un feu. L'autre méthode consiste à utiliser les équations des formes connues pour dessiner un feu sous la forme d'un cercle, d'une ellipse, d'un carré ou d'un rectangle. Une fois les images obtenues, PyFirescore permet de calculer les erreurs en comparant deux images entre elles.

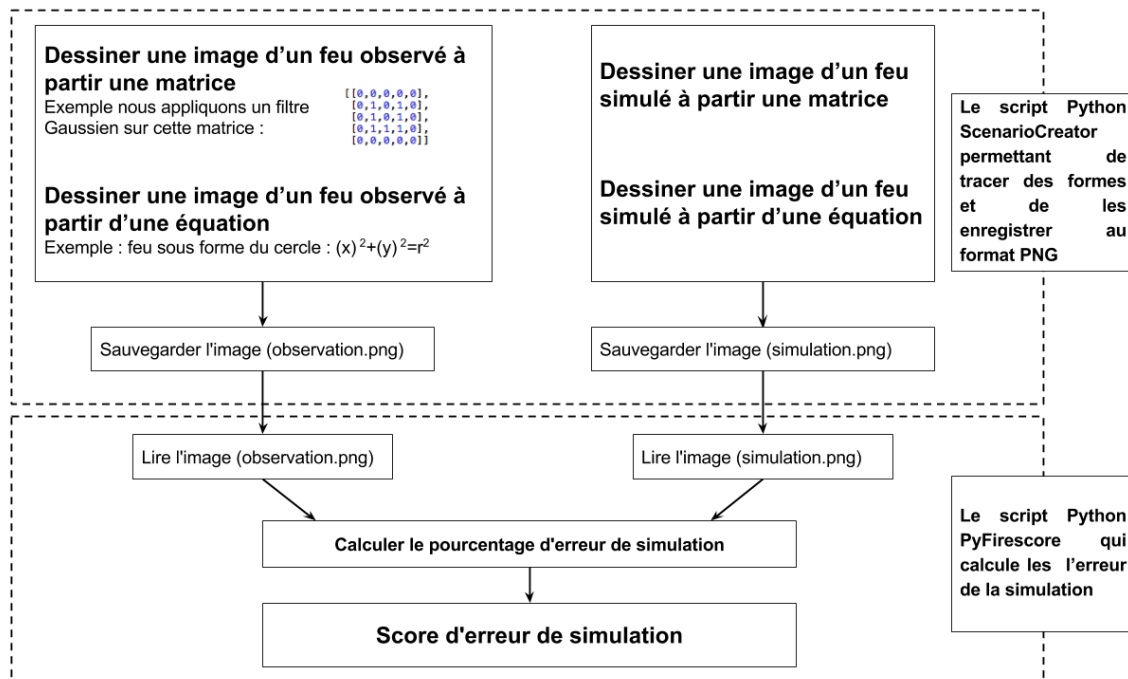


Figure 2.3. Processus de calcul des scores

2. Développement des scénarios idéaux

Les scénarios idéaux sont des formes simples comme les cercles, les ellipses ou les carrés. Bien que ces formes ne soient jamais rencontrées dans la réalité, elles permettent sans difficulté de calculer le niveau de concordance entre deux formes voisines. Les méthodes de calcul de similarité qui s'approcheront le plus de ces valeurs théoriques peuvent être considérées comme plus efficaces que les autres pour ce type de cas et seront donc plus efficaces dans des cas réels similaires.

Nous avons défini un jeu de scénarios en utilisant ces formes simples, mais en nous basant sur l'expérience acquise au cours des années en observant les différences entre les observations et les résultats que peuvent donner les simulateurs. Par exemple, il est courant de constater que la surface totale brûlée soit identique entre l'observation et la simulation, mais que les zones brûlées ne correspondent pas. Il arrive également que les formes soient identiques mais

que les tailles soient différentes ; ceci est généralement dû à des vitesses de propagation qui ne sont pas correctement estimées. Nous avons testé les métriques sur deux catégories de formes, (1) des formes régulières faciles à tester mais peu ou pas répandues et (2) des formes irrégulières composées de plusieurs formes régulières, plus complexes à évaluer mais plus proches de cas réels.

2.1. Formes régulières

Nous proposons ici d'utiliser quatre formes régulières permettant de valider les méthodes et de vérifier qu'elles concordent parfaitement avec les résultats attendus (Tableau 2.1). La première correspond à un cercle parfait pour lequel la simulation donne un résultat identique. La seconde correspond à une observation et une simulation s'effectuant dans des sens contraires, tandis que la dernière correspond à un scénario dans lequel la simulation donnerait un résultat identique mais avec un facteur de taille différent.

En appliquant les méthodes de concordance sur ces formes simples, nous avons pu vérifier que les résultats obtenus sont cohérents par rapport à ce que nous avons calculé analytiquement.

2.2. Formes irrégulières

Nous présentons maintenant une application des méthodes sur des formes plus complexes correspondant toujours à des cas idéalisés mais plus proches des résultats et des observations. Nous avons défini trois cas notés A, C et E correspondant à des « observations » et pour lesquels le feu s'est propagé transversalement sous la forme d'un ellipsoïde à partir du point d'ignition en blanc à l'intérieur de l'ellipsoïde. À la gauche de l'ellipsoïde, nous avons une zone non brûlée qui correspond, par exemple, à un obstacle. Nous proposons trois résultats théoriques notés B, D et F afin de calculer le niveau de similarité (Tableau 2.2).

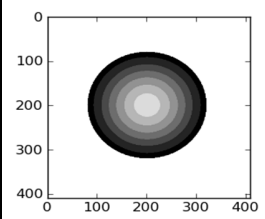
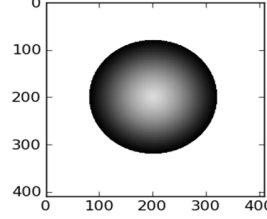
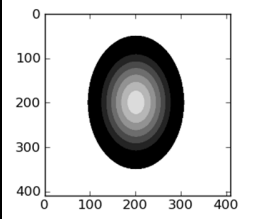
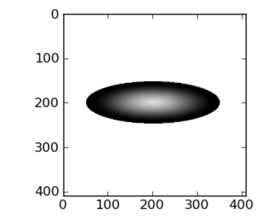
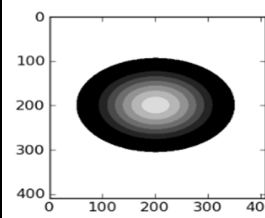
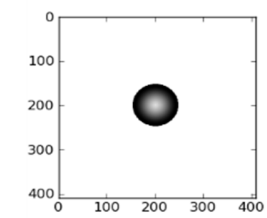
Observation	Simulation	Scores			
		Sorensen	Jaccard	Kappa	RA
		1.0	1.0	1.0	1
		0.469	0.307	0.351	NA
		0.248	0.14	0.18	NA

Tableau 2.1. Scores d'erreurs pour les formes régulières

Le premier scénario présente une simulation (B) parfaite par rapport à l'observation (A) et qui a la même forme ellipsoïde. Elle se propage de la même façon à partir de son point de déclenchement. La différence provient du comportement au niveau du point d'ignition qui, dans le cas de la simulation, brûle plus tôt que dans le cas proposé par l'observation. Cette zone a ainsi une couleur plus claire, ce qui montre que l'évolution et la dynamique du feu simulé sont différentes de celles de l'observation. Par contre, le contour final en noir correspond exactement à l'observation.

Pour le second scénario, la simulation (D) suit la même propagation et a la même forme que l'observation (C). Mais dans ce cas, tout l'ellipsoïde est consommé. Pour le troisième scénario, la simulation (F) est identique aux précédentes mais avec, cette fois, deux zones non brûlées sur chaque côté. Dans ces deux cas simulés, le comportement au niveau du point d'ignition est

identique à celui du cas B. Les nuances de gris pour la simulation représentent l'évolution du feu au cours du temps.

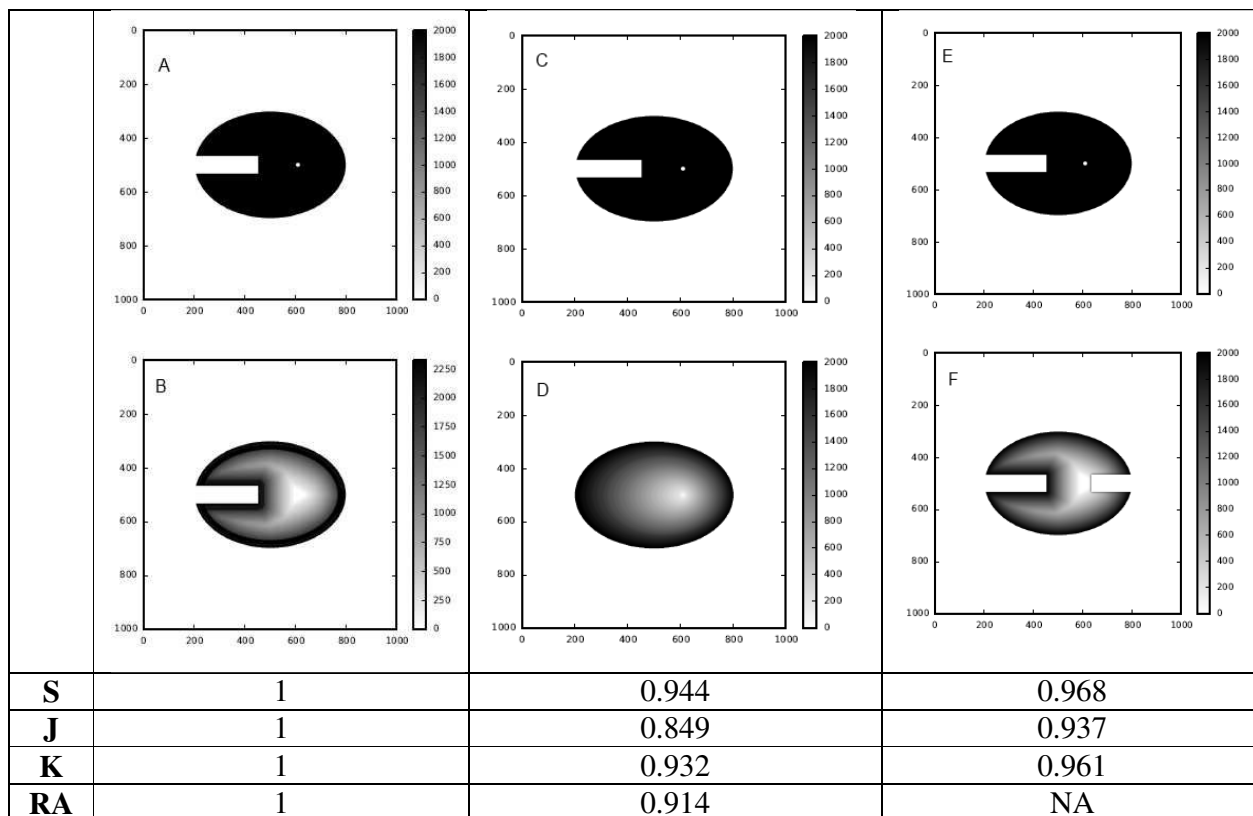


Tableau 2.2. Scores d'erreurs pour les formes irrégulières

Nous pouvons interpréter les scores comme suit :

- Pour le premier scénario, nous avons obtenu 1 partout puisque les contours finaux de l'observation et de sa simulation sont les mêmes. Pourtant, il y a bien une différence au niveau de l'évolution du feu. Cette évolution est illustrée par une teinte plus claire que la zone centrale de la simulation. Le problème vient du fait que les méthodes proposées ne font pas intervenir une comparaison d'un point de vue dynamique.
- Pour le deuxième scénario, nous avons obtenu des scores logiquement acceptés puisque le contour final de l'observation est différent de celui de sa simulation.

CHAPITRE 2 - CALCUL D'ERREURS DE SIMULATIONS

- Pour le troisième scénario, nous avons obtenu des scores plus faibles puisque le contour final est particulièrement déformé par rapport à l'observation. Aussi, la méthode de rapport des aires se révélant inadaptée à une étude systématique, elle n'a pas été prise en compte dans la suite de ce travail.
- Enfin, comme nous venons de le constater, les formes irrégulières mettent en évidence le fait que la méthode du rapport des aires est inadaptée dans le cas de l'étude des feux de forêt. Aussi, elle ne sera pas prise en compte dans la suite de notre travail.

Nous pouvons donc conclure de cet exemple que les indices de Sorensen, de Jaccard et les coefficients de Kappa sont des méthodes convenables permettant d'évaluer une similarité entre observation et simulation. Cependant, ces indices ne prennent pas en compte la dynamique temporelle dans les formules, puisque ces méthodes sont des méthodes de concordance des formes. Cet aspect peut ainsi être limitatif dans un contexte d'évaluation des modèles de simulation dans le cas des feux de forêt où la dynamique des phénomènes est primordiale. En effet, dans la réalité, les surfaces brûlées dépendent généralement de la topologie et de la nature du terrain et des conditions climatiques, leur évolution n'étant pas linéaire. Aussi, il est assez facile d'obtenir un score sur le contour final d'un feu. En revanche, il est bien plus difficile de discriminer des fronts de flamme sur leur dynamique par rapport aux observations.

3. Dynamique des phénomènes

Afin de déterminer une métrique permettant de représenter la dynamique du phénomène, nous proposons deux nouvelles méthodes de calcul des erreurs. Ces méthodes sont présentées en détail dans la revue *International Journal of Wildland Fire*, dans un article constituant le restant

de ce paragraphe. Ces deux méthodes, aussi résumées dans les deux paragraphes suivants, permettent de comparer les temps d'arrivée et la forme du feu afin de fournir un score d'erreurs lors de simulations, qui dépendra donc aussi de l'évolution du feu en fonction du temps.

3.1. L'Accord de Temps d'Arrivée (ATA)

L'Accord de Temps d'Arrivée (Arrival Time Agreement) est une méthode qui prend non seulement en compte les surfaces brûlées simulées et observées (notés S et S^0) mais également le temps dans les simulations et les observations (notés t et t^0). Bien que, dans la plupart des observations, nous n'avons accès qu'à une seule valeur de t^0 , c'est-à-dire le temps correspondant à l'état final du feu t_f^0 et donc à la surface totale brûlée, nous proposons dans [Filippi et al., 2014] une méthode permettant de prendre en compte les évolutions successives du feu au cours du temps.

Dans cette partie, nous donnons uniquement la formule dans le cas où l'on n'a accès, pour le feu observé, qu'à la surface totale brûlée et au temps d'arrivée du feu (t_f^0).

Dans ce cas particulier, le score ATA se compose de trois intégrales : la première compare les temps d'arrivée sur les zones communes, notées comme ayant brûlé aussi bien par l'observation que par la simulation. La seconde et la troisième comparent les différences entre les zones brûlées dans l'une et pas dans l'autre.

$$ATA = 1 - \frac{1}{|S(t_f) \cup S^0(t_f^0)| \max(t_f, t_f^0)} \left[\int_{S(t_f) \cap S^0(t_f^0)} \max(\tau(t) - \tau^0(x), 0) dt + \int_{S(t_f) \setminus S^0(t_f^0)} \max(t_f^0 - \tau^0(t), 0) dt + \int_{S^0(t_f^0) \setminus S(t_f)} \max(t_f - \tau^0(t), 0) dt \right]$$

3.2. L'Accord des Formes (SA)

L'Accord de Forme (Shape agreement) – comme l'ATA – prend en compte la dynamique de la simulation. Nous donnons ici la forme générale de l'équation :

$$SA = 1 - \frac{1}{t_f} \left[\int_{]0,t_f^o]} \frac{|S(t) \setminus S^o(t_f^o)|}{|S(t)|} dt + \int_{]0,t_f^o]} \frac{|S^o(t_f^o) \setminus S(t)|}{|S^o(t_f^o)|} dt \right]$$

Cette méthode est calculée en fonction du temps final des feux simulés (noté t_f) et des deux intégrales qui sont composées des surfaces brûlées simulées et observées.

Sur l'exemple suivant, nous calculons les scores ATA et SA correspondant à deux représentations matricielles pour un feu observé et un feu simulé. Dans le cas du feu observé, si nous n'avons accès qu'au temps final, la matrice est composée de 1 et de 0 correspondant aux cellules brûlées et non brûlées. Toutes les cellules sont donc supposées avoir brûlé au même instant t_f^o .

4. Publication

Une description formelle des différentes méthodes de calcul d'erreurs implémentées dans le cadre de cette étude sont présentées en détail dans un article publié dans l'*International Journal of Wildland* en 2013. Dans cet article, nous appliquons les méthodes à un panel de cas de formes régulières et au cas d'un feu réel. « Representation and evaluation of wildfire propagation simulations Filippi Jean-Baptiste, Mallet Vivien, Nader Bahaa (2014). *International Journal of Wildland Fire* 23, 46–57.<http://dx.doi.org/10.1071/WF12202> ».

Representation and evaluation of wildfire propagation simulations

Jean-Baptiste Filippi^{A,D}, Vivien Mallet^{B,C} and Bahaa Nader^A

^ASciences Pour l'Environnement (SPE) and Centre National de la Recherche Scientifique (CNRS), University of Corsica, BP 52, F-20250 Corte cedex, France.

^BInstitut National de recherche en informatique et en automatique (INRIA), BP 105, F-78153 Le Chesnay cedex, France.

^CCentre d'Enseignement et de Recherche en Environnement Atmosphérique (CEREA) (Ecole des Ponts ParisTech, Électricité De France (EDF) R&D; université Paris Est), 6–8 avenue Blaise Pascal, Cité Descartes Champs-sur-Marne, F-77455 Marne-la-Vallée, France.

^DCorresponding author. Email: filippi@univ-corse.fr

Abstract. This paper provides a formal mathematical representation of a wildfire simulation, reviews the most common scoring methods using this formalism, and proposes new methods that are explicitly designed to evaluate a forest fire simulation from ignition to extinction. These scoring or agreement methods are tested with synthetic cases in order to expose strengths and weaknesses, and with more complex fire simulations using real observations. An implementation of the methods is provided as well as an overview of the software package. The paper stresses the importance of scores that can evaluate the dynamics of a simulation, as opposed to methods relying on snapshots of the burned surfaces computed by the model. The two new methods, arrival time agreement and shape agreement, take into account the dynamics of the simulation between observation times. Although no scoring method is able to perfectly synthesise a simulation error in a single number, the analysis of the scores obtained on idealised and real simulations provides insights into the advantages of these methods for the evaluation of fire dynamics.

Additional keywords: error, notation, score, scoring method.

Received 30 November 2012, accepted 30 May 2013, published online 9 October 2013

Introduction

Model evaluation usually requires comparing predicted to observed values and is critical to establish potential errors and credibility (Appel *et al.* 2011). Forest fire propagation models have been developed for over 60 years. Originally these models provided a scientific method to determine a propagation speed that could easily be compared with observed rate of spread. The comparison only involved simple statistics. With the availability of computer simulation, fire growth models became available and required new scoring methods. A few specific or widely recognised mathematical or statistical methods were developed to evaluate their results (Fujioka 2002). However, defining scoring methods for such models is not straightforward, as there is a strong diversity in model's intents and scales, as well as diversity in available observations.

Depending on their purpose, the forest fire models have been built to predict either the stationary zero-dimensional propagation speed (Rothermel 1972) or the full fire front evolution over time (Linn *et al.* 2002; Finney 2004). This paper focuses on models that can provide results at the scale at which the phenomenon is most usually observed (several hectares), e.g. Firestation (Lopes *et al.* 2002), Farsite (Finney 2004), FireFire (Filippi *et al.* 2010) and many others that may be

found in Sullivan (2009). It is important to note that using a scoring method will definitely not help to find the best model. All of these models have some kind of parameterisation and are sensitive to the quality of input data (e.g. wind, fuel). This set of methods can help to better estimate these parameters when observation is available. Error methods are also required to estimate a confidence level in case of perturbed ensemble simulations or model ensemble simulations.

Many scoring methods are readily available in the literature; most of these methods come from the field of image analysis or geophysics. A few of them have been specifically developed for fire models (Fujioka 2002). This paper focuses only on those that have been applied to forest fire simulation evaluation. If all these methods succeed in computing an error, they sometimes disagree and most of them are built to compare an observation to a simulation result at a given observation time, without taking into account the dynamical aspect of the observed and simulated fires. Moreover, a score is only applicable if it is compatible with operational forecast and takes into account the observation data available in that context.

Some reviews and investigations have been proposed for evaluation methods (Fujioka 2002), synthesised and extended by Finney (2000) in an internal report for the United States

Department of Agriculture. The main conclusion is that it is very difficult (if not impossible) to synthesise simulation error in a single number and that a visual analysis provides a much better representation of the models performance for the trained analyst. We agree that trying to provide a model score for a limited set of observations of fire occurrence is not relevant to estimate model performance. Nevertheless, and much like Finney (2000), we believe that an analysis of a large dataset may be appropriate if it is possible to identify the necessary data and develop quantitative and objective tools to carry out the testing.

This paper provides a formal mathematical representation of a wildfire, reviews existing scoring methods using this formalism, proposes new methods and presents the associated tool that may be used to compute these scores. Using the proposed software, these methods are applied to synthetic simulation cases that are designed to expose their strengths and weaknesses. The methods are then tested for the simulation of two Mediterranean fires using observations that were available for the event. Finally, an overview of the software package is given, along with some recommendations about what method may be best appropriate to compare wildfires and to estimate simulation error.

Methods

Representation of a fire front and notation

This section focuses on the mathematical representation of the state of a firespread model seen as a dynamic model. In addition to the model's state, other data is manipulated to carry out a wildfire simulation, such as elevation, fuel distribution and wind maps, but these datasets are neither a direct observation of a wildfire or prognostic results of the simulation. Fuel evolution over time (burned, suppressed, water quantity), fire area and fire surfaces are the typical data found in existing wildfire observation databases, even if available instruments do limit the quantity and quality of information that is available. After analysing the French Promethee database, the European EFFIS database and fire reports, one can note that different information levels, types and formats can exist in different forest fires data sources. A formal, mathematical representation of a forest fire must represent all this available information that can be divided in four levels.

For the simplest data, each event is composed of a time of occurrence, a scalar expressing the total area burned and an imprecise localisation (a location name or reference index in a low resolution grid used by firefighters). For the second level, the data are composed of one or many accurate ignition points, an ignition time and the final burned surface in the form of a polygon. In the third level of data, it is possible to find timely information about the evolution of the fire front over time, essentially the global fire perimeter over time, but also the time at which the fire reached specific locations (such as a road, a house, a ridge). Finally the most detailed data contains information about the actions performed by the firefighters and specific local information (e.g. flame height, intensity and spot-fire). Reports also often contain information about the wind, temperature and moisture evolution, but even if this information is important to simulate a wildfire, such information does not constitute a direct result of a wildfire propagation simulation to be evaluated.

No systematic file format exists for these datasets, but the relative simplicity of the first two levels make them easy to manipulate and transform. The data with the best quality and in

the largest quantity is usually available in reports from the firefighters, with images and maps that must be manually processed in order to be digitally manipulated.

Mathematical notation required to formalise scoring methods must be able to represent the most complex available information.

Fire representation

Let t be the time and X the spatial position. We introduce the fuel consumption $\alpha(t, X)$ which is the ratio between the fuel mass available at t and position X , and the fuel mass initially available at X . It is set to 0 wherever no fuel has been burned yet, and to 1 where all of the fuel has been consumed. At locations where the fire is active, $\alpha(t, X)$ is between 0 and 1, depending on what proportion of fuel has been burned, relative to the amount of fuel initially available. For example, $\alpha(t, X)$ is 0.8 when 80% of the fuel initially available at X has been burned. Note that $\alpha(t, X)$ may also be in $]0, 1[$ at locations where the fire is not active anymore but did not burn all fuel. In locations where no fuel is available, no combustion can take place and $\alpha(t, X)$ is set to 0, for any t .

The front $F(t)$ is defined as the closure of the region where the fuel is being burned at t : $F(t) = \overline{\{X | \alpha Y(t, X) > 0\}}$ where $\alpha Y(t, X)$ is the time derivative of $\alpha(t, X)$. The burned area is defined as $S(t) = \{X | \alpha(t, X) > 0\}$. The first-arrival time of the front at X is set to $+\infty$ if the front never reaches X , otherwise it is set to $\tau(X) = \inf\{t | \alpha(t, X) > 0\}$.

We identify the observed values with the exponent (superscript) o . For instance, the observed burned area at t is $S^o(t)$.

Model's state

In order to describe the full state of the model, the fuel consumption is needed as well as its time derivative that identifies the regions where the fire is active. The state of the system is thus defined as $s(t, X) = (\alpha(t, X), \alpha Y(t, X))$.

Additional notation

We note $|S|$ the area of the surface S . Using the Heaviside function H , defined so that $H(x) = 0$ if $x \leq 0$ and $H(x) = 1$ if $x > 0$, we have $|S(t)| = \int H(\alpha(t, \cdot))$. We denote ∂S the boundary of S , and $|\partial S|$ the length of the curve ∂S .

We denote Ω the complete domain of interest. We assume that $S(t), S^o(t) \subset \Omega$ for any time t .

The simulation is run from t_0 to t_f , and the observed fire is active from t^o_0 to at most t^o_f where it is observed that the fire has stopped – note that the fire might have stopped earlier. Depending on the available information, t^o_f may not be available. The final simulation time used to carry out the simulation can be: (1) the time t^X_f at which the simulated fire arrived to self-extinction – this is the only final time that can be used to forecast the final burned area, when observations are still unknown; (2) the time t^o_f when the final observation is taken; (3) the time t^o_f at which the area of the simulated fire equals the area of the observed burned surface or (4) the time t_f at which the simulated fire has the best agreement with the observations.

Review of error evaluation methods

Visual comparison has traditionally been the main means of comparing observed and simulated fire patterns. It appears that it is more relevant to compare fire front shapes over time than comparing fields because the phenomenon is active only at the

interface between burned and unburned fuel. Because of this specificity, most methods applied to forest fire evaluation come from the fields of image analysis and geospatial statistics. Far from being exhaustive, this review tries to present several methods that have been employed to study forest fires.

Sørensen similarity index

Sørensen similarity coefficient (or Sørensen similarity index) is a statistical index, introduced in botany by Sørensen (1948). It computes the value (portion) of similarity between two samples. Perry *et al.* (1999) used this index to assess the agreement of fire simulation with observation.

Sørensen index specifically calculates the degree of inter-agreement between two sets. Here the two sets are the simulated and observed burned areas. Intersection of the two burned areas is divided by total burned areas. The result is a value between 0 and 1; 1 means a perfect agreement between observation and simulation, and 0 means there is no agreement. Sørensen similarity index is:

$$S = \frac{2|S^o(t) \cap S(t)|}{|S^o(t)| + |S(t)|} = \frac{2 \int H(a^0(t, \cdot) a(t, \cdot))}{\int H(a^0(t, \cdot)) + \int H(a(t, \cdot))} \quad (1)$$

Score range: [0,1]
Best score: 1.

Jaccard similarity coefficient

Jaccard similarity coefficient was originally developed by Jaccard (1901). It is a statistical index similar to the Sørensen index.

Jaccard's index is also a straightforward comparison method. The value is defined as the area of the intersection divided by the area of the union of the two sample sets (simulated and observed burned surfaces). The value ranges between 1 and 0, where 1 means perfect similarity between the two sets and 0 means disagreement. Jaccard similarity coefficient is:

$$J = \frac{|S^o(t) \cap S(t)|}{|S^o(t) \cup S(t)|} = \frac{2 \int H(a^0(t, \cdot) a(t, \cdot))}{\int H(a^0(t, \cdot)) + \int H(a(t, \cdot))} \quad (2)$$

Score range: [0,1]
Best score: 1.

Kappa statistics

Kappa coefficients, which are statistical measures of agreement, have been developed by Cohen (1960). They have gained widespread use in assessing model-simulated vegetation distribution (Diffenbaugh 2003). In the field of forest fire error, kappa statistics have been used by Arca *et al.* (2007) for the estimation of the error of the Farsite simulator (Finney 2004) in a Mediterranean area. They have also been used to compare and detect changes in vegetation maps (Monserud and Leemans 1992).

Cohen's kappa coefficient is a statistical measure of inter-rater agreement for several categories and a method to classify the accuracy (Cohen 1960; Banerjee *et al.* 1999) that relies here on a cell by cell comparison between observed and simulated datasets to construct an error matrix. Kappa quantifies an overall agreement that is relative to the whole domain area minus possible random agreements (an overall probability that a region is either burned or unburned).

Table 1. Interpretation of kappa values

Kappa range	Interpretation
<0	No agreement
]0.0,0.2]	Slight agreement
]0.2,0.4]	Fair agreement
]0.4,0.6]	Moderate agreement
]0.6,0.8]	Substantial agreement
]0.8,1.0]	Almost perfect agreement.

Relative agreement between two compared maps is estimated by:

$$P_a = \frac{|S^o(t) \cap S(t)|}{|\Omega|} + \frac{|\Omega \setminus S^o(t) \cup S(t)|}{|\Omega|} \quad (3)$$

Random agreement is calculated as:

$$P_e = \frac{|S^o(t)||S(t)|}{|\Omega|^2} + \frac{|\Omega \setminus S^o(t)||\Omega \setminus S(t)|}{|\Omega|^2} \quad (4)$$

After the probability of agreement by change is removed, the kappa coefficient reads:

$$K = \frac{P_a - P_e}{1 - P_e} \quad (5)$$

If kappa equals 1, there is a perfect agreement; if kappa equals 0, there is no agreement between simulated and observed fire maps. Negative values occur when agreement is weaker than expected but this rarely happens. Following Landis and Koch (1977), the interpretation of kappa values is provided in Table 1.

Ratio of areas

This method is a ratio of selected areas between observed and simulated fire shapes. It was introduced by Fujioka (2002). It describes the accuracy of agreement between two raster maps. The ratio of the areas is the sum of ratios between observed and simulated burned sector areas. In simple cases (like an ellipsoidal fire), the sectors divide the burned surfaces in portions that originate from the ignition. Each sector spans the region between the angles h and $h+\Delta h$, where the angle h is defined from a reference direction (usually north) and Δh represents the angular length of the sector. If the observed area is larger than the simulated area this ratio is greater than 1, so it must be divided by the union of the two areas.

Although this method can be used to estimate simulation error, it is important to note that it has been primarily designed to estimate local error in order to dynamically fit simulation parameters and to enhance simulation results that are obtained while the fire is running.

Let $A_{\Delta h}h$ be the simulated burned area of the sector between the angles h and $h+\Delta h$. Let $A^o_{\Delta h}h$ be its observed counterpart. The ratio of the areas reads.

$$RA = \frac{\Delta h}{2\pi} \sum_{h=0, \Delta h, 2\Delta h, \dots, 2\Delta h - \Delta h} \frac{A^o_{\Delta h}(h)}{A_{\Delta h}(h)} \quad (6)$$

Score range: [0,+∞].
Best score: 1.

Note that the score can be equal to 1 even if the simulation is not perfect. Indeed, among the different angles, there can be compensations between ratios higher than 1 and lower than 1. Also note there may be large errors in directions where the front stayed close to the ignition point.

In case the burned surfaces are not convex, additional treatments should be carried out because the sectors are more complex to define. However, in this paper, we do not use these *ad hoc* corrections that depend on the simulated case that cannot be automated. As a consequence, the sectors all start from the ignition point.

Propositions of two new evaluation methods

The previous method may be automatically applied for simple shapes (ellipses or convex shapes) where a distance from the ignition point corresponds directly to a measure of the front propagation distance. Although this is usually true in constant fuel and topography conditions, this may not be the case with changing wind, complex terrain and heterogeneous fuels. Moreover, poor information about wind or ignition point might result in very similar, but rotated or translated shapes that may not necessarily represent a wrong behaviour of the simulation model. Kappa coefficients require the definition of an area of interest (evaluation domain), whereas others only take into account observed and simulated areas. The problem here is that there exists no formal way to determine the extent of the evaluation domain. As pointed out in Finney (2004), a simulation is also successful if it has effectively reproduced the area where fire has not been propagated (and observed).

Most previous methods compare an observed fire to a simulated fire at the same time (i.e. the observation time) (Arca et al. 2007). This creates a large dependency of the error on the observation time, which is often not precisely evaluated – even the fire duration is uncertain as it lies between an estimated fire ignition and some time when the fire is ‘fully extinguished’. In particular, these methods are unable to identify a simulation that would have provided a good or bad fire shape at intermediate times.

A general approach could be to keep the simulation running until the fire stops, and to compare final burned shapes, whatever the final observation and simulation times. The problem here is that fire simulation models are usually run with very poor information about the fire suppression events, and they often make use of stationary models for the rate of propagation (such as in the Rothermel model). Hence, there is no real final simulation time as simulation is likely to run until there is no more burnable fuel.

Another approach would be to compare the full simulation to one observation at some time, taking into account every intermediate state or step to provide a composite score. The arrival time agreement and shape agreement methods have been built to address these two issues. The first method is better suited to evaluate a simulation on a fully extinguished fire, whereas the second method may be better adapted to evaluate a running fire simulation.

Arrival time agreement

The arrival time agreement is based on the simulated arrival times (denoted $T(X)$ at point X) and the observed arrival times

$T^o(X)$. Nevertheless, $T^o(X)$ is usually unknown because observations are only available for a few times (sometimes just one, when the fire has stopped). When a burned surface is observed at time t^o , we set $T^o(X)$ to t^o at all points X in the burned surface. Hence $T^o(X)$ is an upper bound on the arrival time. If several burned surfaces are observed at different times, we set $T^o(X)$ to the minimum time at which X is known to be burned.

We define the score at time t^o , which takes into account only the observations known at t^o . In practice, one will often choose $t^o = t^o_j$, but the score is defined more generally for as.

$$ATA = 1 - \frac{1}{|S(t) \cup S^o(t^o)| \max(t_f, t^o) - t^o_0} \left[\int_{S(t) \cap S^o(t^o)} \max(T(X) - T^o(X), 0) dX + \int_{S(t) \setminus S^o(t^o)} \max(t^o - T(X), 0) dX + \int_{S^o(t^o) \setminus S(t_f)} \max(t^o - T^o(X)) dX \right] \quad (7)$$

Score range: [0,1]

Best score: 1.

The score is composed of three integrals. The first one is the discrepancy between simulated and observed arrival times at locations burned in the simulation and in reality. When the difference $T(X) - T^o(X)$ is negative, we cannot conclude that the simulated fire arrived too early because $T^o(X)$ is an upper bound on the arrival time; hence the maximum time taken between $T(X) - T^o(X)$ and 0. The second integral is for locations that were burned in the simulation but not in reality. This may be because of early burning, with an advance in time of at least $t^o - T(X)$. Similarly, the third integral is for locations that are not burned by the simulation at time t_f so that the delay is at least $t_f - T^o(X)$.

If we know that t^o is greater or equal to t^o_j , any point in $|S(t_f) \setminus S^o(t^o)|$ should never have been burned. In this case, we can replace the second term $\max(t^o - T(X), 0)$ with $|t_f - T(X)|$.

Shape agreement

When an observed burned area $S^o(t^o)$ is given for time t^o only, this observation provides some information about the other times: for times $t < t^o$, the area burned at t is included in $S^o(t^o)$, and for times $t < t^o$, the burned area at t includes $S^o(t^o)$. Consequently, whenever a simulation, whose exact dynamics are known, does not satisfy these conditions this should lead to some penalisation. Following this idea, we introduce the shape agreement over time that accumulates errors in time for the misplaced burned areas in the simulation:

$$SA = 1 - \frac{1}{t_f + \max(t_0, t^o_0)} \left[\int_{\max(t_0, t^o_0)}^t \frac{|S(t) \setminus S^o(t)|}{|S(t)|} dt + \int_{t^o, t_f} \frac{|S^o(t^o) \setminus S(t)|}{|S^o(t^o)|} \right] \quad (8)$$

The first term corresponds to the area burned in the simulation but not burned in reality until t^o . The second term corresponds to

unburned areas in the simulation that are known to be already burned at t^o in reality.

Score range: [0,1]

Best score: 1.

This score can be extended for the case where there are several observations at different times $t^o_{(1)}, \dots, t^o_{(n)}$. If we denote in addition $t^o_{(0)} = \max(t_0, t^o_0)$ and $t^o_{(n+1)} = t_f$, then the score reads.

$$SA = 1 - \frac{1}{T} \sum_{i=1}^n \left[\int_{|t^o_{(i-1)}, t^o_{(i)}|} \frac{|S(t)| \setminus S^o(t_{(i)})}{|S(t)|} dt + \int_{|t^o_{(i)}, t^o_{(i+1)}|} \frac{|S^o(t_{(i)})| \setminus S(t)}{|S^o(t_{(i)})|} dt \right] \quad (9)$$

where $T = t_{(1)}^o - t_{(0)}^o + 2(t_{(n)}^o - t_{(n-1)}^o) + t_{(n+1)}^o - t_{(n)}^o$.

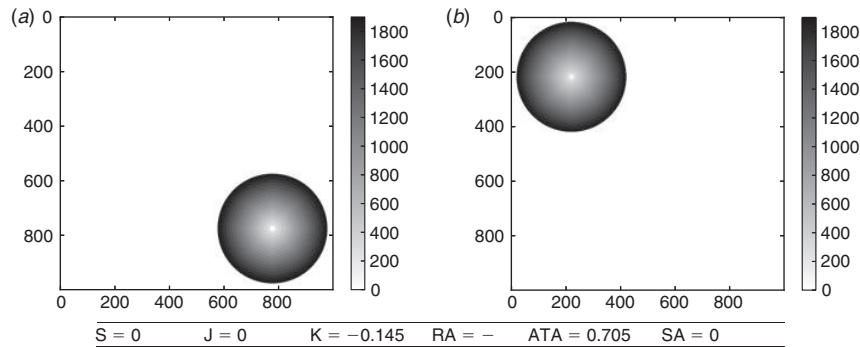


Fig. 1. Full error: the observed (a) and simulated (b) burned surfaces are disjoint. The grey scale represents the time in seconds, axes units are metres. Variables in the above table uses the following naming scheme: S for Sorensen similarity index, J for Jaccard similarity coefficient, K for kappa coefficient, RA for ratio of areas, ATA for arrival time agreement, SA for shape agreement.

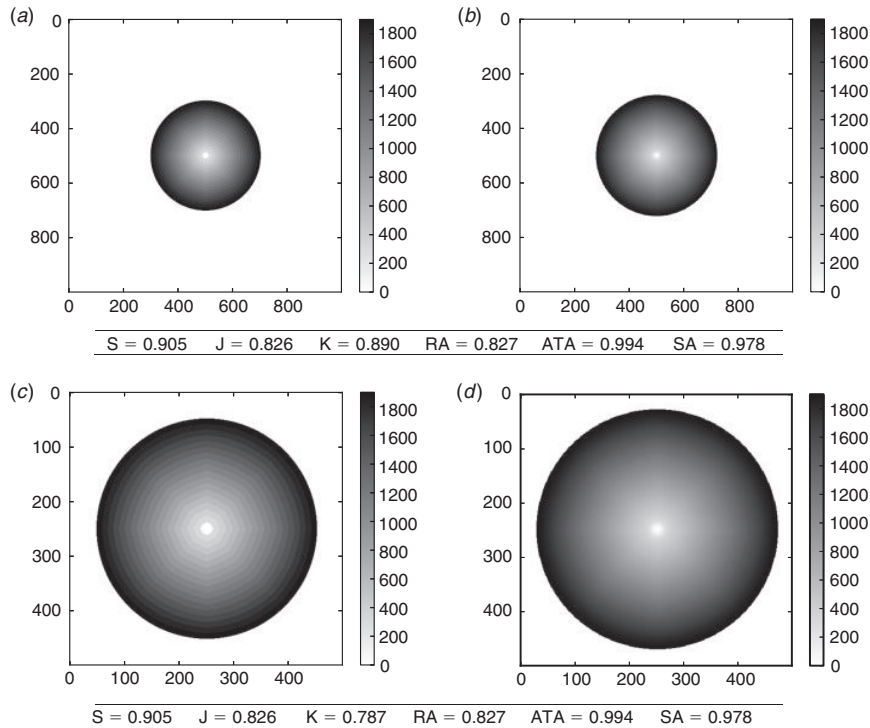


Fig. 2. Dependence on domain size: in both cases, the burned surface in the simulation (b,d) is larger than in the observation (a,c), and the ratio between the two areas is the same. But the part of the domain that is covered by the burned surfaces differs. The grey scale represents the time in seconds, axes units are metres. See Fig. 1 for the variable naming scheme.

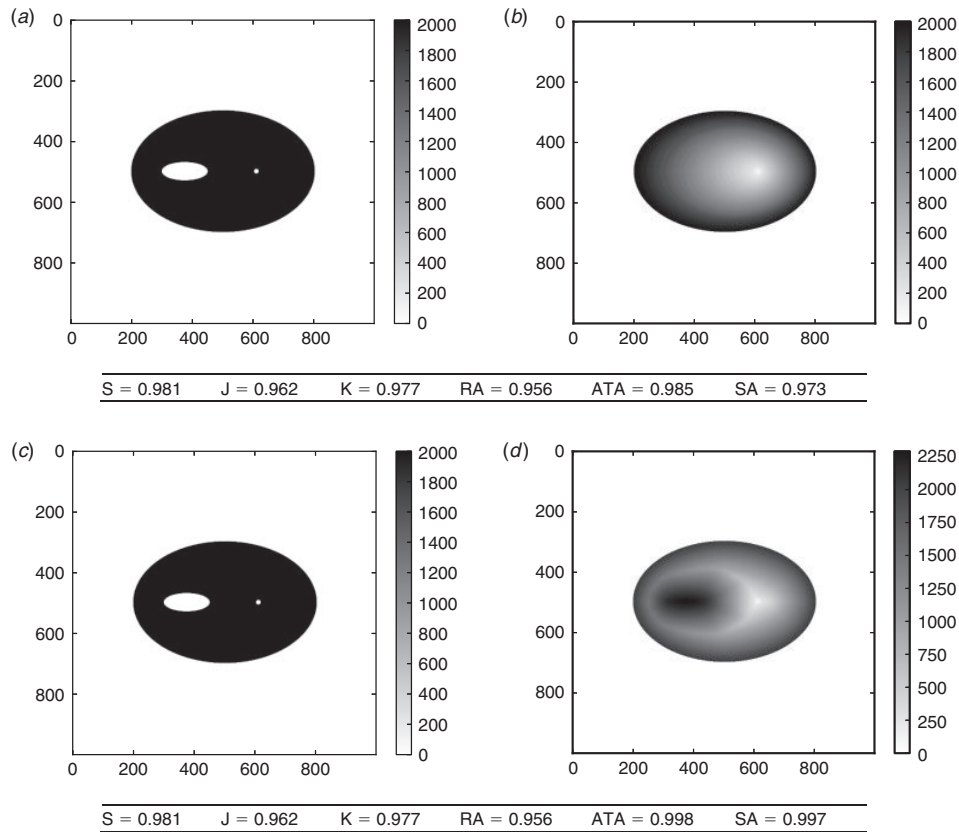


Fig. 3. Erroneous shape with different dynamics: in reality (a,c), the fire leaves an island of unburned fuel. The simulations burn this island. Nonetheless, the second simulation (d) burns the island later and shows a dynamics which is more likely than the first simulation (b). The grey scale represents the time in seconds, axes units are metres. See Fig. 1 for the variable naming scheme.

If the fire is known to have stopped at or before, it is possible to add a term to penalise the overburned areas like $\int_{]t_{(n)}^o, t_{(n+1)}^o]} |S(t) \setminus S^o(t_{(n)}^o)| / |S(t)| dt$.

Results and discussion

In this section, several tests are run in order to highlight the applicability, strengths and weaknesses of the different methods.

All tests are carried out using a description of the dynamics in the form of fields of arrival times. In Section, the methods are applied to evaluate simulations for two real forest fires. In Section, various synthetic cases are analysed so as to illustrate the behaviour of the scores in typical situations.

The results are illustrated with the maps of observed and simulated arrival times (Figs 1–7). In real cases, the observed arrival times are actually upper bounds on the arrival time as the burned surface cannot be observed continuously.

As the reader will notice, the scores magnitude of one method cannot be compared with the magnitude of another method. Also it is generally not possible to identify a threshold (for scores) that would indicate that a simulation is reliable or not. Instead, for one scoring method, the score values should be compared with

each other so as to determine which simulation is the best. In other words, it is hard to state whether a model better simulated one fire than another, but a scoring method should help to rank the simulations for a given fire.

Synthetic cases

In this subsection, unless stated otherwise, the final simulation time t_f is set to the final observation time t_f^o .

Full error

We consider the case where the simulation completely fails because the simulation ignition point is far from the real initial ignition point. The maps of arrival times and the score are reported in Fig. 1. Sørensen similarity index, Jaccard similarity coefficient and shape agreement are equal to zero and therefore clearly identify the simulation failure. The arrival time agreement is not zero, but it is very low compared with normal cases. The kappa coefficient is negative, which is consistent with the very poor agreement between the two burned surfaces. The ratio of areas could not be computed because the ignition points differ between the simulation and reality, so that the sectors could not be properly defined.

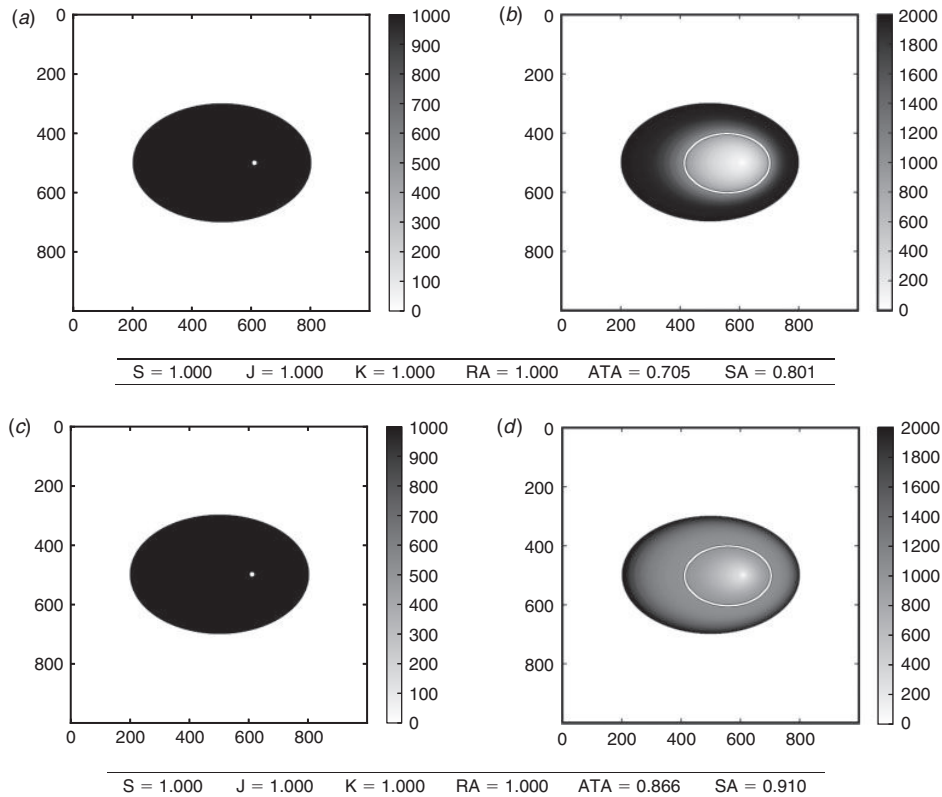


Fig. 4. Arrival times: the simulations are identical until the observed time. At t_f^o , they both reached the white curve, whereas the observed fire burned the whole ellipse (a,c) (note that the grey scale is not the same on the observation side and on the simulation side). After t_f^o , the simulations run until $t_f = \bar{t}_f^s$ when they both burned as much as the observed fire. The first simulation (b) takes more time to burn the remaining fuel, whereas the second simulation (d) is faster; therefore, closer to the reality. The grey scale represents the time in seconds, axes units are metres. See Fig. 1 for the variable naming scheme.

Dependence on domain size

In Fig. 2 we consider an isotropic fire propagation, with a simulation that is slightly faster than in reality. Two comparisons are made here, one with a large simulation domain and another with a smaller domain. We see that all scores remain identical, except the kappa coefficients which heavily depends on geographical distances in the map. One should consequently be cautious when using this indicator to compare the performance of different simulations that may not be run over the same domain.

Erroneous shape with different dynamics

In this test (Fig. 3) the observed fire has propagated as an ellipse leaving an island of unburned fuel. We compare this to two erroneous simulations that burn the island. The first simulation propagates as a simple ellipse, without any specific change of speed over the island. The second simulation has different dynamics as the front first avoids the island and then burns it in the end. It is possible that this simulation burned the island too early. However its dynamics are more likely than that of the first simulation.

In both cases, the Sørensen similarity index, the Jaccard similarity coefficient, the kappa coefficient and the ratio of areas

are identical. Indeed, they rely only on the final burned surfaces, which are the same in both cases.

On the contrary, the arrival time agreement and the shape agreement take into account the dynamics of the fire, and they can identify which simulation shows better agreement with reality. In the first simulation the island is burned early, which is highly inconsistent with the observations. The island is burned later in the second simulation, hence our two methods put less penalisation on the unduly burned island.

Arrival times

In this case (Fig. 4), the observed fire has propagated as a simple ellipse. The simulations also propagate as ellipses, but at the observed time they are both late; their burned surfaces are included in the observed burned surface. Until this time, the two simulations are associated with the same map of arrival times; indeed, their rates of speed have been identical.

Then, we let the simulations run so that $t_f = \bar{t}_f^s$, i.e. we let the simulations run until their burned areas are the same as the observed burned area. Right after t_f^o , the first simulation is slower than the second simulation. The second simulation fills most of the observed burned area rather fast, whereas the first

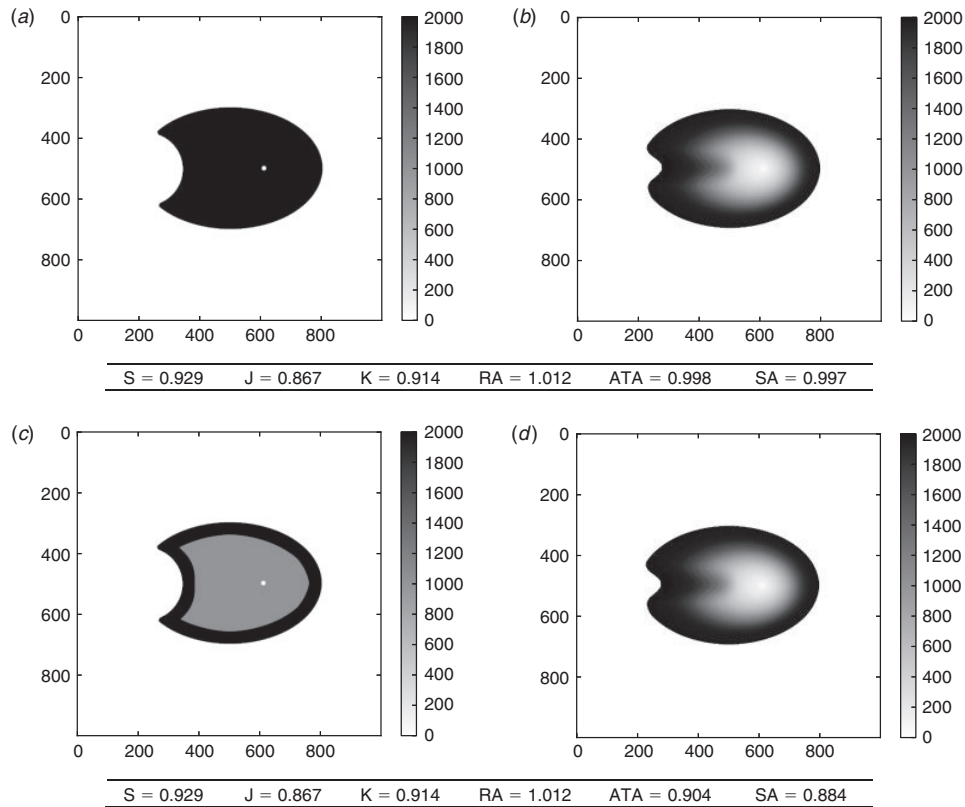


Fig. 5. Use of rich observations: the same simulation (*b,d*) is compared with two different observation sets (*a,c*). The first observation set (*a*) only contains the final burned surface, whereas the second observation set (*c*) includes the burned surface at an intermediate time. The grey scale represents the time in seconds, axes units are metres. See Fig. 1 for the variable naming scheme.

simulation takes more time. At the end of the simulation period, the first simulated fire expands faster compared with the second fire. Both simulated fires have burned the observed surface at the same final time t_f . The difference is that, for a long time after t_f , the second simulation has burned a larger area than the first simulation. Consequently, the second simulation is closer to the reality during this period. Its arrival time agreement and shape agreement are thus better than those of the first simulation.

On the contrary, the other scores fail to identify the best simulation. This is because of the fact that the two simulations are the same at t_f and perfectly match the observed area.

Use of more observations

In Fig. 5, we compare the same simulated fire to two different observation sets of the same fire. The first observation set only contains the final burned surface and its time t_f . The second observation set contains, in addition, one burned surface at an intermediate time.

The Sorensen similarity index, the Jaccard similarity coefficient, the kappa coefficient and the ratio of areas always give the same result as they are solely based on the final burned surface. On the contrary, the arrival time agreement and the shape agreement show different results that allow identification of the errors at the intermediate time. When compared to the

final burned area, the final simulated area is erroneous, hence the scores are lower than 1. With the intermediate burned surface, we see that the simulated fire spread was first slower than in reality. In the observations, the fire expanded rather fast and then slowed down, which is not reproduced by the simulation. Consequently, introducing the intermediate observation should lower the evaluation scores, which is the case for the arrival time agreement and the shape agreement.

Real cases

This section presents two typical applications of the scoring methods. In the first application, the different scoring methods compare simulations with different stop conditions. In the second application, the results of models with different physical parameterisations are compared with observations consisting of burned surfaces at different times.

Contrary to previous results where both observations and simulations were synthetic, here the observations are field observations, and the simulations were carried out by the fire-spread model ForeFire (Filippi *et al.* 2010).

Suartone fire

The Suartone fire occurred in south-east Corsica on 28 July 2003 near the village of Suartone. About 456 ha were burned. The fire was detected at 1500 hours local time. This fire first spread

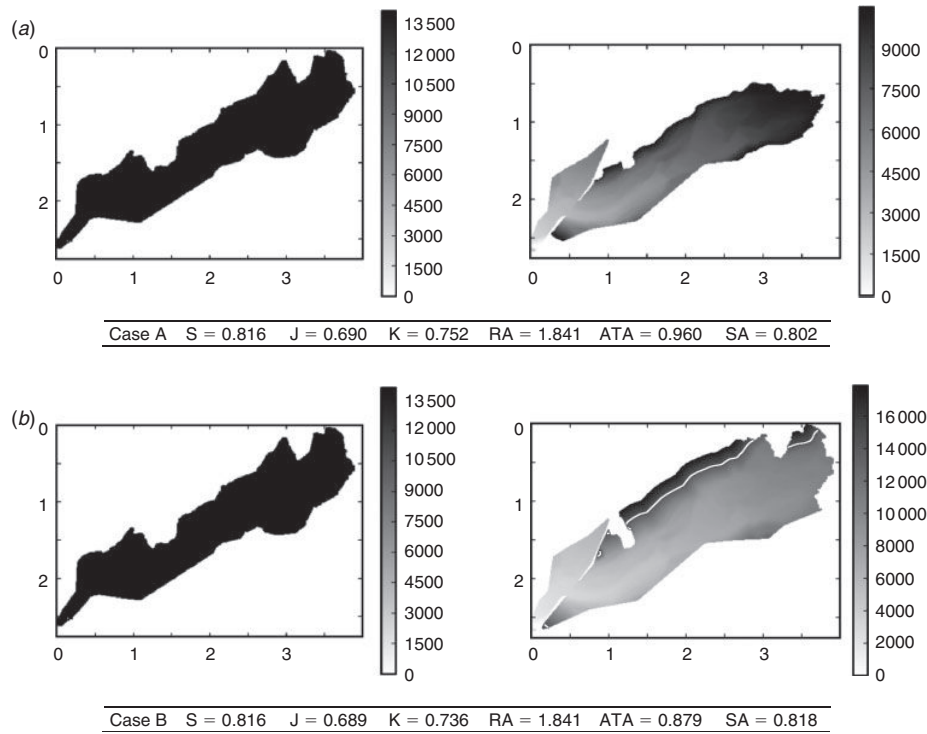


Fig. 6. Suartone fire: the fire burned 456 ha towards north-east. The first simulation (a) is run until the simulated fire has reached the same area as the observed fire whereas the second simulation (b) is continued until the virtual fire stops propagation. The white line corresponds to the contour at time t_f^o . The grey scale represents the time in seconds, axes units are kilometres. See Fig. 1 for the variable naming scheme.

moderately in an area surrounded by shrubs before it jumped a road, shifting on its right flank. As this flank was $\sim 350\text{--}400$ m wide, it became the fire head and accelerated driven by both a western wind and upslope effect. Finally, fire suppression action was taken on both flanks of the fire to control it and the fire ended its spread towards the sea ~ 1900 hours local time.

No quantitative information is available on the fire attacks and, depending on the model, the simulation may not extinguish the fire by itself. In this case, a condition must be found to stop the simulation. One option is to stop the simulation at the observed time, but this time may significantly maximise the actual fire extinction time. Most of the fire information available for reanalysis is in the same form, with a precise final contour but a final time that may be some time after the actual fire extinction.

In this test, two simulations were run from and with the same reference model, but until $t_f = \bar{t}_f$ (i.e. until the simulation has reached the same area as the final observed area) in case A, and until $t_f = \bar{t}_f^X$ in case B. The surface burned in case B is significantly larger than in case A. More details on the simulation settings for this case can be found in Santoni *et al.* (2011).

Sørensen, Jaccard and ratio of areas provide very similar, if not equal, scores in both cases because the effect on these scores of the overestimation in case B is equivalent to the effect of the underestimation in case A. Kappa coefficient gives a worse result for case B, because the overestimated area is large relative to the domain size. Likewise, arrival time agreement strongly penalises the overestimation in case B because the simulation burns significant areas long after the observed final time t_f^o . On

the contrary, shape agreement clearly favours case B as it strongly penalises the unburned area of case A after the early simulation end at $t_f = \bar{t}_f$.

Lançon-Provence fire

The Lançon-Provence fire took place in 2005 in the south of France and burned ~ 800 ha of shrubs and forest. On that day, a north-westerly wind was blowing, providing extreme propagation conditions. The fire started at 0940 hours local time and spread moderately until 1200 hours. Then the fire head gained intensity and propagated rapidly towards the south until it stopped ~ 1630 hours local time. In this case, the burned surface was recorded at different times during the ongoing fire, providing more observations than in the case of the Suartone fire. More details about the simulation setting for this case can be found in Filippi *et al.* (2010).

Several simulations of the Lançon-Provence case are run with different parameterisations. Case A is the reference simulation with the most likely (though highly uncertain) vegetation parameters for the day. In case B, the fuel height is divided by two, providing an overall slower propagation speed. In case C, the surface to volume ratio of dead fuel is multiplied by two, leading to a much faster propagation, in particular down wind. In case D, the surface to volume ratio is divided by two, constraining the fire to propagate down wind. The four simulations generated results with clearly different dynamics. All simulations were run until $t_f = \bar{t}_f^X$, so the simulation was expected to overestimate the burned area as fire attacks were not taken into

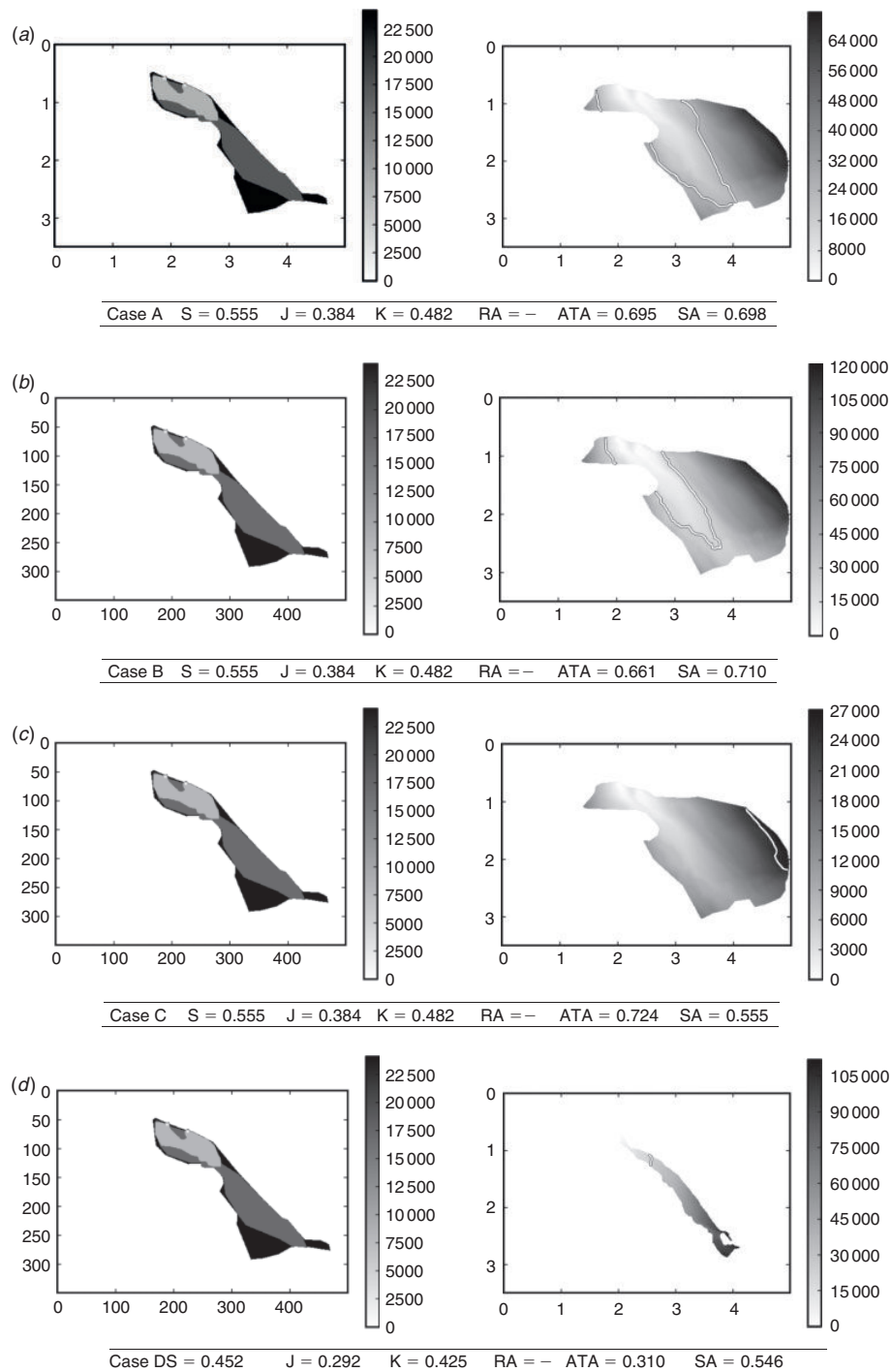


Fig. 7. Lançon-Provence fire: the fire expands towards south-east and burns 800 ha. Four simulations are run with (a) the reference parameters set, (b) fuel height half of the reference, (c) dead fuel surface to volume ratio double of the reference and (d) dead fuel surface to volume ratio half of the reference. The white lines correspond to the contour at time t_f^0 . The grey scale represents the time in seconds, axes units are kilometres. See Fig. 1 for the variable naming scheme.

```
simulation = NewForestFireFromNC("simulation_file.nc")
simulation_resolution = simulation.GetResolution()
simulation_domain = simulation.GetDomainShape()
observation = NewForestFireFromNC("observation_file.nc",
                                   resolution = simulation_resolution,
                                   domainShape = simulation_domain)

S = ComputeSorensenIndex(simulated, observed)
ATA = ComputeArrivalTimeDiscrepancy(simulated, observed)
SA = ComputeShapeAgreement(simulated, observed)
```

account in the simulation. In the first three cases, all burnable vegetation was consumed up to the point that the fire reached a non-burnable vegetation, resulting in an identical final burned area but with very different fire dynamics.

The ratio of area method was not able to provide a score for any of these simulations because of the convex shape of the fire. Applying the method would have required manual relocation of the calculation points at certain simulation steps. Table 2 synthesises the case ranks according to the scoring methods. Sorensen, Jaccard and kappa scores give the same results for cases A, B and C because the final areas are identical in these three cases. The three scores find that case D is the worst simulation. Sorensen and Jaccard penalise the simulation more than kappa. Shape agreement and arrival time agreement provide very different results. Arrival time agreement strongly penalises under-prediction, i.e. strongly penalising a fire that missed burned areas. Consequently, case C is marked as the best simulation, as it manages to never underestimate burned area (all simulations except C underestimated the burned area at the south of the fire). Case A is the next best score for arrival time agreement, followed by case B and a much lower score for case D that strongly underestimates burned areas.

Shape agreement method is a stronger marker of overall shape accordance and ranks case B as the best simulation, with a close case A. This ranking is due to the fact that the case B (slower) underestimations of the southern part of the fire are slightly lower than case A (quicker) overestimations of the northern part of the fire. Overall these two scenarios provide the shapes that are most similar to the observation at $t = t_f^o$. As shape agreement is equally penalising under- and over-prediction, the scores are about the same for cases D and C – case D under predicts about as much as C over predicts.

Software

The scoring methods have been implemented in a Python library with minimum dependencies. It is available at <http://sourceforge.net/projects/pyfirescore/> (accessed 12 July 2013).

Table 2. Cases ranked by score for Lançon-Provence fire

Rank	Sorensen	Jaccard	Kappa	Arrival time	Shape
1st	A/B/C: 0.555	A/B/C: 0.384	A/B/C: 0.482	C: 0.724	B: 0.710
2nd				A: 0.695	A: 0.698
3rd				B: 0.661	C: 0.555
4th	D: 0.452	D: 0.292		D: 0.310	D: 0.546

It relies on NumPy (Oliphant 2006) and SciPy (Jones *et al.* 2001). The base format for forest fire simulation and observation is NetCDF, with a fire data convention such as proposed by Nader *et al.* (2011). If the information is not directly available as a well formatted NetCDF file, it is possible to import data as bitmap image, points list or matrices in different file formats and encoding. Such imports can be scripted and can benefit from the numerous input/output libraries already available for Python. Below is a commented example of score computations with the library.

The previous lines first load simulated data from a NetCDF file. The spatial resolution of the simulation grid (i.e. the grid corresponding to the matrix of arrival times) and its extent are retrieved, so that the observations can be mapped to this grid. Once the simulation and the observations are loaded, a single call to a function carries out the score computation.

Conclusion

This paper has reviewed a set of evaluation methods and proposed two new evaluation scores, using formal notation. A software tool is provided along with the paper so that a Python implementation of the methods is available to the reader.

A set of synthetic cases was built in order to illustrate the differences and advantages or drawbacks of all methods. The paper stressed the importance of scores that can evaluate the dynamics of the model, as opposed to methods relying on snapshots of the burned surfaces computed by the model.

The analysis of scores obtained on idealised and real cases demonstrates some advantages of the dynamics-aware methods. However, it appears that no scoring method is able to perfectly synthesise a simulation error in a single number. The two proposed methods seem more appropriate if one wants to specifically evaluate the quality of the simulation dynamics. These methods can always be applied when the simulation arrival times or intermediate simulated fronts are recorded in the course of the simulation. Therefore, we recommend, whenever possible, to compute either the arrival time agreement or the shape agreement. In addition, these scores can be seamlessly applied with one final observed surface or with several intermediate observed burned surfaces. They always evaluate the whole simulation, from its ignition to its extinction.

The availability of efficient evaluation methods can help to couple simulation models and observations. There is indeed a need for sound comparison between model states and observations if one wants to apply data assimilation algorithms, like a Kalman filter (Beezley and Mandel 2008) or the four-dimensional variational assimilation.

The methods described in this paper are adapted to the evaluation of one model, for one simulation. However, a reliable evaluation of modelling system should involve several simulation cases and maybe an ensemble of simulations. This is especially true when dealing with large fires, propagating during several days, where the complexity of the propagation makes it very difficult for one model to represent the diversity of the events. The ensemble of simulations could be made of perturbed simulations with the same model or with simulations from different propagation models. This would require different comparison scores, but probably based on the dynamic-oriented approaches of this paper. These composite scores, relevant for probabilistic forecasts and risk assessment, should be the subject of future research.

Acknowledgements

This research is supported by the Agence Nationale de la Recherche, project ANR-09-COSI-006 IDEA. The authors thank Marc Finney who helped to define the problem and the anonymous reviewers for their comments on this work.

References

- Appel KW, Gilliam RC, Davis N, Zubrow A, Howard SC (2011) Overview of the atmospheric model evaluation tool (AMET) v1.1 for evaluating meteorological and air quality models. *Environmental Modelling & Software* **26**(4), 434–443. doi:10.1016/J.ENVSOF.2010.09.007
- Arca B, Duce P, Laconi M, Pellizzaro G, Salis M, Spano D (2007) Evaluation of {FARSITE} simulator in Mediterranean maquis. *International Journal of Wildland Fire* **16**(5), 563–572. doi:10.1071/WF06070
- Banerjee M, Capozzoli M, McSweeney L, Sinha D (1999) Beyond kappa: a review of interrater agreement measures. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique* **27**(1), 3–23. doi:10.2307/3315487
- Beezley JD, Mandel J (2008) Morphing ensemble Kalman filters. *Tellus* **60A**(1), 131–140.
- Cohen J (1960) A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* **20**(1), 37–46. doi:10.1177/001316446002000104
- Diffenbaugh NS (2003) Vegetation sensitivity to global anthropogenic carbon dioxide emissions in a topographically complex region. *Global Biogeochemical Cycles* **17**(2), 1067. doi:10.1029/2002GB001974
- Filippi J-B, Morandini F, Balbi JH, Hill DR (2010) Discrete event front-tracking simulation of a physical fire-spread model. *Simulation* **86**(10), 629–646. doi:10.1177/0037549709343117
- Finney MA (2000) Efforts at comparing simulated and observed fire growth patterns. USDA Forest Service, Rocky Mountain Research Station, Technical Report INT-95066-RJVA. (Missoula, MT)
- Finney MA (2004). Farsite: fire area simulator-model development and evaluation. USDA Forest Service, Rocky Mountain Research Station, Technical Report RMRS-RP-4. (Missoula, MT)
- Fujioka FM (2002) A new method for the analysis of fire spread modeling errors. *International Journal of Wildland Fire* **11**(4), 193–203. doi:10.1071/WF02004
- Jaccard P (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles* **37**, 547–579.
- Jones E, Oliphant T, Peterson P, et al. (2001). SciPy: open source scientific tools for Python. Available at <http://www.scipy.org/> [Verified 14 June 2013]
- Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. *Biometrics* **33**(1), 159–174. doi:10.2307/2529310
- Linn R, Reisner J, Colman JJ, Winterkamp J (2002) Studying wildfire behavior using FIRETEC. *International Journal of Wildland Fire* **11**(4), 233–246. doi:10.1071/WF02007
- Lopes A, Cruz M, Viegas D (2002) Firestation – an integrated software system for the numerical simulation of fire spread on complex topography. *Environmental Modelling & Software* **17**(3), 269–285. doi:10.1016/S1364-8152(01)00072-X
- Monserud RA, Leemans R (1992) Comparing global vegetation maps with the Kappa statistic. *Ecological Modelling* **62**(4), 275–293. doi:10.1016/0304-3800(92)90003-W
- Nader B, Filippi JB, Bigambiglia P-A (2011) An experimental frame for the simulation of forest fire spread. In 'Proceedings of the 2011 Winter Simulation Conference', 11–14 December 2011, Phoenix, AZ. (Eds S Jain, RR Creasey, J Himmelspach, KP White, M Fu) pp. 1010–1022. (IEEE)
- Oliphant TE (2006) 'Guide to NumPy.' (Trelgol Publishing: USA) Available at <http://www.numpy.org> [Verified 29 August 2013]
- Perry GLW, Sparrow AD, Owens IF (1999) A {GIS-supported} model for the simulation of the spatial structure of wildland fire, Cass Basin, New Zealand. *Journal of Applied Ecology* **36**(4), 502–518. doi:10.1046/J.1365-2664.1999.00416.X
- Rothermel R (1972) A mathematical model for predicting fire spread in wildland fuels. USDA Forest Service, Intermountain Forest and Range Experiment Station, Research Paper INT-115. (Odgen, UT)
- Santoni P, Filippi JB, Balbi J-H, Bosseur F (2011) Wildland fire behaviour case studies and fuel models for landscape-scale fire modeling. *Journal of Combustion* **2011**, 613424. doi:10.1155/2011/613424
- Sørensen T (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter* **5**, 1–34.
- Sullivan AL (2009) Wildland surface fire spread modelling, 1990–2007. 3. Simulation and mathematical analogue models. *International Journal of Wildland Fire* **18**(4), 387–403. doi:10.1071/WF06144

5. Conclusion

Dans ce chapitre, nous avons effectué une revue des méthodes de comparaison des erreurs entre les résultats d'une simulation et les observations de terrain. Nous avons constaté que ces méthodes ne prenaient pas en compte la dynamique du système et ne comparaient que l'état final du feu, quelles que soient la durée et la forme que celui-ci a pris au cours du temps. Ces résultats ont tendance à sous-estimer l'erreur commise lors des simulations. Afin de prendre en compte la dynamique du système, nous avons implémenté deux méthodes qui font intervenir le temps dans les calculs d'erreurs.

L'article publié dans *International Journal of Wildland* présente les résultats obtenus à partir d'un jeu de données en utilisant les méthodes classiques et les deux méthodes que nous proposons dans le cadre de cette thèse. Il y est montré que ces méthodes sont, à la différence des autres méthodes, discriminantes pour déterminer les simulations dans lesquelles la dynamique est plus semblable à l'observation. Toutefois, les résultats ne sont pas totalement satisfaisants ; ceci est dû en particulier au fait que nous ne disposons que rarement des temps intermédiaires. Aussi, nous arrivons à la conclusion qu'aucune méthode n'est capable de calculer l'erreur de simulation d'une manière parfaite et absolue, car un seul cas ne peut être représentatif. Une évaluation de modèle requiert donc le traitement de plusieurs cas d'étude de manière systématique, ce qui sera réalisé au sein du cadre logiciel présenté dans le chapitre suivant.

CHAPITRE 3

ARCHITECTURE LOGICIELLE

Dans la première partie de ce document, nous avons présenté une approche de modélisation à événements discrets basée sur un formalisme unificateur développé par B.P. Zeigler qui est utilisé comme support pour estimer les erreurs commises par les algorithmes de simulation. Cette approche nous a également permis de formaliser une démarche utilisant un ensemble de scénarios afin d'automatiser la génération des résultats issus de la simulation et leur analyse à partir des méthodes de scores présentées au chapitre précédent. Nous avons développé la notion de cadre expérimental (CE) développée par [Cheon, 2007] qui nous permet de spécifier les conditions d'expérimentation et d'observation du système, afin de prédire ou de traiter les modèles dans les conditions définies par les modélisateurs.

La démarche consiste, dans un premier temps, à proposer un format d'enregistrement des données permettant de représenter, suivant un « méta-langage », les données issues de la simulation et celles en provenance des observations. En utilisant ce langage, nous construisons ensuite une base de données de feux observés. Nous proposons ensuite une démarche de simulation permettant d'obtenir des résultats à partir des conditions initiales issues de la base de données des feux observés. Enfin, nous sommes en mesure de comparer les deux jeux de données pour en extraire les scores d'erreurs.

1. Formalisation du cadre expérimental proposé

La structure générale du cadre expérimental tel que nous l'avons proposé dans le cas d'utilisation des feux de forêt a été présentée dans le premier chapitre (Figure 1.5). Nous allons

CHAPITRE 3 - ARCHITECTURE LOGICIELLE

maintenant décrire son comportement. En utilisant l'approche DEVS, le cadre expérimental est représenté comme un modèle couplé permettant de décrire la structure et l'enchaînement des opérations entre les sous-composants ; les sous-composants Générateur, Accepteur et Traducteur sont représentés par des modèles atomiques. Enfin, la base de données NetCDF et le Simulateur (ForeFire) ne sont pas traités comme des composants DEVS, mais sont liés au cadre expérimental via des liaisons d'entrée/sortie.

La base de données NetCDF contient l'ensemble des données expérimentales et environnementales nécessaires à l'exécution d'une simulation ainsi que les résultats des simulations. Cette base de données permet non seulement de conserver au même format toutes les observations effectuées dans le cas de feux réels, mais également les résultats des expérimentations. Aussi, il est possible d'utiliser pour une nouvelle simulation les résultats issus d'une précédente simulation.

Ces données sont transmises en premier au Générateur pour connaître les points d'ignition, les conditions initiales, les paramètres du terrain, les conditions climatiques, et les paramètres du combustible à envoyer au simulateur. Le simulateur est alors prêt à effectuer une nouvelle simulation. Celle-ci est dirigée par l'Accepteur qui déclenchera et arrêtera la simulation au moment opportun. L'Accepteur reçoit également de la part du Générateur les informations nécessaires au contrôle de la simulation, comme la date de départ, la date d'arrivée du feu observé et en déduit les conditions d'arrêt à vérifier au moment de la simulation. Enfin, le Traducteur réceptionne les résultats de la simulation, les met en forme et transmet à l'Accepteur pour évaluer les scores.

Enfin, le cadre expérimental est un modèle à travers lequel l'utilisateur est en capacité de choisir l'expérience ou les expériences à traiter. Le cadre expérimental enregistre dans son échéancier la liste des expériences à traiter, transmet la première au Générateur et attend la fin de

la simulation et le traitement par l'Accepteur des résultats obtenus pour communiquer les expériences suivantes.

1.1. Représentation du CE suivant l'approche DEVS

Le cadre expérimental (Figure 3.1) est un composant de type modèle couplé, qui propose une interface à l'utilisateur afin que celui-ci puisse définir les expériences à réaliser ainsi que les conditions à appliquer. Le cadre expérimental peut être activé via son port d'entrée « Start » et la réception des résultats de la simulation via son port « Résultats » ou vecteurs de sortie. Il communique avec l'extérieur via son port « Simule » les vecteurs d'entrée qui servent à initialiser une simulation, contrôle la simulation via son port « Contrôle » et retourne une table correspondant aux erreurs calculées pour chaque expérience en fonction des critères de comparaison effectués via son port « Erreurs ».

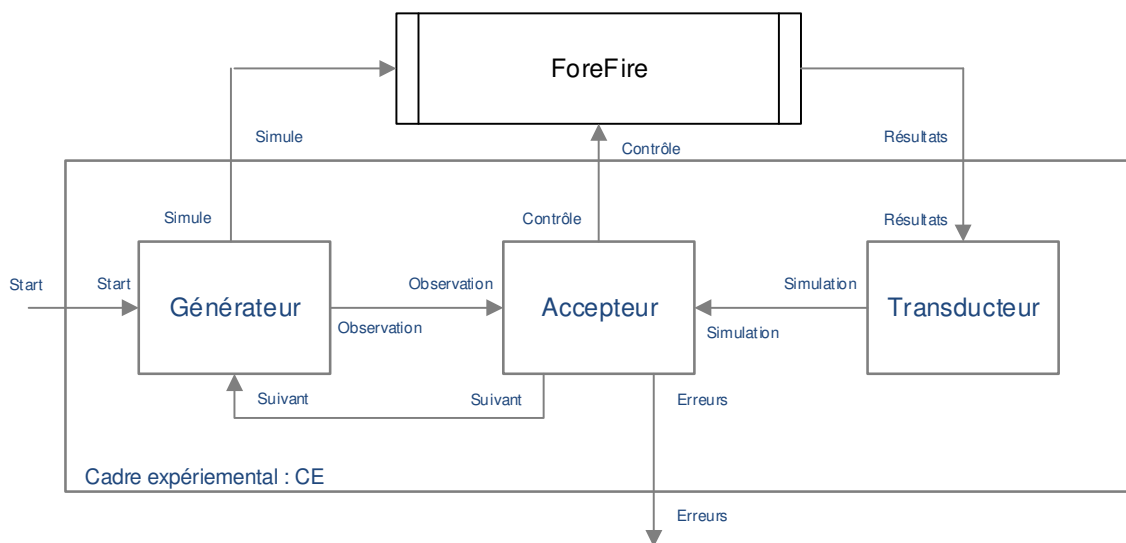


Figure 3. 1. Cadre expérimental : cheminement des données

- $X = \{(Start, v_{start}), (Résultats, v_{résultats})\}$

$v_{start} = \{start, stop\}, v_{résultat} = résultats de la simulation.$

- $Y = \{(Contr\hat{o}le, v_{contr\hat{o}le}), (Erreurs, v_{erreurs})\}$
 $v_{contr\hat{o}le} = Conditions\ d'arr\hat{e}ts; v_{erreurs} = Matrice\ d'erreur$
- $D = \{G\acute{e}n\acute{e}rateur, Accepteur, Transducteur\}$
- $EIC = \{(CE.start, Generateur.start), (CE.r\acute{e}sultats, Transducteur.r\acute{e}sultats)\}$
- $EOC = \{(G\acute{e}n\acute{e}rateur.Simule, CE.Simule), (Accepteur.Erreurs, CE.Erreurs),$
 $(Accepteur.Contr\hat{o}le, CE.Contr\hat{o}le)\}$
- $IC = \{(G\acute{e}n\acute{e}rateur.Observation, Accepteur.Observation), (Accepteur.Suivant,$
 $G\acute{e}n\acute{e}rateur.Suivant), (Transducteur.Suivant, Accepteur.Suivant)\}$

1.1.1. G\acute{e}n\acute{e}rateur

Comme nous l'avons indiqu\acute{e} dans le premier chapitre, le G\acute{e}n\acute{e}rateur r\acute{e}ceptionne la liste des exp\acute{e}riences \`a traiter et cr\acute{e}e les vecteurs d'initialisation qui seront transmis au simulateur. Deux \acute{e}tats sont g\acute{e}r\acute{e}s au sein de ce mod\`ele, une phase qui indique si une simulation est en cours ou non, et une liste d'exp\acute{e}riences \`a r\acute{e}aliser. Les exp\acute{e}riences sont des structures compos\acute{e}es de deux parties : la premi\`ere (v_{init}) est un vecteur d'initialisation \`a destination du simulateur, la seconde ($v_{observation}$) est un vecteur correspondant \`a l'\acute{e}tat final du feu observ\acute{e}. Le G\acute{e}n\acute{e}rateur re\`çoit du cadre exp\acute{e}rimental l'exp\acute{e}rience \`a traiter via une liste d'ID correspondant \`a des num\acute{e}ros d'exp\acute{e}riences enregistr\acute{e}s dans la base de donn\acute{e}es. Pour chaque ID, le G\acute{e}n\acute{e}rateur effectue une lecture sur cette base pour en r\acute{e}cup\acute{e}rer toutes les informations utiles \`a la simulation ainsi que les r\acute{e}sultats des observations. La fonction `read()` retourne donc un couple compos\acute{e} des valeurs ($v_{init}, v_{observation}$). Nous donnons ici une description formelle du comportement du G\acute{e}n\acute{e}rateur au sens du formalisme DEVS.

- $S = \{(phase, liste)\}$ et $S_0 = \{pasive, \emptyset\}$
 $phase \in \{active, pasive\}, liste = \{(v_{init}^1, v_{observation}^1), \dots, (v_{init}^n, v_{observation}^n)\}$
- $X = \{(Start, v_{ids}), (Suivant, v)\},$
 $v_{ids} = \text{liste des id des expériences à traiter}, v = \emptyset$
- $Y = \{(Simule, v_{init}^1), (Observations, v_{observation}^1)\}$
- $t_a(active, -) = (\sigma = \infty)$ et $t_a((pasive, liste.count = 0)) = (\sigma = \infty)$
- $t_a((pasive, liste.count \neq 0)) = (\sigma = 0)$
- $\delta_{ext}(((phase, liste), e), (Start, v_{ids}))$
 $= (phase, \text{pour tous les id dans ids } \{liste.add(read(id))\})$
- $\delta_{ext}(((phase, liste), e), (Suivant, v)) = (pasive, liste)$
- $\lambda((pasive, liste \neq \emptyset)) =$
 $send\{(Simule, Liste[0].v_{init}), (Observations, Liste[0].v_{observation})\}$
- $\lambda((pasive, liste = \emptyset)) = send\{(Observations, \emptyset)\}$
- $\delta_{int}(phase, liste \neq \emptyset) = (active, liste.remove(0))$

1.1.2. Accepteur

Le rôle de l'Accepteur consiste, premièrement, à contrôler la simulation. Pour cela, il est en mesure de calculer les conditions d'arrêt de la simulation à partir des observations (fonction *Contrôle*). Il doit également générer les matrices des feux observés et simulés (Mat_o, Mat_s), puis effectuer les comparaisons en utilisant les méthodes choisies. Pour cela,

L'Accepteur dispose d'une fonction (*creerMat*) permettant de générer ces matrices à partir des vecteurs d'observation et de simulation. Tous les résultats (tables d'erreurs) sont temporairement conservés par le simulateur dans une liste et restitués sur la sortie « Erreurs » lorsque toutes les expériences ont été traitées. L'Accepteur dispose d'une fonction de comparaison des matrices « compare (Mat_o, Mat_s) » qui retourne une table d'erreurs. Enfin, l'Accepteur contrôle un état (phase) qui indique si l'application est à l'arrêt en cours de simulation, terminée ou si une nouvelle simulation doit être effectuée.

- $S = \{phase, ListeErreurs, Mat_o, Mat_s\}$ et $S_0 = \{arrêt, \emptyset, \emptyset, \emptyset\}$;
 $phase \in \{arrêt, simule, nouveau, fini\}$, $ListeErreurs = table\ des\ erreurs$
- $X = \{(Observation, v_{observation}), (Simulation, v_{simulation})\}$
 $v_{observation}$ et $v_{simulation} = vecteurs\ des\ observations\ et\ de\ simulation$
- $Y = \{(Suivant, \emptyset), (Contrôle, v_{controle}), (Erreurs, v_{erreurs})\}$,
 $v_{controle} = vecteur\ de\ contrôle\ de\ la\ simulation$; $v_{erreurs} = table\ des\ erreurs$
- $t_a(arrêt\ ou\ simule, -, -, -) = (\sigma = \infty)$
- $t_a(fini\ ou\ nouveau, -, -, -) = (\sigma = 0)$
- $\delta_{ext}(((phase, -, Mat_o, -), e), (Observation, v_{observation} \neq \emptyset)) =$
 $(nouveau, -, creerMat(v_{observation}), \emptyset)$
- $\delta_{ext}(((phase, -, -, -), e), (Observation, \emptyset)) = (fini, -, \emptyset, \emptyset)$
- $\delta_{ext}(((phase, -, -, Mat_s), e), (Simulation, v_{simulation})) =$
 $(fini, -, -, creerMat(v_{simulation}))$

- $\lambda((fini, ListeErreurs, \emptyset, \emptyset)) = send(erreurs, ListeErreurs)$
- $\lambda((nouveau, -, Mat_o, -)) = send(contrôle, v_{observation} = controleArret(Mat_o))$
- $\delta_{int}(fini, ListeErreurs, \emptyset, \emptyset) = (arrêt, \emptyset, \emptyset, \emptyset)$
- $\delta_{int}(fini, ListeErreurs, Mat_o, Mat_s) =$
 $(arrêt, listeErreurs.add(compare(Mat_o, Mat_s)), \emptyset, \emptyset)$
- $\delta_{int}(nouveau, -, -, -) = (simule, -, -, -)$

1.1.3. Transducteur

Le Transducteur reçoit de la part du simulateur les résultats de la simulation ($v_{simulation}$). Il est en mesure de les mettre en forme pour les enregistrer dans la base de données de feu, mais également d'en extraire un vecteur de simulation ($v_{simulation}$) qu'il transmettra à l'Accepteur (fonction vecteurSim).

- $S = \{phase, v_{simulation}\}$ et $S_0 = \{pasive, \emptyset\}$;
 $phase \in \{active, pasive\}, v_{simulation} = vecteur\ issu\ de\ la\ simulation$
- $X = \{(résultats, v_{résultats})\}, v_{résultats} = résultats\ de\ la\ simulation$
- $Y = \{(Simulation, v_{simulation})\}$
- $t_a(active, -) = (\sigma = 0)$
- $t_a(pasive, -) = (\sigma = \infty)$
- $\delta_{ext}(((pasive, -), e), (Résultats, v_{résultats})) = (active, vecteurSim(v_{résultats}))$
- $\lambda((active, v_{simulation})) = send(Simulation, v_{simulation})$

- $\delta_{int}(active, v_{simulation}) = (pasive, \emptyset,)$

1.2. Les conditions d'arrêt des simulations

L'Accepteur gère et arrête le processus d'expérimentation si les conditions d'arrêt sont vérifiées. Les conditions choisies sont : la surface totale brûlée de la simulation est égale à la surface totale brûlée d'observation, et la durée de vie du feu simulé est égale à douze heures (12h).

Pour la première condition d'arrêt, le simulateur continue la simulation en calculant à chaque pas de la simulation la surface totale du feu simulé. S'il n'atteint pas la même valeur du feu observé, il reste en simulation. Sinon, la simulation est arrêtée.

Pour la deuxième condition d'arrêt, le simulateur continue à simuler en respectant le fait que la date de la fin du feu simulé (date où le feu expérimental s'éteint) est égale à sa date de déclenchement ajoutée à douze heures de durée de vie.

Si la formalisation est l'étape permettant de cadrer le problème conceptuellement, sa réalisation demande de s'interroger sur les différents acteurs afin d'identifier les cas d'utilisation, les formats et l'environnement de développement.

2. Cas d'utilisation

L'étude des cas d'utilisation est continue tout au long du processus de formalisation du CEFF. Ce chapitre présente les types d'interactions retenues en utilisant la notation UML. La Figure 3.2 présente le diagramme final des cas d'utilisation. Nous avons huit acteurs :

- **Fournisseur de données des incendies** : il veille à ce que les données des feux de forêt soient fournies à l'analyste pour être traitées. Ces données sont les données des incendies portant sur des cas réels.

- **Fournisseur de données météorologiques:** il veille à ce que les données de météo (climat, vent, etc.) soient fournies correctement comme paramètres pour simuler les feux.
- **Fournisseur de données topographiques:** il veille à ce que les données des topographies telles que la forme et l'élévation soient fournies.
- **Analyste :** il extrait les données des incendies fournies et les transcode en un format standard (NetCDF).
- **Simulateur :** il choisit les paramètres d'une simulation (conditions initiales génératrices des feux à partir des fichiers CDF, météo, topographie). Lorsque ces paramètres sont définis, il est chargé de lancer les simulations.

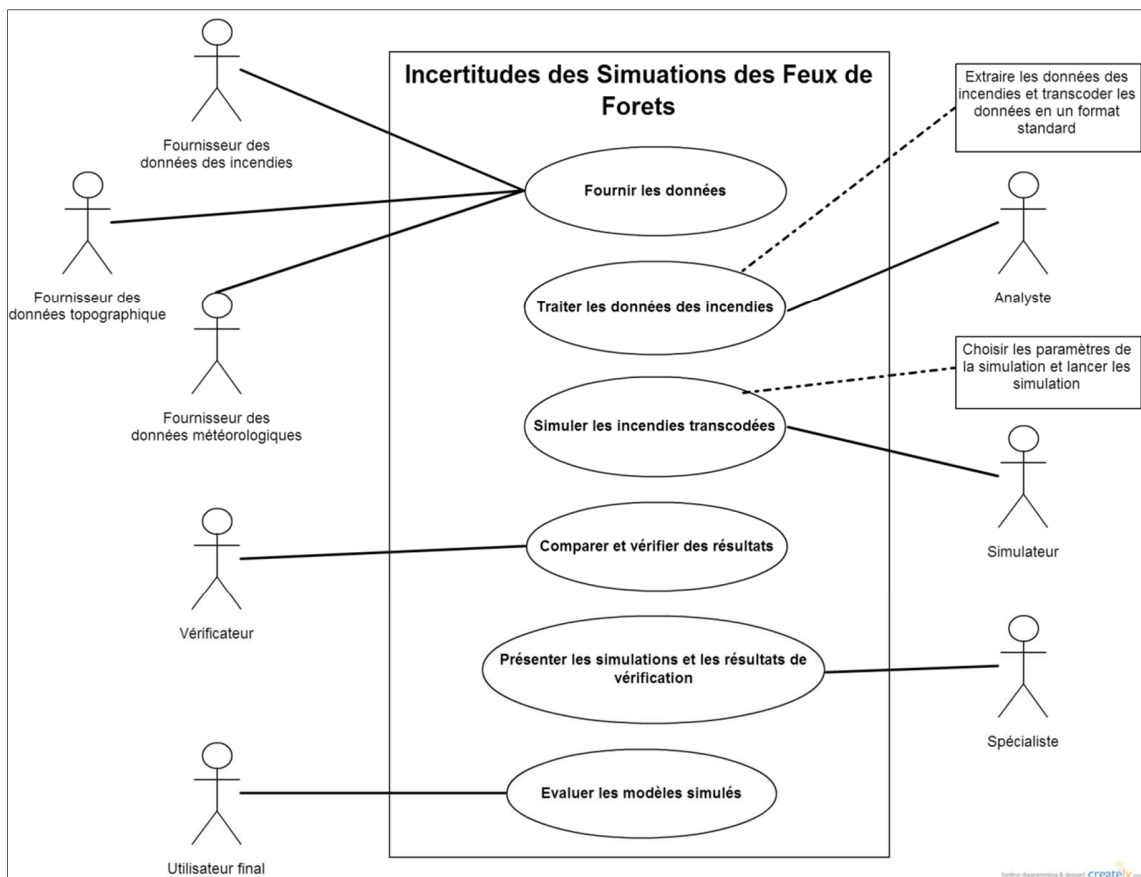


Figure 3.2. Diagramme final des cas d'utilisation

- **Vérificateur** : il analyse les simulations obtenues et vérifie si elles sont cohérentes ou prévues. Il compare les simulations avec les observations. Les résultats de ces comparaisons sont des indices d'erreur de la simulation.
- **Spécialiste** : il est chargé de présenter les données des feux de forêt simulés ou un fichier (KML) compatible avec Google Earth.
- **Utilisateur final** : il utilise les indices d'erreurs obtenus pour évaluer les modèles simulés.

Ce diagramme (Figure 3.2) sert à définir les tâches à réaliser après le processus de formalisation du cadre. Ainsi, il nous permet de définir les acteurs qui interagissent au sein du cadre proposé et les exigences de ces interactions.

Nous pouvons en déduire que le format des fichiers CDF est adapté, car il est utilisé par les géophysiciens. Aussi, il est adapté pour sauvegarder les données des feux de forêt (observés et simulés). De même, nous avons utilisé le format KML lors de la présentation et la vérification des résultats obtenus puisque il est un format standard pour la visualisation rapide, facile à interpréter et à utiliser.

L'implantation informatique nécessite un encodage d'information et l'intégration dans un environnement de développement adapté aux données géophysiques. Ces aspects sont présentés dans la section suivante.

3. Base de données des feux observés

La construction d'une base nécessite la définition d'un lexique, d'un format de ces données et d'un conteneur de base de données, aucun standard n'existant pour les données

incendies. Cette section propose un lexique ainsi que sa réalisation au standard NetCDF et l'implantation d'une API en Python de manipulation et de transcodage.

3.1. Lexique (Glossary)

Un lexique est une liste de termes qui décrivent un domaine spécifique de connaissances incluant les définitions et les explications de tous ces termes.

Dans notre cas, nous avons collecté des données basées sur les observations des feux de forêt. En d'autres termes, une analyse du domaine a été réalisée à travers la consultation des fiches de terrain illustrant des cas réels des feux de forêt. Ces fiches ont été rédigées par un analyste du domaine à partir des rapports d'observation de sapeurs-pompiers.

Les rapports des sapeurs-pompiers sont une forme textuelle pour présenter un feu. Ils contiennent leurs observations réalisées au cours des actions de lutte contre le feu. Voici quelques lignes qui sont extraites de Fiches Terrain Feu de Suartone.

T : Town, Village

Suartone (Commune de Bonifacio)

T : Burned surface

457 Ha

T : Ignition time

EX: 12/19/1987 12:54:41

28/07/2003 15h00

TP : Ignition location

Ex: E9.5424 N41.5478 DFCI : NB86E35

E9°13'30.48" N41°28'21.93"

TP : Fuel moisture content : location : species,(value)

EX : Conca : Arbousier, 50,22%

Erica Arborea 28%.
Fuel moisture is defined here as $m_{water}/(m_{total})$ $m_{total}=m_{water}+m_{dryfuel}$

TS : Intermediate fire lines (time and shape file for each line)

Dessin sur photo ou trace gps

Aucune

Nous avons étudié deux cas réels : Suartone, Corse, France en 28/07/2003 à 15h00 et Favone, Corse, France en 08/07/2009 à 15h02. Le Tableau 3.1 montre le lexique qui a été défini à partir des données collectées. Il est évident que ce tableau décrit les comportements de n'importe quel feu de forêt observé dans la nature.

3.2. La configuration du conteneur NetCDF

La gestion des données est la définition du type de chaque terme présenté dans le lexique ci-dessous. D'après la description, un type est donné pour chaque terme suivi par une unité. La nomination des données a été réalisée en respectant au maximum la convention de nommage NetCDF Climate and Forecast (CF) Metadata Convention [Eaton et al ., 2006]. Nous avons choisi le NetCDF - Version 4 et HDF - comme conteneur puisqu'il nous offre un ensemble de bibliothèques et logiciels qui permettent la création, l'accès et le partage de données scientifiques sauvegardées sous forme de tableaux. Le cadre expérimental proposé a été configuré avec les données du lexique. D'abord, nous avons défini les dimensions qui formeront le cadre expérimental de notre modèle NetCDF. Dans le Tableau 3.2, la première colonne présente le nom de la dimension donnée en respectant la convention CF, le deuxième est la longueur dont ILLIMITÉE (Unlimited) signifie une dimension dynamique, et le troisième montre l'explication.

Les variables ont ensuite été définies à partir du lexique des observations avec les attributs désirés, en respectant la convention CF. Nous avons utilisé uniquement l'attribut unité pour sauvegarder les unités de ces données (Tableau 3.3). Afin de perturber les paramètres initiaux, nous avons ajouté deux nouveaux termes sur le fichier NetCDF d'observation, à savoir la vitesse et la direction du vent.

Données	Data	Explication
Information de la localisation du feu	Fire description	Description de la localisation du feu comme le code postal, la commune, le code INSEE, etc.
Le temps du déclenchement du feu	The ignition time of fire	Date et heure du déclenchement du feu
Le temps de la fin du feu	The final time of fire	Date et heure de la fin du feu
Les points du déclenchement du feu	Ignition points of fire	Points où le feu a été déclenché
La surface totale brûlée	Total burned surface	Représente l'aire du terrain brûlé
L'action de lutte contre le feu	Availability of fighting tool	Description de l'action de la lutte contre le feu. Elle est sous forme d'un ensemble d'informations (temps d'alerte, heure d'arrivée, emplacement, équipement, forces, quantité d'eau, etc.)
Le feu passe un point connu	Fire passes known points	Décrit si le feu passe à un endroit connu (arrivée sur une route nationale, etc.)
Les taux de la propagation du feu	Rate of fire spread	Distance horizontale sur laquelle le front du feu se déplace par unité de temps
L'intensité de l'incendie	Fire intensity	Représente la chaleur dégagée par le front du feu
La hauteur de la fumée	Smoke height	Estimation de la distance entre le bord de la flamme et le point où la fumée disparaît
La hauteur de la flamme	Flame height	Estimation des intensités de rayonnement des flammes
Les contours	Contours	Ensemble de points (latitude, longitude, altitude) qui représentent la forme de feu

Tableau 3.1. Lexique des données des feux de forêt

Nom	Longueur	Explication
fire_representation	1	Toute information sur les feux sera enregistrée dans le fichier NetCDF. Elle est sous forme de texte, comme le nom.
time	1	Il est consacré à la dimension temporelle (dimension physique)
point2D	2	Il définit la forme du point composé de la longitude et de la latitude (dimension d'indice)
number_of_ignite_points	ILLIMITEE	Il gère le nombre de points de déclenchement du feu (dimension d'indice)
Wind	1	Il présente la dimension du vent qui est utilisée par les variables vitesse et direction du vent
datetime_for_fire	1	Il définit la forme des variables le temps de déclenchement / fin d'un incendie
info_fighting_tool	ILLIMITEE	Il définit la dimension pour toutes les informations concernant l'action de lutte
info_passed_points	ILLIMITEE	Il définit la dimension pour toutes les informations concernant le feu qui passe par des points connus comme une route nationale
number_of_contour_points	ILLIMITEE	C'est une dimension d'indice pour présenter le nombre de points dans un contour
number_of_contours_observations	ILLIMITEE	Elle montre le nombre de contours observés par feu. Chaque Tobs est le temps d'observation d'un contour. Exemple: à Tobs1 = 12h00, il est contour1 et à Tobs2 = 12h25, il est contour2. Dans ce cas, cette dimension est 2, cela signifie qu'il y a 2 contours avec 2 temps d'observation
burned_surface_value	1	C'est la forme présentant la variable surface brûlée
fire_spread_value	1	Il définit la forme de la variable vitesse de propagation du feu
number_of_observations_of_the_rate_of_fire_spread	ILLIMITEE	C'est une dimension d'indice pour présenter le nombre d'observations pour la vitesse de propagation du feu
fire_intensity_value	1	Il définit la forme de la variable intensité du feu
number_of_observations_fire_intensity	ILLIMITEE	C'est une dimension d'indice pour présenter le nombre d'observations de l'intensité du feu
flame_height_value	1	Il définit la forme de la variable hauteur de flamme
number_of_observations_flame_height	ILLIMITEE	C'est une dimension d'indice pour présenter le nombre d'observations quant à la hauteur de la flamme
smoke_height_value	1	Il définit la forme de la variable hauteur de la fumée
number_of_observations_smoke_height	ILLIMITEE	C'est une dimension d'indice pour présenter le nombre d'observations quant à la hauteur de la fumée

Tableau 3.2. Les dimensions qui forment le cadre expérimental

Le Tableau 3.3 montre les variables définies pour le cadre expérimental en fonction des dimensions. La première colonne présente le nommage, le deuxième présente le(s) dimension(s) requise(s) et le troisième présente l'unité définie.

Nom de la dimension donnée en respectant la convention CF	Dimension(s)	Attribut (unit)
fire_id	fire_representation	-
fire_name	fire_representation	-
fire_description	fire_representation	-
ignite_time_of_fire	datetime_for_fire	-
ignite_time_of_fire	datetime_for_fire	-
fire_ignite_points	number_of_ignite_points , point2D	-
wind_speed	wind	m·s ⁻¹
wind_from_direction	wind	Degrees
fighting_tool	info_fighting_tool	-
fire_pass_known_points	info_passed_points	-
burned_surface	burned_surface_value	ha
front_contours	number_of_contour_observations	-
front_contour n	number_of_contour_points, point2D	-
table_of_fire_spread	number_of_observations_of_the_rate_of_fire_spread	-
rate_of_fire_spread	fire_spread_value	m·s ⁻¹
table_of_fire_intensity	number_of_observations_fire_intensity	
fire_intensity	fire_intensity_value	kW·m ⁻¹
table_of_smoke_height	number_of_observations_smoke_height	
smoke_height	smoke_height_value	m
table_of_flame_height	number_of_observations_flame_height	-
flame_height	smoke_height_value	m

Tableau 3.3. Les variables du cadre expérimental

3.3. Les traitements et les filtres

Les analyses des données descriptives et géographiques de Prométhée nous ont permis de classer les observations en plusieurs catégories.

CHAPITRE 3 - ARCHITECTURE LOGICIELLE

Les observations de *Type1* sont parfaites et acceptées instantanément sans aucun traitement. Une observation qui appartient au *Type2* sera rejetée car elle ne peut servir dans le cadre expérimental pour être comparée aux résultats de la simulation. Si le point d'ignition est situé en dehors de la zone brûlée, dans une zone incombustible comme un lac ou une mer, etc., et si nous disposons d'un contour final, alors cette observation est de *Type3*.

<i>Type1.</i>	Le point d'ignition et le contour sont cohérents
<i>Type2.</i>	L'observation ne dispose que d'un point d'ignition
<i>Type3.</i>	Le point d'ignition est mal placé mais le contour est cohérent
<i>Type4.</i>	Seul le contour est disponible.

Pour les observations de *Type3* et *Type4*, des traitements seront nécessaires pour corriger et améliorer l'observation.

Nous pouvons identifier les traitements suivants :

- Pour *Type3* : les traitements effectués correspondent à une série de changements des points d'ignition afin de retrouver à dire d'expert les bonnes conditions initiales.
- Pour *Type4* : le point d'ignition n'existe pas. Dans ce cas-là, il est supposé que le point d'ignition est le même que le premier point du contour, correspondant au point signalé. Par conséquent, ce cas est transformé en *Type3*, ce qui nécessite les traitements appliqués à ce type (*Type3*).

Voici des vues aériennes des incendies observés avec leurs simulations qui illustrent les cas *Type1*, *Type2*, *Type3* et *Type4*.

Deux exemples de cas acceptés (*Type1*) :

- Le cas Porto-Vecchio est un cas accepté.

Les informations sur ce cas :

- Date : 21/08/2006-16:42

CHAPITRE 3 - ARCHITECTURE LOGICIELLE

- L'observation est un polygone plein rouge (droite)
- La simulation est un polygone rouge (gauche)
- Le point d'éclosion est 9.26673317 41.53903393
- Surface brûlée : 3.0 ha



Figure 3.3. Le cas Porto-Vecchio

- Le cas Santo-Pietro-di-Tenda est un cas accepté.

Les informations sur ce cas :

- Date : 29/08/2003-00:07
- L'observation est un polygone plein rouge
- La simulation est un polygone rouge
- Le point d'éclosion est 9.202331 42.615511
- Surface brûlée : 5532.51 ha



Figure 3.4. Le cas Santo-Pietro-di-Tenda

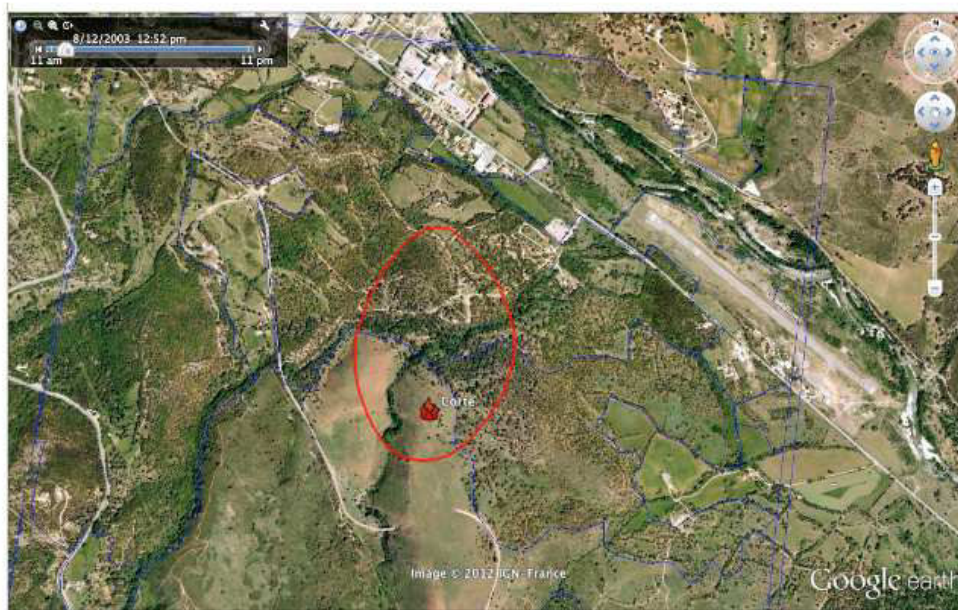


Figure 3.5. Le cas Corte

Un exemple des cas rejetés (Type2) :

- Le cas Corte est un cas refusé (n'a pas de contour):

Les informations sur ce cas :

- Date : 12/08/2003
- L'observation est un point seulement
- La simulation est un polygone rouge (mais sans aucun intérêt)
- Le point d'allumage est 9.176419, 42.288533
- Surface brûlée : 27.68 ha

Un exemple de cas modifié (Type3, Type4) :

Le cas Propriano original (point d'éclosion sur la route):

Les informations sur ce cas :

- Date : 24/08/2008-12:51
- L'observation est un polygone plein rouge (gauche)
- La simulation est un polygone rouge (droite)
- Le point d'éclosion est 8.90268584 41.66956104
- Surface brûlée : 2.8 ha



Figure 3.6. Le cas Propriano avant la modification

Le cas Propriano modifié : le nouveau point d'éclosion est 8.903133 41.669294



Figure 3.7. Le cas Propriano après la modification

3.4. API

Une API a été développée en Python et nommée « Fire2NetCDF ». La version « 1.3 » de cette API est désormais disponible au téléchargement par la communauté.

Elle se compose de deux scripts principaux dont le premier est la couche de l'ontologie qui est une classe abstraite "*ForestFire.py*". Le deuxième script, "*Fire2NetCDF.py*", permet de sauvegarder les données dans des fichiers NetCDF ; ce script permet également de lire ces fichiers. De plus, un script Python "*fire2kml.py*" permet d'exporter les fichiers créés au format KML. Donc nous avons :

- "*ForestFire.py*":
 - Classe qui représente tous les comportements d'un feu de forêt observé.

- Des attributs qui représentent toutes les informations d'une observation existante dans le tableau 1.3 et qui est nécessaire pour initialiser le modèle.
- "*Fire2NetCDF.py*" :
 - Ce script permet de sérialiser les instances de la classe *ForestFire* en fichiers NetCDF et en opérations inverses.
- "*fire2kml.py*" :
 - Ce script contient les fonctions nécessaires pour écrire un fichier KML.

Cette API est capable de présenter n'importe quel feu observé par une instance de la classe "*ForestFire.py*" quelle que soit la source de données (soit une fiche terrain qui a été rédigée par les pompiers, soit une base de données existante) grâce à cette classe.

Une fonction de génération en KML du script "*fire2kml*" assure la transformation des fichiers NetCDF en générant des fichiers au format KML depuis les instances de *ForestFire* et s'assure qu'ils sont compatibles avec Google Earth.

La structure du cadre expérimental proposé et l'API est publiée dans l'article de la section suivante.

3.5. Publication

Les processus de la construction de la base d'observation sont décrits dans la publication « *An experimental frame for the simulation of forest fire spread* » qui présente le travail de définition d'un format et l'API *Fire2NetCDF*. Elle a été présentée dans les procédures de *Winter Simulation Conference en Arizona, USA* – Décembre 2011.

Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

AN EXPERIMENTAL FRAME FOR THE SIMULATION OF FOREST FIRE SPREAD

Bahaa Nader
Jean Baptiste Filippi
Paul Antoine Bisgambiglia

University of Corsica
UMR CNRS 6134
Campus Grimaldi, Corte 20250, France

ABSTRACT

Wildfire is a constant risk due to its danger on both human and natural resources so modeling and simulation is an important tool to understand and forecast this phenomenon. A basic element of any simulation model is to define a way to store, compare and exchange observation and model results. Without a clear and standardized data structure, results and observations lack usability, inter-comparability and expressiveness.

In this article we propose a well defined data format and API that can represent observation and model outputs. It provides a convenient way to transform fire data and can export any simulated or observed fire into KML for easy visualization. This specification enables the implementation of an experimental frame that is independent of simulation code.

A database of more than 600 fires has been compiled using the API, enabling large scale reanalysis for any code that can be adapted to the proposed experimental frame.

1 INTRODUCTION

In the last quarter century, the phenomenon of wildfire has attracted researchers in all scientific domains as it has become a major player of disturbance in the circumboreal forest.

As any research subject, wildfire requires observation data to investigate phenomenon and build models. Nevertheless and unlike other geophysical areas, there is not a standard way to collect and store data. Also, wildfire is an environmental system that often acts over large spatial scales, long time frames and heterogeneous units of study. The difficulty is not only formulating this system, but also conceiving higher-level problems with the interaction between modeling paradigms, models interoperability and model reusability. Finally, storing simulations' results is essential to compare simulated fire with an observation or another simulation; but this can only be done in a reusable way.

General theory of modeling and simulation, introduced by Zeigler (Zeigler, Praehofer, and Kim 2000), is the basis of our modeling and simulation environment. This approach of modeling provides a well defined architecture to specify models of a wide variety of modeling paradigms (Zeigler 1987). The DEVS formalism is a set-theoretic formalism (Zeigler, Praehofer, and Kim 2000) and supports specification of discrete event systems in a hierarchical, modular form (Zeigler 1987). It includes a formal representation capable of mathematical manipulation just as differential equations perform this role in continuous systems.

The concept of the Experimental Frame is also developed in the general theory and standard data interchange structure and format. These are the requirements to actually build an experimental frame.

The goal of this paper is to fill the need for a framework adapted to manipulate wildfire observation and simulation data. Currently, this framework is applied to the DEVS based ForeFire code, which simulates forest fire growth (Balbi, Morandini, Silvani, Filippi, and Rinieri 2009, Filippi, Morandini, Balbi, and Hill 2010).

Nader, Filippi, and Bisgambiglia

Data definition for the phenomenon of wildfire as well as currently used formats is available in section two and a presentation of the Experimental Frame for the proposed framework in section three. Section four illustrates a detailed presentation for the proposed framework. Section five presents applications of this proposed framework in our simulation environment of forest fire.

2 REPRESENTATION OF THE WILDFIRE PHENOMENON

The phenomenon of wildfire is a natural part of most ecosystems (Flannigan, Amiro, Logan, Stocks, and Wotton 2006). It is the uncontrolled combustion of the vegetation fuels that occur in the countryside or a wilderness area (Miller, Landres, and Alaback 2000). The type of vegetation burned differ from one fire to another; so there are different names for this phenomenon, such as forest fires, hill fire, grass fire and brush fire. Each wildfire has its unique proprieties that describe the fire behavior. Those proprieties are the speed of the spread of a fire from the original source, the total burned surface, the ability of jump gaps (roads and fire breaks) and the intensity of fire. There are others factors influencing fire activity such as weather climate and the combustible material present (Ottmar, Sandberg, Riccardi, and Prichard 2007).

2.1 Types of Wildfire Data Presentation

Historically, wildfire incidents have been documented as reports mostly to analyze firefighting successes and failures with no computerized formats available. There are two types of presentation available in those reports: graphical and textual.

The graphical presentation is in map form. For example, wildfire could be presented on a map like a set of symbols or keys. The ignition point of a wildfire can be 'X', the first contour can be as a set of 'Squares' and the final contour as a set of 'Circles'. The map legend shows the corresponding data for each key. Actual icons and symbols are usually normalized at the country level.

The textual presentation is in the form of text report. Firefighters write these reports to describe the wildfire that they fight against. A report consists of field observations for wildfire's parameters such as fire spread, fire intensity, shape of fire, location of final fire line with a time of observation.

Most of the available information for observed fires exist in this last form. This way of presenting the data is very flexible but it is not convenient for automatic processing, especially in the context of an experimental frame for simulation software.

2.2 Data Formats of Use in Wildfire Simulation Software

Nowadays there is a fundamental need to simulate the wildfire in a more efficient way (Finney 1994b). For this domain of research, spatial data of fire provided to simulator is important. This data are essential for compare and ameliorate simulated models and specific data formats have been developed to fit each of these simulation software.

2.2.1 Farsite

FARSITE(Finney 1994a) is a simulator of two-dimensional fire growth model (Service 1998) that automatically computes wildfire growth and behavior for long time periods under heterogeneous conditions of terrain, fuels and weather. It was developed by Mark A. Finney.

It combines spatial information on topography, fuels, weather and wind files (Finney and Ryan 1995). This means FARSITE requires a spatial coincident landscape raster information to run (Keane, Garner, Schmidt, Long, Menakis, and Finney 1998).

Two formats are mandatory to run simulation in FARSITE: The Landscape (.LCP) file contains raster data obtained from a GIS for terrain and fuels. All this raster data must be in ASCII grid format, from either ARC/INFO (or ArcView) or GRASS. ASCII raster files must be used to generate a Landscape (.LCP) file

Nader, Filippi, and Bisgambiglia

before running a simulation; Initial Fuel Moistures (.FMS): ASCII text file, required to begin the process of calculating site specific fuel moistures at each timed step of the simulation (Finney and Ryan 1995).

For each specific run, a weather and custom model must be defined. Run-specific files: Custom Fuel Model (.FMD) files, Conversion (.CNV) files, Weather (.WTR) files present weather and Wind (.WND) files: for wind distribution. FARSITE output files, consist of Raster Output Maps (arrival time, fire intensity, etc.) and fire contour polygon files (Finney 1994a).

2.2.2 Landfire

LANDFIRE (Rollins, Keane, Zhu, Menakis, Hann, and Shlisky 2003) is known as Landscape Fire and Resource Management Planning Tools. It combines vegetation, fire and fuel distribution.

LANDFIRE produces outputs in the form of geospatial products (spatial layers) that describe potential and existing vegetation type, surface, canopy fuel characteristics, fire behavior, fire regime classes, biophysical, environmental site potential and simulated historical fire regimes conditions. These outputs (fuel and risk maps) are compatible with ArcGrids and ArcMap grids, also they are consistent, standardized across the United States (Rollins, Keane, Zhu, Menakis, Hann, and Shlisky 2003), but are not suited to store fire incidents.

2.2.3 FlamMap

FlamMap developed by Finney (Finney 2006). It is a tool to simulate and analyze fire behavior (Finney 2006). It computes potential fire behavior characteristics (spread rate, flame length, fireline intensity, etc.). All mentioned input formats for FARSITE are required for FlameMap. The calculations are performed independently for each cell on the gridded landscape (Finney 2006).

2.2.4 Asphodel

ASPHODEL is software of tactical situation (SITAC - *Situation TACTiques* in French) under development, in laboratory of the University of Savoy, France (Dumond, Garitat, Gosse, Michelis, Simonpieri, and Tran 2009). The SITAC is the French graphical representation of a wildfire incident including the fighting actions. The aim of this project is to manage human and material resources deployed to forest fire fighting. It is capable of drawing a SITAC on different base maps.

2.2.5 ForeFire

ForeFire has been developed in our laboratory of University of Corsica, Corsica, France (Balbi, Morandini, Silvani, Filippi, and Rinieri 2009). It is based on the environment of DEVS formalism and simulates the forest fire propagation model. Also, ForeFire requires fuels distribution, weather, wind and elevation of terrain (Filippi and Balbi 2009).

These requirements are in the form of input files that are defined in a XML file (Filippi, Morandini, Balbi, and Hill 2010). This file contains simulation parameters, such as the ignition date of forest fire, simulation duration, the point of ignition, the segment of the forest fire shape and the resolution. As mentioned before, it defines input files. These files are: Fuel Model is an XML file, describes the fuel distribution of the terrain; Shape: it is (.shp) file, describes the shape (form) of the terrain; Elevation: it is (.asc) file, presents the elevation of the forest fire location and Wind files, are (.asc), present the wind distribution.

Again, most of these formats have been developed to fit specific software, with no real effort to provide a standard format to store and manipulate a wildfire incident. Next section presents the proposed Experimental Frame for wildfire simulation, where a standard data format is required to actually implement the framework.

Nader, Filippi, and Bisgambiglia

3 EXPERIMENTAL FRAME FOR THE PROPOSED FRAMEWORK

An Experimental Frame (EF) is a specification of the conditions under which a system is observed or experimented. It is also the operational formulation of the objectives required in a modeling and simulation project (Zeigler, Praehofer, and Kim 2000).

Simulation using DEVS environment facilitates the development of Experimental Frame, which leads to specifying a set of various scenarios. Also, it leads to generating results that can help in comparing the performance of different models. Any EF consists of a generator, an acceptor and a transducer. The generator generates input segments to the system in a known, desired fashion. The acceptor monitors an experiment to verify the desired conditions are met. The transducer observes and analyzes the system outputs. The generator and transducer elements can only be implemented if data exist which can be interpreted by the simulation software.

The EF for the proposed framework, Figure (1), consists of the generator, the acceptor and the transducer. The generator stimulates the Forest Fire Simulation with known data segments [a Fire State from a NetCDF file (.nc), Landscape, Weather and Simulation Parameters]. The acceptor manages the ForestFire simulation, respecting the Simulation Parameters. Finally, the transducer analyzes the output and stocks the State of Fire in NetCDF (.nc) file, which could be used as an input for the generator.

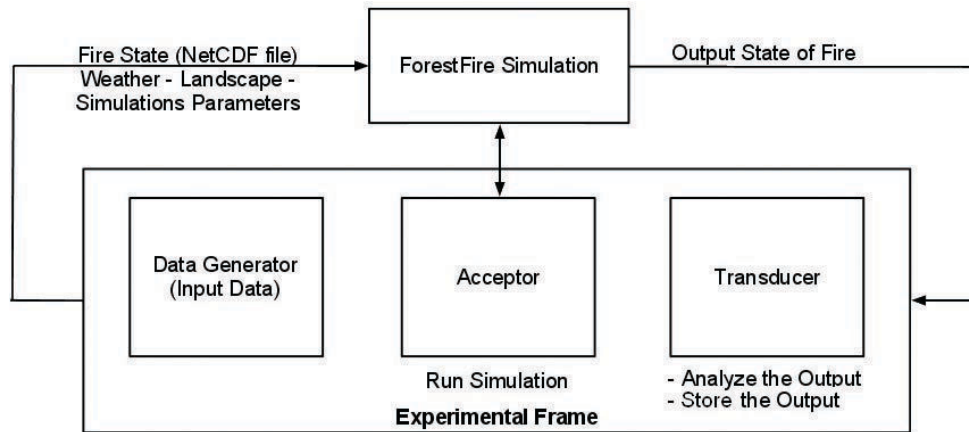


Figure 1: The Experimental Frame for the proposed framework.

The Experimental Frame requires a well defined Input/Output diagram. Figure (2) shows the I/O diagram for the EF for the proposed framework. Input consists of four blocks: 1- Fire State from a NetCDF file (.nc): describes the initial state of fire such as its ignition points, ignition time, area and firefighting action. 2- Landscape defines the distribution of fuels (combustible materials) and the altitude. 3- Weather defines range of wind (speed and direction) and also the weather in this location. 4- Simulation Parameters define the numerical parameters of simulation like duration and resolution of simulation.

These four blocks are considered as the data segments of the data generator.

Output consists of one block: Final Fire State: describes the final state of fire; such as its contours, end time, rate of fire spread. This final state is stored by the transducer in a NetCDF (.nc) file.

As shown in Figure (1), the output file (.nc) can be used as input to the data generator. In this case, contours will be the ignition segment points of the fire and end time will be the ignition time.

Finally, the simulation block represents the periodic aspect, respecting the Simulation Parameters are managed by the acceptor.

Nader, Filippi, and Bisgambiglia

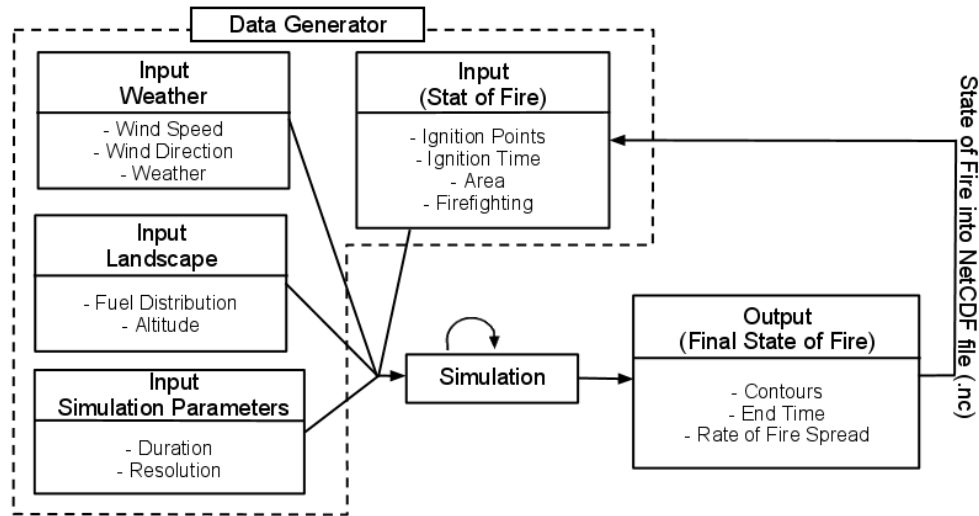


Figure 2: Inputs/Outputs diagram of the Experimental Frame

4 FOREST FIRE EVENT DATA FORMAT

The development of a data framework has three phases: first the construction of a glossary, second the definition of a file format corresponding to the data on the glossary and third the choice and the configuration of a proper container. This section presents those three phases in details.

4.1 Construction of a Glossary and Defining the Format

A glossary is a list of terms for a specific domain of knowledge with the definitions or the explanation for those terms. In our case, we list all data based on observation of forest fire in nature. In other words, we made a domain analysis by consulting information sheets of real cases of wildfire. We studied two real cases: *Suartone, Corsica, France occurred in 28/07/2003 at 3:00PM* and *Favone, Corsica, France in 08/07/2009 at 3:02PM*. Table 1 cited all collected data, its format and units.

Although in Table 1 appears sufficient to describe fire behavior, it is not enough to initiate simulation of this fire, because there are other required factors. These factors are the physical proprieties of the place, where the forest fire spreads. Fuel distribution, altitude, weather and wind are needed to simulate a fire. They are the simulation data and are also considered as inputs of the data generator (Figure 2).

Table 1: Glossary of fire incident data collected from information sheets of terrain and the identification of the format.

Data	Explanation	Format	Unit
The Ignition Time Of Fire	Date and time when the fire is triggered	Date - time	-
The Final Time Of Fire	Date and time when the fire is extinguished	Date - time	-
Ignition Points Of Fire	Points where the fire is triggered	point (altitude, longitude, elevation) or a set of points	-
Total Burned Surface	It represents the total burned surface	Double	Ha
Availability Of Fighting Tool	It mentions the fighting tool applied on ground to fight the spreading of a wildfire	A set of data (information: time of alert, time of arrival, location, equipment, forces, quantity of water, .. etc)	-

Nader, Filippi, and Bisgambiglia

Fire Passes Known Points	It describes the fire line passes any knowing points like a highway, a river, .. etc	A set of data (information: name, location, damage, .. etc)	-
Rate Of Fire Spread	It is the horizontal distance that the fire front moves per unit of time	Double	m/s
Fire Intensity	It represents the heat released per meter of fire front	Double	kW/m
Smoke Height	Is used to estimate the distance between the edge of flame and the point where the smoke disappears	Double	m
Flame Height	It is used to estimate radiation intensities of flames	Double	m
Contours	A set of points (altitude, longitude, elevation) that represents the fire shape	An array of points	-

As a result, wind speed and wind direction can be added to the fire incident data in the glossary (Table 1), because they can be noted with a corresponding observation time. There are two terms added to the glossary: *Wind Speed* is the horizontal motion of the wind in atmosphere and has a major effect on fire's speed of propagation, its format is double and its unit is m/sec; *Wind Direction*, in meteorological reports, is usually given as the direction from which wind is blowing. It defines the direction of fire's propagation, format: double and unit: degree.

Next subsection discusses the container and its configuration with the glossary data.

4.2 Data Container

Last phase of framework development can be represented on two steps: the choice of a container that answers our needs and then the configuration of this container. The purpose of this last step is to reinforce a smooth transformation from forest fire simulation to files and vice-versa. We began this phase by the identification of the essential needs. We need a scientific container which provides a standard format as output and machine-independent. We choose NetCDF - Version 4 and HDF - as a container which responds to our needs and realizes our goal. NetCDF - *Network Common Data Format* - extended from CDF - *Common Data Format* - a library and toolkit that has been developed by NASA. The NetCDF software was developed at the Unidata Program Center in Boulder, Colorado (Rew and Davis 1990). NetCDF consists of a set of libraries and machine-independent data formats that provide the creation, access and sharing of the array-oriented scientific data.

4.3 NetCDF FRAMEWORK PRESENTATION

A NetCDF container consists of a set of parameters, which are the dimensions and the variables (Rew and Davis 1990). We configure our NetCDF container with the collected data provided from the glossary. This next section is dedicated for the presentation of our desired framework.

4.3.1 Dimensions and Variables

A dimension is used to present either a real physical dimension or to index other quantities. Time, latitude, longitude, or height represent the physical type, in climate simulations. A station or a model-run-number is considered as the other type. Each dimension is defined by a name and a length that is an arbitrary positive integer. A table of dimensions (Table 2) is defined for the desired framework. Naming respects the NetCDF Climate and Forecast (CF) Metadata Convention (Eaton, Gregory, Drach, Taylor, and Hankin

Nader, Filippi, and Bisgambiglia

2006). In case a similar dimension is found in CF's table, its same name is used; if not, a proposed name is given.

In order to store a bulk of data in a NetCDF dataset, variables are needed. They are arrays of values of the same type. A variable has a name, a data type and a shape described by its list of dimensions. Also, it has attributes like unit and data range; for the proposed framework, the unit attribute is used to store units of glossary parameters. A set of variables are defined in this designed framework (Table 3). Also, naming respects the NetCDF Climate and Forecast (CF) Metadata Convention (Eaton, Gregory, Drach, Taylor, and Hankin 2006), as dimensions naming.

5 APPLICATION

This section is devoted to the python application that implements the data framework. The ontology layer is abstracted in a Python class "ForestFire.py" and tools to create and manipulate instances are available in Python Scripts. Any simulation or observation can be represented by an instance of the ForestFire class. These instances can be stored and loaded as NetCDF files, or created from scratch using the python interface in order to write transcoder to other file formats. As such it represents the "generator" element of the Experimental Frame.

Any instance can also be used to generate outputs, such as Google Earth KML, database record or other evaluation tool, as such it is also the "transducer" element. The next subsection presents the two main tests developed to validate the interface.

5.1 Representation of Python Script

A python script is created that contains a class ForestFire. This class consists of all attributes that are data cited in Table 3. This section is devoted to tests that we made to validate our framework in the application phase. Tests have been divided into two types: Test1 is defined as instantiating objects from ForestFire's class from scratch and Test2 is defined as using the Online Database called Promethee (ARNOULD and CALUGARU 2008) of Forest fires that occurred over the French territory.

5.2 Test1: Create Forestfire Instance From Scratch

Test1 presents the creation of an object (from ForestFire Class) from scratch, for which, an analyst must read a report and write the appropriate python script. Five forest fires have been analyzed for this test. We note them by place of event: Saint Cyprien, Corsica, France (09 August 2002 at 09:48AM); Favone, Corsica, France (07 August 2009 at 3:02PM); Sainte-Lucie, Corsica, France (29 August 2006 at 4:50PM); Suartone, Corsica, France (28 July 2003 at 3:00PM) and Opslo, Sardegna, Italy (21 August 2004 at 7:00PM).

Each wildfire's sheet contains two types of data; mandatory data that must be defined during the instantiation (name and date); and supplementary data that can be set after the instantiation (all the other data). After the initialization, object is virtually stored in a NetCDF file that is serialized runtime.

This file contains only defined dimensions and variables, corresponding for available data of the wildfire's sheet. The presentation layer verifies the transformation from ForestFire's object to our NetCDF Framework and vice-versa. Google Earth is used as a presentation tool to verify the result. The process is simple; stored data has been retrieved from file and a new ForestFire instance initialized with this data existed in the file. Finally, a function produces file compatible with Google Earth from all instances that were actually defined in this test (Figures 3 and 4).

5.3 Test2: Using Data From Promethee

In Test2, data are fetched from Promethee (ARNOULD and CALUGARU 2008), after objects are created and initialized with data automatically. As mentioned before, Promethee is an online database for forest fires in the Mediterranean region in France. Developed and launched in 1973 and covers 15 departments

Nader, Filippi, and Bisgambiglia

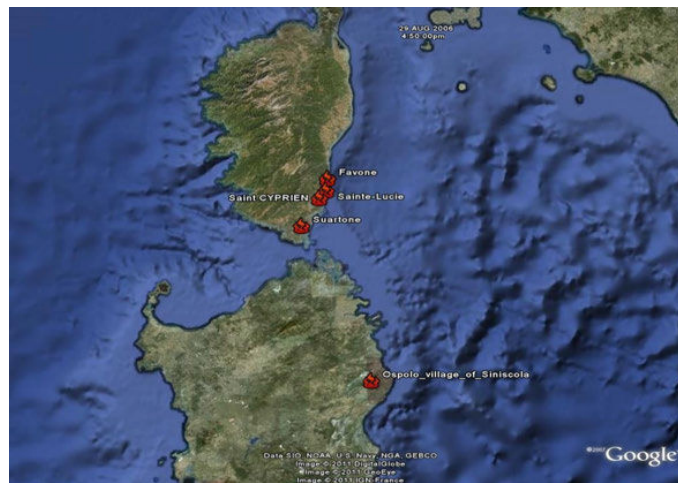


Figure 3: Shows results of Test1. Five forest fires are represented for this test.

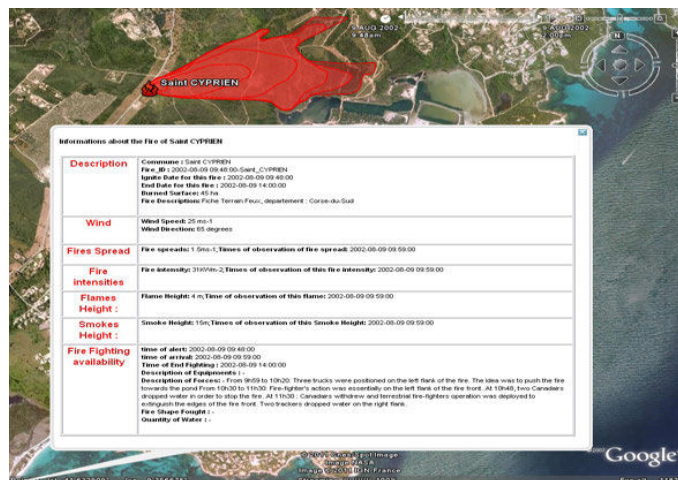


Figure 4: Represents Saint Cyprien’s forest fire, which is one of fires showed in Figure 3), with all available data (point of ignition, contours and ignition date of this fire, extended data like wind , fire Spread ,..etc).

in southeastern of France. Search Criteria are by department, municipality code, time period, contours, total burned surface, .. etc.

The purpose of this test is to retrieve a huge number of forest fires from Promethee. Extract of such is only available in ASCII comma separate values, Fires are created from scratch with a custom reader for this base. ForestFire instances are then serialized in NetCDF, offering the ability to be displayed in KML or used to start simulations. Google Earth has been used to present these results. Figures 5 and 6 show these results of retrieved forest fires (department: North of Corsica, *Haute-Corse*, period: from 01/01/2003 to 31/12/2003). Number of forest fires stored in the current NetCDF base (files) is 677 fires. Transform operation is done in 10 seconds.

Nader, Filippi, and Bisgambiglia

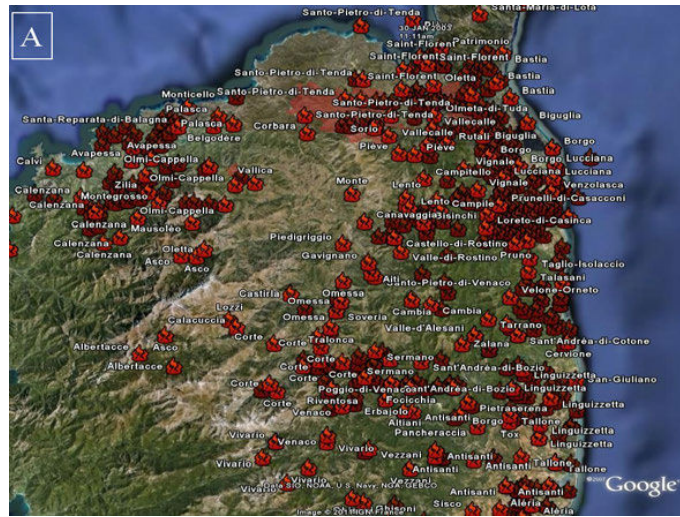


Figure 5: 677 forest fires in North of Corsica, period: from 01/01/2003 to 31/12/2003. (B): Zooming on forest fires and showing extended data of a selected forest fire.

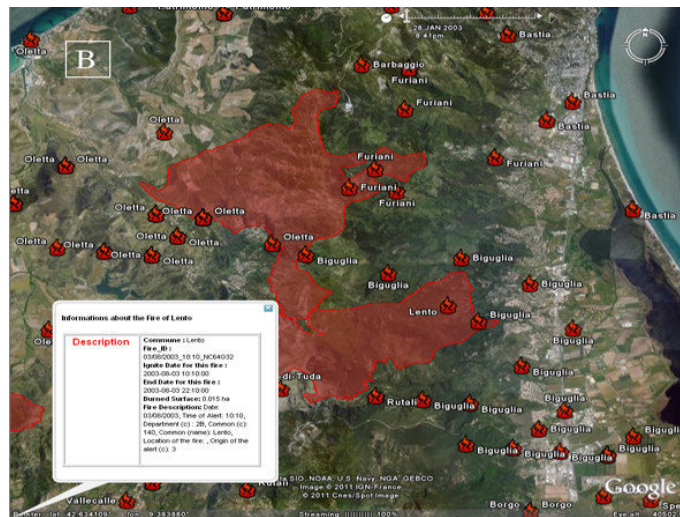


Figure 6: Zooming on forest fires (in A) and showing extended data of a selected forest fire.

6 CONCLUSION

This paper has presented the phases of the developed framework, based on NetCDF. The aim of this framework is to transform simulations results of forest fire to NetCDF files (.nc). This transformation leads to comparing forest fires simulations with corresponding observed fires and to replay stored forest fires in future for other purposes. A set of Python script is available upon request to write data reader/writer to this format.

Nader, Filippi, and Bisgambiglia

Tests have been applied to verify that this framework can represent the available data. The obtained results show that the developed framework is well suited to store forest fires simulation result. We hope to enrich the database with a maximum of events, in order to provide the test base for the scripts.

Our future work is to integrate simulation software directly in this framework in order to realize large automatic reanalysis of forest fire events, that may help to enhance forest fire models for all compliant software.

ACKNOWLEDGMENTS

This research is developed within the IDEA project ANR-09-COSI-006-01

A APPENDICES

Here we present tables of dimensions and variables of the proposed NetCDF framework. Table 2 shows dimensions of the NetCDF framework. Table 3 presents variables defined in the NetCDF framework.

Also, We present some code lines extracted from the python script presented in APPLICATION (section 5).Code lines in Listing 1, shows the creation of an instance of ForestFire from scratch (Test1) and initialization of its attributes with available data exist in the fire report. Listing 2 shows the code used to retrieve fires from NetCDF (.nc) files and generate an appropriate KML file to visualize results of this test.

Code lines in Listing 3, presents functions of Test2. First function is *load* that loads fetched forest fires data from Promethee files into an array. ForestFiretoNC function transforms an array of forest fires into NetCDF files, as done before in Test1. NewForestFireFromNC returns forest fire from a NetCDF file. Finally genKML generates a KML file to visualize the retrieved forest fires from NetCDF files on Google Earth, used before in Test1.

Listing 1: Some lines of the python script are used to initiate an object from ForestFire class and set its attributes and the addition of ForestFire instance to an array

```
Saint_CYPRIEN_Fire = ForestFire(Id = ID, descrp= 'Test1',
    department = 'Corse-du-Sud', f_name = 'Saint_CYPRIEN', commune = 'Saint_CYPRIEN')
Saint_CYPRIEN_Fire.SetBurnedSurface(45)
Saint_CYPRIEN_Fire.SetTimeOfIgnitOfFire(ignit_t)
Saint_CYPRIEN_Fire.SetTimeOfEndOfFire(end_t)
Saint_CYPRIEN_Fire.AddIgnitePoint([9.34329, 41.6446])
Saint_CYPRIEN_Fire.AddFightingTool(t_of_alert = str(time_of_alert),
    t_of_arrival = (str(time_of_arrival)), loc_fighting='', f_shape=fire_shape,
    des_of_equipements= descriptions_of_equipements, q_water=quantity_water,
    des_of_forces=descriptions_of_forces, t_of_end_fighting =str(end_t))
Saint_CYPRIEN_Fire.AddFireSpread(1.5)
Saint_CYPRIEN_Fire.AddDateTimeObsOfFireSpread(str(time_of_arrival))
fires.append(Saint_CYPRIEN_Fire)
```

Listing 2: Code presents the transformation the array of all ForestFire instances to NetCDF files; the retrieving form files; and the generation of KML file for retrieved fires

```
for i, fire in enumerate(fires):
    ForestFiretoNC('%s.nc'%(str(i)+fire.GetCommuneName()), fire)
retrieved_fires = []
for i, fire in enumerate(fires):
    retrieved_fires.append(NewForestFireFromNC('%s.nc'%(str(i)+fire.GetCommuneName())))
genKML('Test1.kml',retrieved_fires)
```

Nader, Filippi, and Bisgambiglia

Listing 3: Script presents loading fires data from Promethee files; transformation fires to NetCDF; and generating a KML file for the retrieved fires from NetCDF files

```
fires = []
fires = Load("PrometheeTest/2 B01_01_2003/data.txt", "PrometheeTest/2 B01_01_2003 /
carto.mif",
            "PrometheeTest/2 B01_01_2003 / carto.mid" )
for i, fire in enumerate(fires) :
    ForestFiretoNC( '%s.nc'%(str(i)+fire.GetCommuneName()), fire )
retrievevid_fires = []
for i, fire in enumerate(fires) :
    retrievevid_fires.append(NewForestFireFromNC( '%s.nc'%(str(i)+fire.GetCommuneName()) ))
genKML('Test2.kml', retrievevid_fires)
```

Table 2: Defined Dimensions for the proposed NetCDF framework.

Dimension	Length	Explanation
Fire Representation Text	1	It is for any fire information in text form, will be saved in NetCDF file like ID, Name and Description
Time	1	It is devoted for the time dimension (physical dimension)
X	UNLIMITED	It presents the spatial dimension in X axis (physical dimension)
Y	UNLIMITED	It presents the spatial dimension in Y axis (physical dimension)
Point 2D	2	it defines the longitude and the latitude at a given point (index dimension)
Number Of Ignition Points	UNLIMITED	It handles number of ignition points of a fire front (index dimension)
Wind	1	It presents the wind dimension used for wind speed and wind direction
Date And Time Of Fire	1	It is devoted to the date and time value of ignition/end time of fire
Info About The Fighting Tool	UNLIMITED	It defines the dimension of all information in case of Availability Of Fighting Tool
Info About Fire Passes Known Points	UNLIMITED	It defines the dimension of all information if the wild fire passes known points as a highway, .. etc.
Number Of Points In A Contour	UNLIMITED	It is a index dimension to present the number of points in a contour
Number Of Observations Of Contours	UNLIMITED	It shows the number of observations for the fire contours. Each TObs -for time of observation-is corresponding for a contour. Example: at Tobs1 = 12h00, there is contour1 and at Tobs2 = 12h25, there is contour2 in this case this dimension is 2, it means that there are 2 contours with 2 times of observation
Burned Surface	1	It is a index dimension to present the total burned surface
Fire Spread Value	1	It defines the dimension of the value of the rate of spread
Number Of Observations Of The Rate Of Fire Spread	UNLIMITED	It is a index dimension to present number of observations for the rate of a fire spread
Fire Intensity Value	1	It defines the dimension of the value of the fire intensity
Number Of Observations Of The Fire Intensity	UNLIMITED	It is a index dimension to present number of observations of the fire intensity
Smoke Height Value	1	It defines the dimension of the value of the fire smoke height
Number Of Observations Of The Smoke Height	UNLIMITED	It is a index dimension to present number of observations for the height of smoke in the air
Flame Height Value	1	It defines the dimension of the value of the fire flame height

Nader, Filippi, and Bisgambiglia

Number Of Observations Of The Flame Height	UNLIMITED	It is a index dimension to present number of observations for the height of flame of such a fire
--	-----------	--

Table 3: Defined Variables for the proposed NetCDF framework.

Variable	Type	Dimension	Unit
Fire ID	Custom	Fire Representation Text	-
Fire Name	Custom	Fire Representation Text	-
Fire Description	Custom	Fire Representation Text	-
The Arrival Time Of Fire Line	Double	X, Y	-
Wind Speed	Double	Wind	<i>m/s</i>
Wind Direction	Double	Wind	<i>degree</i>
The Ignition Time Of Fire	Double	Date And Time For A Fire	-
The Final Time Of Fire	Double	Date And Time For A Fire	-
Ignition Points Of Fire	[Double, Double]	Number Of Ignition Points, Point 2D	-
Total Burned Surface	Double	Burned surface	<i>Ha</i>
Availability Of Fighting Tool	Custom	Info About The Fighting Tool	-
Fire Passes Known Points	Custom	Info About Fire Passes Known Points	-
Front Contours	Custom	Number Of Observations Of Contours	-
A Front Contour	[Double, Double]	Number Of Points In A Contour, Point 2D	-
Table Of Fire Spread	Custom	Number Of Observations Of The Rate Of Fire Spread	-
Rate Of Fire Spread	Double	Fire spread Value	<i>m/s</i>
Table Of Fire Intensity	Custom	Number Of Observations Of The Fire Intensity	-
Fire Intensity	Double	Fire intensity Value	<i>kW/m</i>
Table Of Smoke Height	Custom	Number Of Observations Of The Smoke Height	-
Smoke Height	Double	Smoke Height Value	<i>m</i>
Table Of Flame Height	Custom	Number Of Observations Of The Flame Height	-
Flame Height	Double	Flame Height Value	<i>m</i>

Custom means a Compound Data type. Known data types are Float, Double, String, ..etc. Compound Data Type defines a collection of data types; by example *Availability of fighting tool* is a variable with a Compound Data type which consists of: Time of Alert [data type : String (text)], Time of Arrival [data type : String (text)], Description of Equipments [data type : String (text)], Description of Forces [data type : String (text)], Fire shape fought [data type : String (text)], Time of End Fighting [data type : String (text)] and Quantities of Water used [data type : String (text)].

REFERENCES

ARNOULD, P., and C. CALUGARU. 2008. "Incendies de forêts en Méditerranée : le trop dit, le mal dit, le non dit = Wildfire around the Mediterranean : too much said, or too badly, and too much left unsaid". *Forêt méditerranéenne* 29 (3): 281–296.

Balbi, J. H., F. Morandini, X. Silvani, J. B. Filippi, and F. Rinieri. 2009. "A physical model for wildland fires". *Combustion and Flame* 156 (12): 2217–2230.

Dumond, Y., S. Garitat, F. Gosse, E. Michelis, P. Simonpieri, and P. Tran. 2009. "Asphodèle-business-process oriented software for fighting forest fires.". *Revue Forestière Française* 61 (1): 9–19.

Eaton, B., J. Gregory, B. Drach, K. Taylor, and S. Hankin. 2006. "NetCDF Climate and Forecast (CF) Metadata Conventions".

Nader, Filippi, and Bisgambiglia

- Filippi, J. B., and J. H. Balbi. 2009. "Validation of a physics-based fire spread model and simulation method on a large wildfire accident". *Int. J. of Wildland Fire*, to appear.
- Filippi, J. B., F. Morandini, J. H. Balbi, and D. R. Hill. 2010. "Discrete Event Front-tracking Simulation of a Physical Fire-spread Model". *Simulation* 86 (10): 629.
- Finney, M. A. 1994a. "FARSITE: a fire area simulator for fire managers". In *the Proceedings of The Biswell Symposium, Walnut Creek, California*.
- Finney, M. A. 1994b. "Modeling the spread and behavior of prescribed natural fires". In *Proceedings of the 12th Conference on Fire and Forest Meteorology*, edited by S. of American Foresters, 138–143.
- Finney, M. A. 2006. "An overview of FlamMap fire modeling capabilities". In *Fuels Management-How to Measure Success: Conference Proceedings*, 213–220.
- Finney, M. A., and K. C. Ryan. 1995. "Use of the FARSITE fire growth model for fire prediction in US National Parks". In *The International Emergency Mgt. and Engineering Conf.: globalization of emergency management and engineering, national and international issues concerning research and applications*, 183–189.
- Flannigan, M. D., B. D. Amiro, K. A. Logan, B. J. Stocks, and B. M. Wotton. 2006. "Forest fires and climate change in the 21 st century". *Mitigation and Adaptation Strategies for Global Change* 11 (4): 847–859.
- Keane, R. E., J. L. Garner, K. M. Schmidt, D. G. Long, J. P. Menakis, and M. A. Finney. 1998. "Development of input data layers for the FARSITE fire growth model for the Selway-Bitterroot Wilderness Complex, USA". *Notes of U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station*:66 p.
- Miller, C., P. B. Landres, and P. B. Alaback. 2000. "Evaluating risks and benefits of wildland fire at landscape scales". *Proceedings—Crossing the millennium: Integrating spatial technologies and ecological principles for a new age in fire management comp. LF Neuenschwander and KC Ryan, eds*:78–87.
- Ottmar, R. D., D. V. Sandberg, C. L. Riccardi, and S. J. Prichard. 2007. "An overview of the Fuel Characteristic Classification System-Quantifying, classifying, and creating fuelbeds for resource planning". *Canadian Journal of Forest Research* 37 (12): 2383–2393.
- Rew, R. K., and G. P. Davis. 1990. "The unidata netCDF: Software for scientific data access". In *Sixth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Anaheim, California, American Meteorology Society*, 33–40.
- Rollins, M. G., R. E. Keane, Z. Zhu, J. Menakis, W. J. Hann, and A. J. Shlisky. 2003. "LANDFIRE: A nationally consistent and locally relevant interagency fire, fuels, and risk assessment". In *Proceedings of 2nd International wildland fire ecology and fire management congress*, 16–20.
- Service, U. S. F. 1998. *FARSITE: Fire Area Simulator—Model Development And Evaluation, Research Paper RMRS-RP-4, March 1998*. s.n.
- Zeigler, B. P. 1987. "Hierarchical, modular discrete-event modelling in an object-oriented environment". *Simulation* 49 (5): 219.
- Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic Press.

AUTHOR BIOGRAPHIES

BAHAA NADER is preparing a PhD, in the University of Corsica, France. His main research concern the incertitude of forest fires simulations in ForeFire. In 2009 he received the BSc in Computer and network engineering from the French University in Egypt, Egypt. In the same year he received a Master degree in computer science from University of Corsica, France.

JEAN BAPTISTE FILIPPI is a full time Researcher at the SPE CNRS laboratory of the university of Corsica. He has developed the ForeFire wildfire simulation software and is now leading a national project to provide coupled atmospheric/wildfire software.

PAUL ANTOINE BISGAMBIGLIA is professor of computer science at University of Corsica, Corsica (France) and head of the UMRS CNRS 6134. His research interests include multi-modeling activities and the use of various techniques combination in order to develop a framework for complex natural systems. He use a modeling and simulation formalism based on a discrete event specification (DEVs) proposed by B.P. Zeigler.

4. La base de simulations

La base de données des simulations est basée sur une simulation systématique d'un ensemble d'observations sauvegardées dans la base.

Les conditions initiales pour générer cette simulation sont prises à partir des fichiers NetCDF d'observations obtenues à travers l'API - Fire2NetCDF.

L'algorithme suivant présente la procédure de construction de la base de simulations à partir de la simulation systématique de N fichiers au format proposé (NetCDF) de cas observés. L'implantation de ces travaux a été réalisée dans le cadre du stage d'Hajer Ayed au sein du laboratoire CLIME de l'INRIA [Hajer A., 2011].

4.1. Algorithme de la simulation systématique

L'algorithme présente les tâches de chaque entité du cadre expérimental proposé à la simulation des feux de forêt pour assurer la construction de la base de simulations par simulation systématique. Nous détaillons le Générateur, l'Accepteur, le Transducteur et le script Python développé pour monter le cadre proposé (Figure 3. 8).

4.1.1. Côté générateur du CE

Le générateur fournit les données initiales nécessaires pour lancer les simulations – simulation systématique - des cas des feux observés. Les chemins – des données présentées au-dessus – sont définis à partir d'un fichier au format texte écrit en langage (Lua).

Ces données sont :

- **Une liste des cas d'observation** : ces cas sont les différents incendies observés. La source de données est la base de données de Prométhée et ONF. Elles sont sous forme de fichiers.

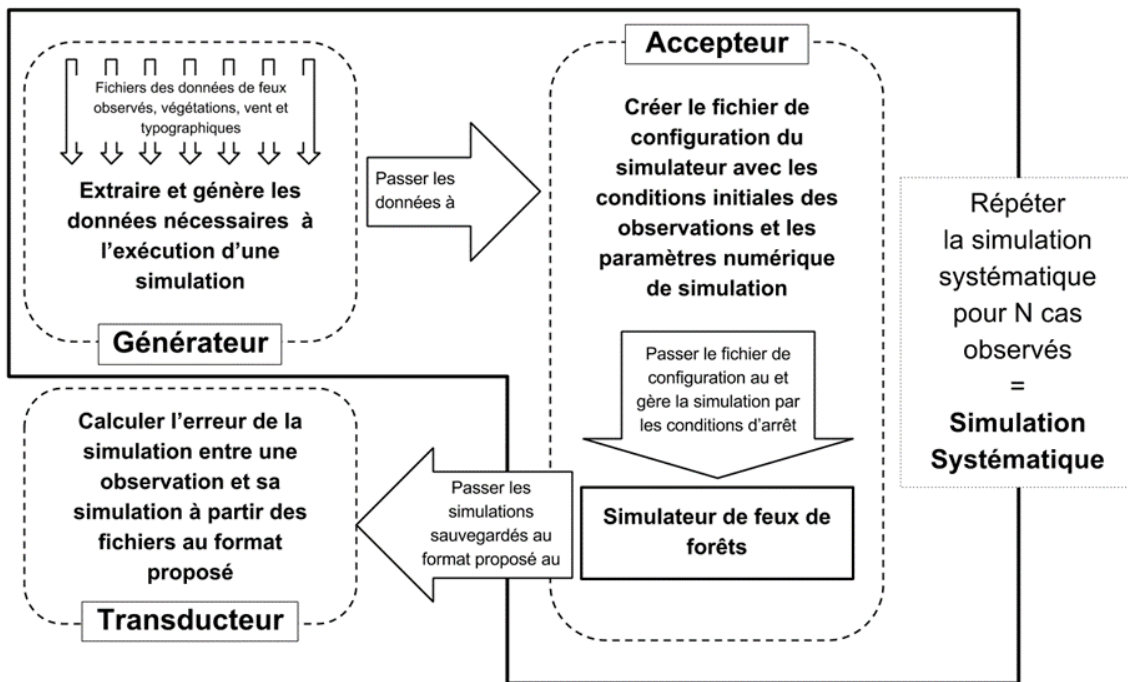


Figure 3. 8. L'algorithme de la construction de la base de simulations

- **Élévation** : la topographie du domaine.
- **Végétation** : décrit les types de végétation existant à l'endroit où le feu a été déclenché.
- **Données climatiques** : présentent les informations relatives au climat.
- **Ortho-photographie** : image aérienne ou satellite de la surface terrestre rectifiée géométriquement.
- **Couverture du territoire** : décrit la couverture végétale du domaine.

Ces conditions initiales des cas des observations sont extraites à partir des fichiers NetCDF. Ces conditions génératrices de la simulation sont associées aux données appropriées de la configuration comme les données relatives au vent, à l'élévation, à la végétation et à la sortie du code de génération du champ de vent WindNinja (vent). Le vent n'était pas perturbé dans tous ces tests ; la perturbation du vent était dès lors toujours égale à zéro (0). Avant de lancer la

simulation du ForeFire, le Générateur génère un fichier, créé pour chaque simulation (présenté en annexe). Il est nommé :

- **Fichier de configuration (.xml)** : Un fichier en format texte, conditions initiales génératrices d'une simulation, (observation, durée de simulation et résolution).

4.1.2. Côté accepteur du CE

L'Accepteur gère la simulation selon les conditions d'arrêt. Si une condition est vérifiée, l'Accepteur donne la main au Transducteur.

4.1.3. Côté transducteur du CE

Les sorties d'une simulation sont les données d'un état final du feu simulé. Le Transducteur assure l'analyse des données obtenues pour filtrer les simulations admises et rejetées. De même, il évalue ces données par celles d'observation en fournissant un pourcentage d'erreur de la simulation par cas. Les travaux et les résultats de ces évaluations seront détaillés dans le prochain chapitre.

4.2. Le champ « vent »

Les calculs des champs de vent pour une application des feux de forêt sont effectués grâce au code WindNinja [Ayad H., 2011]. Ce dernier nécessite des données d'élévation pour le domaine de la modélisation, la direction, la magnitude du vent et la spécification de la végétation dominante dans le domaine.

4.3. La simulation systématique

L'algorithme détaillé au début de ce chapitre est implémenté dans un code (un script Python) pour obtenir la simulation systématique. Le script « simulator.py » qui monte le Générateur et l'Accepteur du cadre expérimental proposé [Ayad H., 2011].

Une boucle est faite sur la liste des cas d'observations pour extraire les données et initialiser les variables des informations de la base pour :

- Créer les répertoires de sortie (un répertoire par feu)
- Déterminer le domaine du feu (la surface totale brûlée, les points d'allumage, la date d'allumage et le domaine de la simulation)
- Extraire les données de l'élévation
- Extraire les données des combustibles du ForeFire
- Extraire les données de la couverture du territoire
- Extraire les données d'ortho-photographie
- Extraire les données de la couverture du territoire
- Générer de la distribution du Vent
- Générer du domaine de végétation.

Quand l'Accepteur génère un fichier de configuration de simulation – ForeFire – et lance la commande de ce simulateur en respectant les paramètres numériques du fichier de la configuration construit, l'Accepteur gère l'expérimentation (la simulation) selon ses variables de contrôle d'exécution (les conditions d'arrêt de simulation).

À la fin de la simulation, les résultats obtenus sont sérialisés par le simulateur dans des fichiers (.nc et .kml).

5. Conclusion

Dans ce chapitre, nous avons pu formaliser le cadre expérimental proposé aux logiciels de simulation de feux de forêt. Cette formalisation permet de cadrer conceptuellement l'exigence d'expérimenter et d'évaluer ces logiciels pour la meilleure performance possible. La formalisation a été faite selon DEVS qui est adapté aux systèmes complexes comme la propagation des feux de forêt.

Tout au long du processus de formalisation du cadre expérimental proposé, nous avons interrogé des différents acteurs afin d'identifier les cas d'utilisation, les formats et l'environnement de développement de ce cadre.

Un diagramme de cas d'utilisation a été réalisé par la suite. Ce diagramme sert à définir les actes et les tâches techniques de développements qui sont présentés et mis en place à partir d'une base de données de cas d'études. Cette base est nécessaire afin de tester les méthodes et de lancer des calculs sur un grand nombre d'incendies. Cependant, aucune ontologie ni format de fichier n'existe pour décrire les données d'observation incendies.

Un travail de définition de format a donc été nécessaire, comprenant un lexique et un algorithme et son API nommée « Fire2NetCDF ».

Le conteneur choisi, NetCDF, permet une intégration facile dans d'autres environnements. Il a aussi été nécessaire de créer les modalités d'accès à ces données : service d'indexation, d'importation et d'exportation (Format Google Earth, SIG). Une base a été convertie à ce format à partir de données parcellaires venant de la base Prométhée et des services forestiers. L'API est désormais disponible au téléchargement par la communauté.

Le chapitre suivant décrit les comparaisons effectuées entre différents scores d'erreurs de différentes simulations obtenues avec l'implantation informatique du cadre proposé.

CHAPITRE 4

RESULTATS D'EVALUATION DES MODELES

Comme vu dans les chapitres précédents, l'évaluation du modèle de simulation nécessite la comparaison des données prévues avec les données observées.

Une étape fondamentale pour évaluer la performance d'un modèle est d'être capable d'évaluer les résultats d'une simulation quelconque par rapport à l'observation associée. Ces méthodes sont présentées dans le chapitre 3, compilées et étendues dans [Filippi et al., 2014].

Un résultat essentiel est qu'une seule valeur ne peut pas être représentative de la performance, nécessitant une subjectivité humaine pour permettre une meilleure compréhension de la performance du modèle.

Néanmoins, le problème de l'évaluation de la performance du modèle doit être abordé puisque il est important de savoir si un paramétrage ou une nouvelle formulation est préférable à une autre, permettant ainsi de poursuivre le processus d'amélioration des modèles, des codes et des données.

Ce chapitre, présenté sous forme de publication, est une réponse à cette problématique. Nous proposons une méthode permettant d'obtenir une indication relative de performance entre modèles. Nous allons, pour cela, nous baser sur l'agrégation de 80 différents cas réels, simulés de manière automatique au sein du cadre expérimental proposé dans le chapitre 2.

Quatre modèles sont ici évalués en utilisant le même code de simulation : un modèle très empirique, deux modèles quasi-physiques et un modèle quasi-physique avec une formulation non stationnaire.

CHAPITRE 4 - RESULTATS D'EVALUATION DES MODELES

Ces quatre modèles sont présentés dans la première section de la publication, suivie du détail de la méthodologie de l'étude et de l'analyse des résultats. L'agrégation des différents scores obtenus par les modèles démontre que l'approche utilisant de nombreux cas est de nature à permettre la classification de ces modèles. Si la méthode proposée ne donne pas de mesure absolue de la performance d'un modèle, nous montrons qu'elle discrimine si une formulation est relativement meilleure qu'une autre.

1. Publication

Publiée dans « **Natural Hazards and Earth System Sciences (NHESS)** » sous le nom « *Evaluation of forest fire models on a large observation database* ».



Evaluation of forest fire models on a large observation database

J. B. Filippi¹, V. Mallet^{2,3}, and B. Nader¹

¹Centre National de la Recherche Scientifique, Sciences Pour l'Environnement, Université di Corsica, BP 52, 20250 Corte, France

²Institut National de Recherche en Informatique et en Automatique, BP 105, 78 153 Le Chesnay CEDEX, France

³Centre d'Enseignement et de Recherche en Environnement Atmosphérique (joint laboratory École des Ponts ParisTech & EDF R & D, Université Paris Est), Marne-la-Vallée, France

Correspondence to: J. B. Filippi (filippi@univ-corse.fr)

Received: 16 April 2014 – Published in Nat. Hazards Earth Syst. Sci. Discuss.: 8 May 2014

Revised: 30 September 2014 – Accepted: 1 October 2014 – Published: 27 November 2014

Abstract. This paper presents the evaluation of several fire propagation models using a large set of observed fires. The observation base is composed of 80 Mediterranean fire cases of different sizes, which come with the limited information available in an operational context (burned surface and approximative ignition point). Simulations for all cases are carried out with four different front velocity models. The results are compared with several error scoring methods applied to each of the 320 simulations. All tasks are performed in a fully automated manner, with simulations run as first guesses with no tuning for any of the models or cases. This approach leads to a wide range of simulation performance, including some of the bad simulation results to be expected in an operational context. Disregarding the quality of the input data, it is found that the models can be ranked based on their performance and that the most complex models outperform the more empirical ones. Data and source codes used for this paper are freely available to the community.

gives limited insights on all aspects of performance, while the analysis of a human eye provides a better understanding of what was good and what went wrong.

Nevertheless, the problem of evaluating model performance must be tackled as it is important to know if a parameterisation or a new formulation is superior, and to continue the process of enhancing models, codes and data. From an operational point of view, it also appears that models are used with a clear lack of systematic evaluation, as noted recently by Alexander and Cruz (2013). A major step in such model evaluation is to compare observed rate of spread (ROS) with simulated ROS, as many data exist in the literature. Cruz and Alexander (2013) carried out such a comparison with the clear and reassuring conclusion that well-built empirical and semi-empirical models may provide a good ROS approximation. Our study focuses on the use of these models to simulate the overall two-dimensional fire spread and its corresponding burned area. Whilst evaluating the absolute model performance is, as yet, out of the question, it is proposed here to evaluate specific model performance. This specific evaluation will be linked to a typical model usage and to a territory. The typical usage proposed corresponds to the plausible “first guess” case, where only an ignition location is known with no direct observation of the wind or fuel moisture near the fire. The selected area is the Mediterranean island of Corsica, where numerous wildfires occur every year in a variety of configurations. The test consists of running simulations with four different models and a large number of observations, and compiling the results in the form of comparison scores between simulated and observed fires. These simulations must be run in a fully automated manner,

1 Introduction

Model evaluation requires comparing predicting observed values and is critical to establish the model's potential errors and credibility. The first step in evaluating model performance is to be able to evaluate single simulation results against observations. Such scoring methods have been the subject of several studies, initiated by Fujioka (2002) and recently compiled and extended in Filippi et al. (2013). Basically, it is clear from all these studies that a single value cannot be representative of a model performance, as it only

without observation biases introduced by manual adjustment of fuel, wind or ignition location in order to enhance results. The overall results are the distributions of scores, using different scoring methods; this fulfills the goal of ranking the models according to their specific use.

The four different models are presented in the first section along with the simulation method used to compute the front propagation. The second section details the evaluation method, data preprocessing and numerical set-up. The results are presented and discussed in the last section, along with focuses on specific cases.

2 Models description

Fire propagation modelling can refer to a vast family of codes, formulations, systems or even data sets (Sullivan, 2009a, b). As this study focuses on the evaluation of large scale fire simulation, our selected definition of “fire model” is the formulation of the fire-front velocity. A velocity is obviously not enough to obtain fire progression and burned areas. A fire-front solver code and input data are needed. These two are the same for all models and described in the next section. Note that the proposed model selection is unfortunately not exhaustive of all existing formulations, but rather focuses on representing some kind of evolution in the model types.

Depending on their complexity, the models can take into account the terrain slope, the atmospheric properties (wind velocity v , air density ρ_a and temperature T_a), a spatial characterisation of the fuels (mass loading σ , density of alive/dead $\rho_{l,d}$, height e , surface to volume ratio S_v , emissivity ϵ_v and moisture content m defined as the fraction of water over total weight) and the fuel combustion properties (ignition temperature T_i , calorific capacity $c_{p,v}$, combustion enthalpy Δh , stoichiometry s and mass exchange rate, due to pyrolysis $\dot{\sigma}$). Each model prognoses the fire-front velocity V in the normal direction to the front \mathbf{n} , pointing towards the unburned fuel.

2.1 3 % model

The first and most simple model makes the major assumption that the fire is propagating at 3% of the wind velocity, as long as there is fuel available, regardless of the vegetation changes or the terrain slope. In practice, in order to compute the velocity everywhere on the fire front, the wind normal to the front $W_s = v \cdot \mathbf{n}$ is taken here as wind velocity and $V = 0.03 W_s$. This “rule of thumb model” is sometimes used by firefighters (with caution, because of its lack of reliability). Here, it will serve the purpose of hopefully being the lowest reference in terms of performance.

2.2 Rothermel model

The quasi-empirical Rothermel model (Rothermel, 1972) forms the basis of the United States National Fire Danger

Rating System and fire behaviour prediction tool (BEHAVE) (Andrews, 1986). It builds on earlier works from Byram and Fons (1952) and is based on a heat balance developed by Frandsen (1971). It highly relies on a set of parameters obtained from wind tunnel experiments in artificial fuel beds (Rothermel and Anderson, 1966) and Australian experiments (McArthur, 1966). This model is also widely used in Mediterranean countries for various purposes (fire risk and behaviour), but usually requires some adjustments by experts in order to be fully efficient on all fuel types. Since this study is created as a blind test for each model, these adjustments were not performed here. The default values given in Rothermel (1972), such as moisture of extinction ($M_\chi = 0.3$) and mineral damping ($\eta_s = 1$) were therefore used. Since its first developments, the Rothermel model has not changed in its formulation, but users have adapted coefficients and added optional sub-models to fit specific cases. In this paper, we have used the latest revision from its original team of authors (Andrews et al., 2013) that includes, in particular, a revised wind speed limit function to lower the spread rate dependence on strong winds.

This quasi-physical model also uses a number of fitted parameters (in US customary units) that read

$$V = \frac{I_r \xi (1 + \phi_v + \phi_p)}{(\rho_d 250 + 1.116 \text{ m exp}(-138/S_v))}, \quad (1)$$

with reaction intensity

$$I_r = R' \sigma \Delta H \eta_s (1 - 2.59(m/M_\chi) + 5.11(m/M_\chi)^2 - 2.59(m/M_\chi)^3). \quad (2)$$

A propagating flux ratio is given by

$$\xi = (192 + 0.2595 S_v)^{-1} \exp(0.792 + 0.681 \sqrt{S_v} (\beta + 0.1)). \quad (3)$$

The wind factor is

$$\phi_v = 7.47 \exp(-0.133 S_v^{0.55}) (\beta / \beta_{op})^{-0.715 \exp(-3.59 \times 10^{-4} S_v)} \quad (4)$$

$$W_s^{0.02526 \times S_v^{0.54}}.$$

The slope factor is

$$\phi_p = 5.275 \beta^{-0.3} \alpha^2. \quad (5)$$

An optimal packing ratio with

$$\beta_{op} = 3.338 S_v^{-0.8189}. \quad (6)$$

The packing ratio is given by

$$\beta = \rho_d / \rho_l. \quad (7)$$

The maximum reaction rate is

$$R'_{Max} = S_v^{1.5} \times (1 / (495 + 0.0594 S_v^{1.5})). \quad (8)$$

The reaction rate is given as

$$R' = (R'_{\text{Max}}(\beta/\beta_{\text{op}}))^A \exp(A(1 - \beta/\beta_{\text{op}})), \quad (9)$$

where

$$A = 1/(4.774S_v^{0.1} - 7.27). \quad (10)$$

In (Andrews et al., 2013), a new wind limit function is imposed over the original model as

$$W_s^{\text{max}} = 96.3I_r^{1/3}. \quad (11)$$

2.3 Balbi model

The Balbi model (Balbi et al., 2009), like Rothermel, can be classified as a quasi-physical model. Its formulation is based on the assumption that the front propagates as a radiating panel in the direction normal to the front. The model verifies that, for a specific wind, terrain and fuel configuration, the absorbed energy equals the combustion energy directed toward the unburned fuel. This energy is the sum of “radiant part” from the flame and a “conductive” part within the fuel layer. The assumption is also made that only a given portion χ_0 of the combustion energy is released as radiation, because the flame is viewed as a tilted radiant panel with an angle γ towards the unburned fuel. The equation governing the propagation velocity of the front reads

$$V = V_0(\epsilon_v, T_i, e, \sigma, m, T_a) + \chi_0 \Delta h \dot{\sigma} f(\lambda, \gamma), \quad (12)$$

where $V_0 = \epsilon_v B T_i^4 e / 2\sigma [c_{p,v}(T_i - T_a) + m \Delta h_w]$ is the contribution of the vegetation undergoing pyrolysis (B is the Boltzmann constant and Δh_w is the water evaporation enthalpy). The second term accounts for the propagation by radiation and reads

$$f(V, \gamma) = \frac{R}{2 + \mu \tau \cos \gamma} (1 + \sin \gamma - \cos \gamma) \mathcal{H}_{R^+}(\gamma), \quad (13)$$

where \mathcal{H}_{R^+} is the Heaviside function for positive reals and μ is an evolution coefficient of the ratio between radiated energy and released combustion energy. The volume–surface ratio is noted S_v , and τ is the burning duration given by the Andersen model (Anderson, 1969). The flame tilt angle γ depends on the slope angle α and wind \mathbf{v} projected onto \mathbf{n} the front normal vector:

$$\tan \gamma = \tan \alpha + \rho_a(\mathbf{v} \cdot \mathbf{n}) / 2(1 + s)\dot{\sigma} \quad \text{with } \mathbf{n}. \quad (14)$$

A major assumption of the Balbi and Rothermel models is that the fire is always travelling at a stationary speed that verifies $V = \kappa/\tau$, and that all energy is absorbed within the fuel bed for the computation of V_0 . Because of these assumptions, front velocity is not dependent on the local fire state (previous intensity, front curvature, depth). It cannot accelerate or go to extinction, and it is only dependent on the local

fuel, wind and terrain properties. These assumptions are required to compute a priori potential rate of spread without knowing explicitly the local front depth λ or its curvature κ , such as in the BEHAVE tool. Later versions of Rothermel added sub-models for acceleration or extinction. A more fundamental approach was developed for the Balbi model with a non-stationary formulation.

2.4 Balbi non-stationary model

By using the front tracking solver, local front depth λ and curvature κ are always available as numerical diagnostics of the front. The introduction of these variables in the model was rather simple as it removed strong assumptions. The updated formulation reads

$$V = V_0(\epsilon_v, T_i, e, \lambda, \sigma, m, T_a) + \chi_0 \Delta h \dot{\sigma} f(\lambda, \gamma), \quad (15)$$

with

$$V_0 = \left(1 - e^{-\frac{\lambda \beta_d}{4}}\right) \epsilon_v B T_i^4 e / 2\sigma [c_{p,v}(T_i - T_a) + m \Delta h_w], \quad (16)$$

and β_d a radiation dumping ratio that depends on fuel packing. The second term removes the requirement for the stationary speed with

$$f(\kappa, \lambda, \gamma) = (1 - \cos \kappa) \frac{\lambda}{2 + \mu \lambda \cos \gamma} (1 + \sin \gamma - \cos \gamma) \mathcal{H}_{R^+}(\gamma). \quad (17)$$

The main disadvantage of the model is that it is now tight to a solver able to locally and constantly diagnose λ and κ with a reasonable numerical cost, introducing, in the process, some additional numerical errors. The solver used for the study is the front tracking code “ForeFire”.

2.5 Fire propagation method

The fire propagation solver ForeFire (Filippi et al., 2009) uses a front tracking method that relies on a discretisation with Lagrangian markers. The outward normal \mathbf{n}_i of marker i defines the direction of propagation, and v_i is the local front speed. The maximum distance d_m allowed between two consecutive markers is called the perimeter resolution. If two markers are further away than this distance, a remapping of the front is carried out in order to keep the resolution constant. A filtering distance $d_f = d_m/2$ is also needed to avoid the over-crossing of two markers and potential inversion of the norm. The advection scheme is a first-order Euler scheme in space:

$$\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^{(n)} + \delta l \cdot \mathbf{n}_i^{(n)}, \quad t_i^{(n+1)} = t_i^{(n)} + \frac{\delta l}{v_i^{(n)}}, \quad (18)$$

where the $\cdot^{(n+1)}$ refers to the value of a variable at the next step, and \cdot_i the marker index. Spatial increment δl determines

the resolution of the front propagation and should be smaller than the smallest space scale influencing the fire propagation, which are usually fire breaks, such as roads; i.e. in typical simulations $\delta l \approx 1$ m.

3 Evaluation method

A selection of 80 fire cases have been compiled into an observation database for this study. For each fire, the required initial data are preprocessed to generate the initial conditions and the data required by the selected propagation model. The simulations are then run by distributing the computation of the different cases. Once a simulation result is available for an observation/simulation pair, the comparison is computed with the scores introduced in Sect. 3.3.

3.1 Observation database

The observation database is composed of 80 fires that all occurred in the Mediterranean island of Corsica between 2003 and 2006. These cases were extracted from the Prométhée database (<http://www.promethee.com/>), a repository of wild-fire observations managed by the “Institut Géographique National” (<http://www.ign.fr/>). Data within this database offer precise burned surface for many wildfires, along with an information on the ignition date and approximate location. Nevertheless, this information cannot always be trusted, so each case was reviewed with a field expert in order to validate ignition points and date, which reduced the amount of relevant observations. The selection of the Corsica island was made because field expertise was available, as well as adequate data and homogeneity in fire dynamics, given the relatively limited data set (mostly shrubs and Mediterranean maquis). Fire sizes in the selection ranged from 1 hectare to several hundreds of hectares in order to be representative of all potential model uses.

3.2 Data preprocessing

The first step in order to launch a simulation is to compile and format data in a way that they can be processed by the fire propagation solver. For the fire simulation code ForeFire, the input data are composed of a configuration file, an elevation field, one or several fields of wind direction and speed, a land use field and a fuel field.

The design of the study implied that the data were not tailored for any model or test case. All of the preprocessing is thus automatically done for the ignition location and date. We are aware that this automatic generation will generate input data that might seem extremely unrealistic compared to the observation. In particular, wind direction and fuel state can be significantly different from the real values. For the considered fires, or for any fire that may ignite at anytime, anywhere, the exact inputs to the simulation models are not

Table 1. Values of the parameters for main fuel types.

	Shrubs	Pine	Mixed	Maquis
σ (kg)	0.6	2	1.5	0.8
e (m)	0.5	3	2.5	2
m (%)	8	17	12	10
Δh (MJ kg ⁻¹)	18.6	16	17	18.6

observed or stored. This study tries to rely on the best available data provided in an operational context.

The simulation configuration defines the size of the domain, the date and the numerical set-up. The rest of the input data are extracted from data files with the same domain extent and date using the Geospatial Data Abstraction Library (GDAL) and its tool `ogr2ogr` (GDAL development team, 2013) using a conformal projection.

The elevation field originates from the Institut Géographique National, Base de Données ALTIude (IGN BDAlti) at a 25 m resolution. It is originally available in DEM ASCII format (Digital Elevation Model, American Standard Code for Information Interchange).

The wind field used in the simulation is extrapolated with the “WindNinja” mass consistent code (Forthofer, 2007) from station data. The extrapolation is especially expected to rectify the wind velocity in valleys. WindNinja inputs consist of the elevation field and the wind station data. The wind field is outputted at the same resolution as the elevation field and is used for the whole duration of the simulation (under the strong assumption that the wind does not change direction during the fire).

WindNinja input data originate from the nearest of 12 automatic 10 m high weather stations of the Météo France network, relatively evenly distributed across the island. The selection algorithm is very simple and ranks the distance from the ignition point to all stations, selecting the closest, and uses the nearest date and time available after the reported ignition time. It would definitely have been more relevant to select the closest upstream station, but such data are unfortunately completely unavailable for most of the fires because of the poor density of the station network. The closest method was thus selected because it was thought to be the most adapted first guess in an automatic, operational context.

Fuel distribution data are taken in vector format (shape file) from the Institut Géographique National, Institut français de l’environnement (IGN IFEN), with locally 10 fuel classes corresponding to the main burning vegetation in Corsica. Fuel parameters are derived from Anderson et al. (1981) for the mixed/grass and pine forest types, while the “Proterina” parameterisation (Santoni et al., 2011) is used for the shrubs and maquis class. Among the 10 possible fuels, these four classes (shrubs, pine, mixed, maquis) were the only burning fuels across all cases, with a large majority of maquis. Values for these fuels are found in Table 1.

As little as is known about the fuel state, the fuel moisture is set as moderate to high water stress for every fuel model, which corresponds to high fire danger on a summer's day.

3.3 Comparison scores

Our evaluation relies on scoring methods that were analysed (and, for two of them, proposed) in Filippi et al. (2013). We denote $\mathcal{S}(t)$ as the burned surfaces at time t in the simulation. The final simulation time is t_f . At the end of the observed fire t_f^o (superscript "o" stands for observation), the observed burned surface is $\mathcal{S}(t_f^o)$. Ω is the simulation domain. The arrival time at point X is denoted as $\mathcal{T}(X)$ for the simulation and $\mathcal{T}^o(X)$ for the observation. Finally, the area of a surface \mathcal{S} is denoted as $|\mathcal{S}|$. We assume that the ignition time is 0.

We relied on the following scores, with $t_u = \max(t_f, t_f^o)$:

$$\text{Sørensen similarity index} = \frac{2|\mathcal{S}^o(t_u) \cap \mathcal{S}(t_u)|}{|\mathcal{S}^o(t_u)| + |\mathcal{S}(t_u)|}, \quad (19)$$

$$\text{Jaccard Similarity Coefficient} = \frac{|\mathcal{S}^o(t_u) \cap \mathcal{S}(t_u)|}{|\mathcal{S}^o(t_u) \cup \mathcal{S}(t_u)|}, \quad (20)$$

$$\text{Kappa Statistics} = \frac{P_a - P_e}{1 - P_e}, \quad (21)$$

where

$$P_a = \frac{|\mathcal{S}^o(t_u) \cap \mathcal{S}(t_u)|}{|\Omega|} + \frac{|\Omega \setminus (\mathcal{S}^o(t_u) \cup \mathcal{S}(t_u))|}{|\Omega|}, \quad (22)$$

and

$$P_e = \frac{|\mathcal{S}^o(t_u)| |\mathcal{S}(t_u)|}{|\Omega|^2} + \frac{|\Omega \setminus \mathcal{S}^o(t_u)| |\Omega \setminus \mathcal{S}(t_u)|}{|\Omega|^2}, \quad (23)$$

$$\begin{aligned} \text{ATA} = 1 - & \frac{1}{|\mathcal{S}(t_f) \cup \mathcal{S}^o(t_f^o)| \max(t_f, t_f^o)} \\ & \left[\int_{\mathcal{S}(t_f) \cap \mathcal{S}^o(t_f^o)} \max(\mathcal{T}(X) - \mathcal{T}^o(X), 0) dX \right. \\ & + \int_{\mathcal{S}(t_f) \setminus \mathcal{S}^o(t_f^o)} \max(t_f^o - \mathcal{T}(X), 0) dX \\ & \left. + \int_{\mathcal{S}^o(t_f^o) \setminus \mathcal{S}(t_f)} (t_f - \mathcal{T}^o(X)) dX \right], \quad (24) \end{aligned}$$

$$\text{SA} = 1 - \frac{1}{t_f} \quad (25)$$

$$\left[\int_{|0, t_f^o|} \frac{|\mathcal{S}(t) \setminus \mathcal{S}^o(t_f^o)|}{|\mathcal{S}(t)|} dt + \int_{|t_f^o, t_f|} \frac{|\mathcal{S}^o(t_f^o) \setminus \mathcal{S}(t)|}{|\mathcal{S}^o(t_f^o)|} dt \right]. \quad (26)$$

The Sørensen similarity index and Jaccard similarity coefficient compare the final simulated and observed areas.

Kappa statistics compute the frequency with which simulation agrees with observation (P_a), with an adjustment (P_e) that takes into account agreement by chance. The arrival time agreement and the shape agreement both take into account the dynamics of the simulation. Even though there is only one observation at the end of the fire (i.e. the final burned surface), these scores were designed to partially evaluate the time evolution of the simulation dynamics. Further details on these scores may be found in Filippi et al. (2013).

3.4 Simulation set-up

The four models were run using the ForeFire code. In all, 80 cases resulted in a total of 320 simulations. Each simulation is set-up using the ignition point that defines where to carry out the simulation. The simulation domain is centered on the ignition point. In north-south and east-west directions, the domain size is about four times the extension of the fire. The spatial increment δl depends on the fire size. If the final observed burned area is A , then $\delta l = \max(1, \log_{10} A - 4)$ m, if A is in m^2 . The filtering distance d_f is set to $20\delta l$.

All simulations were carried out at most until the burned area equals the observed final burned area. In practice, on small fires, the area burned in the simulation may be larger because the stopping criterion is checked every 7 min (so the over-development cannot be more than 7 min of fire propagation). The simulation can also stop earlier if the front velocity is 0 everywhere (stopped fire).

4 Results

In this section, we investigate the performance of the models on all fire simulations. All simulations are concisely described in the Appendix, with plotted contours (simulations and observation). The performance of each model for an individual case is hardly analysed since it can vary a lot, depending on the quality of the data. We essentially draw conclusions that are supported by the overall performance. Moreover, it is expected that a large number of scores will be very low, mostly because of the poor input data (whose quality cannot be known in advance). We have decided not to drop these low scores to be representative of actual model performance in an operational context.

4.1 Distribution of the scores

An important aspect of the comparison is to select and understand the way scores are presented. At first, let us consider the distribution of the 80 scores for each model and scoring method. The 80 scores are sorted here into an ascending order and plotted. It is important to note that the sorting is carried out independently for each model. Therefore, the number in the abscissae corresponds to a rank in the list of sorted simulations (per model), but not to the same

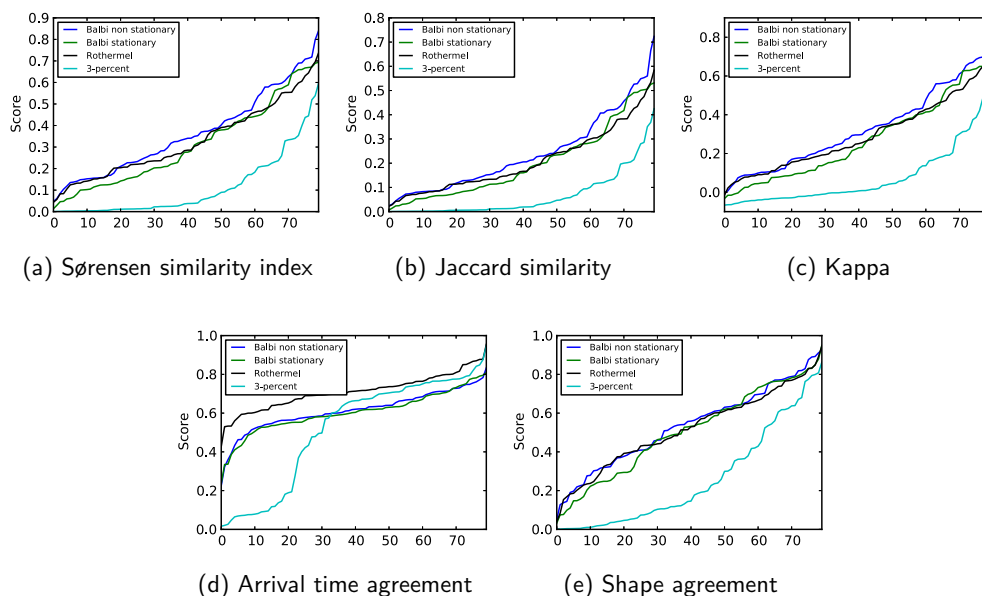


Figure 1. The distribution of the scores for the 80 cases and for the four models. The sorting was carried out independently for each model, hence one cannot compare individuals scores.

fire case in all four distinct evaluated models. Figure 1 shows the distributions for all scoring methods. The distributions for the classical scores – the Sørensen similarity index, Jaccard similarity coefficient and kappa coefficient – are very similar with worst score distributions for the 3 % model. The Balbi model gives significantly better results. The Rothermel model arguably brings additional improvements over the Balbi simulations for the lower scores in the distribution. The non-stationary Balbi model clearly provides the best overall results for this data set. Less clear ranking is found with the distributions of shape agreements (Fig. 1e), except that the 3 % model shows poor performance. Rothermel model is a clear first in the distributions of the arrival time agreements (Fig. 1d), with a close 3 % model for the higher scores. We point out that all observed fires are assumed to last 24 h at most (because no data for the duration of the fire were found), which is a very rough approximation of the actual duration of the fire. In these conditions, arrival time agreement is a less relevant scoring method than the other scores, as it strongly penalises over-prediction. Nevertheless, it points out that the Rothermel model is less prone to over-prediction than all the other models. This is probably due to its wind limit function, which slows down the front and is generally diffusive.

In Fig. 1, the scores are sorted in ascending order, independently for each model. It is therefore impossible to compare the performance for each individual fire and to make sure that a model consistently shows better performance. The score distributions are shown in Fig. 2 with an ascending sorting based on the mean score across the models. The sorting is

therefore the same for all models and their scores remain paired in the plot. There is clearly a large variability in the performance of the models at each individual fire. There is, however, an overall ascending tendency and, more importantly, for most fires, the non-stationary Balbi model shows the best performance, with Rothermel coming second and the 3 % model being the worst. It means that there is a consistent improvement for all types of fires and conditions. The conclusion is not so clear concerning the arrival time agreement, since it may lack relevance in this context. This scoring method requires a precise fire duration, while we assume that all fires lasted 24 h. Considering all scores, the large variance in the models' performance suggests that the generation of large ensembles of simulations may be needed in the simulation process of a single fire. Indeed, with a single simulation for a single fire and poor data quality, a poor simulation is likely to occur.

Based on all the distributions, one conclusion is that the non-stationary Balbi model consistently gives the best results. The second model is the Rothermel, with stationary Balbi as a close third. It is worth noting that the switch to a non-stationary model, where the fire depth is taken into account, leads to significant improvements, at least as high as the changes due to the fire spread rate. This suggests that the representation of the fire in the numerical model is a key aspect of a simulation and makes it more “physical” by removing assumptions to the model can enhance its versatility and performance. This suggests that more physics is needed to attain state-of-the-art model performance, even in a context

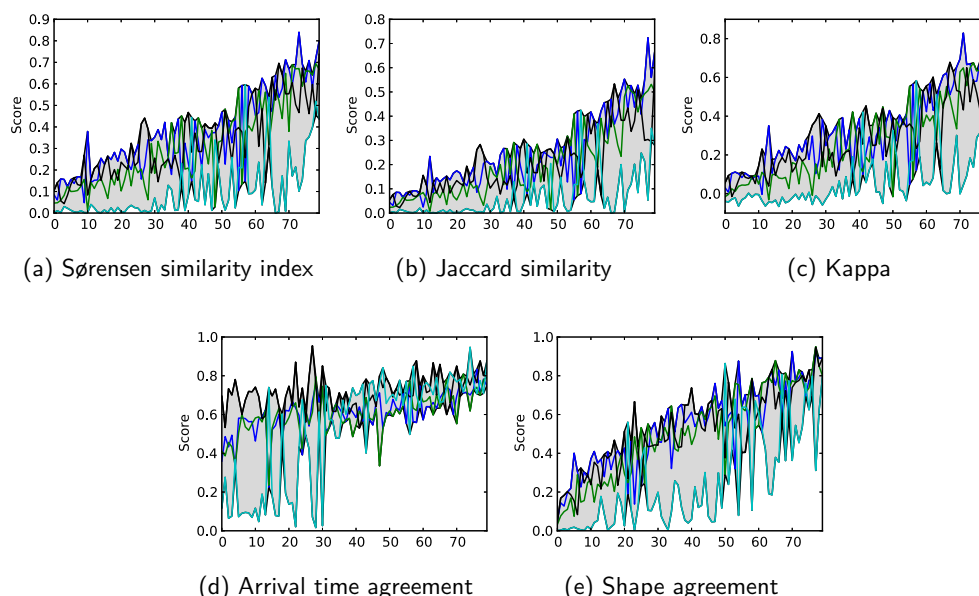


Figure 2. The distribution of the scores for the 80 cases and for the four models. The sorting was carried out with respect to the mean score across the models. Hence, at a given abscissa, one can read the four scores for a given fire case. The grey area is delimited by the upper and lower bounds of the lines.

where the input data may be of poor quality. Overall, it is interesting to note that, despite the poor data quality (which we might expect in operational forecasts), the models can be ranked, hence even a poor database can help objective model development.

4.2 A look at the individual simulations

For many simulations, the input data are very poor, hence a dramatically low performance. When the data are probably reliable, the models may perform reasonably well, which is detected by the scoring methods. For each scoring method and model, we found the simulation with the highest performance across the 80 fires. These selected cases are shown in Table 2.

The Sørensen, Jaccard and kappa scores essentially identify the same “best” cases. It is consistent with the strong similarities found between the score distributions for these scoring methods (Sect. 4.1). The dynamic aware scores (arrival time agreement and shape agreement) select a variety of fire cases. It is more difficult to understand the reason for these selections from the final contours, since these scores take into account the full dynamics. Figures A1–A3 present all simulation results of the data set. It can be observed that there are obvious numerical biases in these simulation results. In particular, the smaller fires (< 1 ha, such as R3 Alata, or O0 Propriano) are systematically over-predicted by both Balbi models that tend to predict higher propagation velocities. Since the area criterion to stop the simulation is checked every

7 min, the fire growth becomes relatively large considering the small fire size. Nevertheless, as all simulations were to be handled with the same method, the same delay was kept for all cases. Note that we included these much smaller fires in the test database because they are the most frequent fires to be fought.

Human supervision in these simulations would have helped to rectify the automatic wind data or misplaced ignition point that clearly appear to be erroneous in many of these simulations. Such examples of supervised rectification are carried out for the case “C2” in the Appendix – Ghisonaccia (95 ha). The results before and after tweaking the parameters can be seen in Fig. 3. The nearest station apparently provided a bad estimate of a very local wind direction. Ignition location data also appeared to be slightly misplaced in the original report as it was not directly in the main propagation axis. Hence, the wind direction and the ignition location were manually adjusted, in order to better reproduce the final burned area.

In an operational context, the burned area is not known in advance, but better results can still be obtained with better models. Let us consider more closely two fires that appear in Table 2: the Oletta case (21 August 2004; N3 in the Appendix) and the Santo Pietro di Tenda (1 July 2003; B0 in the Appendix). In Fig. 4, we plotted the former case. In this case, it is clear that the Balbi model shows a much better performance than the other models. The overall shape of the final simulated contour by Balbi model has a similar aspect

Table 2. The best simulations, according to the scoring method and the fire model. “ATA” stands for arrival time agreement. “Shape” refers to shape agreement. The final contours may be found in Figs. A1–A3, and further information about the fires is available in Table A1.

Score	Non-stationary Balbi	Stationary Balbi
Sørensen	N3 – Oletta (21 Aug 2004)	N1 – Barbaggio (1 Jul 2003)
Jaccard	N3 – Oletta (21 Aug 2004)	N1 – Barbaggio (1 Jul 2003)
Kappa	N3 – Oletta (21 Aug 2004)	B0 – Santo Pietro di Tenda (1 Jul 2003)
ATA	N3 – Oletta (21 Aug 2004)	H0 – Corbara (1 Jul 2003)
Shape	N3 – Oletta (21 Aug 2004)	B0 – Santo Pietro di Tenda (1 Jul 2003)
Score	Rothermel	3 %
Sørensen	J3 – Olmeta di Tuda (21 Aug 2004)	B1 – Calenzana (30 Jun 2005)
Jaccard	J3 – Olmeta di Tuda (21 Aug 2004)	B1 – Calenzana (30 Jun 2005)
Kappa	J3 – Olmeta di Tuda (21 Aug 2004)	B1 – Calenzana (30 Jun 2005)
ATA	G2 – Loreto di Casinca (20 Apr 2005)	M3 – Lumio (12 Aug 2003)
Shape	B0 – Santo Pietro di Tenda (1 Jul 2003)	M3 – Lumio (12 Aug 2003)

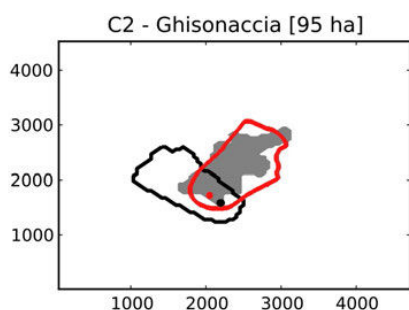


Figure 3. Case of the Ghisonaccia fire (20 August 2003; C2 in the Appendix) as observed (burned area in grey) and simulated by the Rothermel model with raw automatic (black) and corrected (red) parameters. Coloured dots correspond to the ignition points.

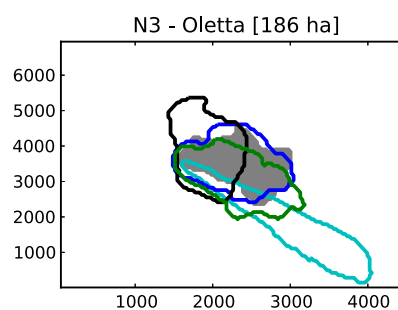


Figure 4. Case of the Oletta fire (21 August 2004; N3 in the Appendix) as observed (burned area in grey) and simulated by the non-stationary Balbi model (blue), the stationary Balbi model (green), the Rothermel model (black) and the 3 % model (cyan).

ratio than the observed one, while the Rothermel model and the 3 % model show large over-burning at the rear and the head, respectively, of the fronts. So, even when the wind direction is correct, there can still be large differences between the model simulations. When we only consider the Balbi simulations for this fire case, we might conclude that a model can reach a good prediction skill once proper input data are provided. Nevertheless, the performance of the other models questions the reliability of the input data. It is likely that the input data might be erroneous or compensating for deficiencies in the Balbi model. It is fairly possible that the Balbi model would underperform with the exact data. This suggests that exposing the actual prediction skill of fire simulation requires the use of several models and the exploration of the full probability distribution of the uncertain data.

In Fig. 5, we plotted the case for which the Balbi model and Rothermel model attained their best shape agreement. The visual agreement with the observed contour does not seem as good as in the previous fire case. Hence the dynamics of the fire must have significantly contributed to the

shape agreement. This suggests that taking into account the front dynamics in the evaluation can cast a different light on a model.

4.3 Forecast reliability

In an operational context, the ability of a model to predict that a location will burn can be of high importance. One way to assess the reliability of a model in this context is to evaluate its burn probability. When the model predicts that a fire will burn in a given location, we compute the frequency with which this actually happens. Conversely, we are interested in the frequency with which a model correctly predicts that a location will burn. We refer to this indicator as the detection skill. The detection skill may be perfect if the model burns very large regions: any location that is actually burned in reality will be burned in the simulation as well. On the contrary, the first indicator (burn probability) will be perfect if the fire stops right after it started; at the ignition point, the fire was observed, so the location was burned, just like

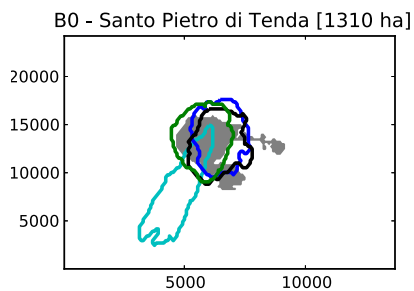


Figure 5. Case of the Santo Pietro di Tenda fire (1 July 2003; B0 in the Appendix) as observed (burned area in grey) and simulated by the non-stationary Balbi model (blue), the stationary Balbi model (green), the Rothermel model (black) and the 3 % model (cyan).

Table 3. Burn probability and detection skill for the four models, all fires included. The burn probability is the frequency with which the model correctly predicts that a location will be burned in reality. The detection skill is the frequency with which a burned location (in observations) is also burned in the simulation.

Indicator	Non-stationary Balbi	Stationary Balbi	Rothermel	3 %
Burn probability	0.29	0.25	0.31	0.11
Detection skill	0.41	0.35	0.36	0.11

the model “predicted”. Consequently, we computed both the burn probability and the detection skill. See Table 3.

The burn probability and detection skill are consistent with previous results. The non-stationary Balbi is the best model in terms of detection. The Rothermel model is, however, slightly better for the burn probability. These are followed by the (stationary) Balbi model that is close to the Rothermel model in terms of detection skills. When the Rothermel model predicts that a location will be burned, the real fire burns it in 31 % of all cases. If the real fire reached a given location, the non-stationary Balbi model would predict it in 41 % of all cases. It is noteworthy that the performance can vary a lot from one model to another. It is obvious that the 3 % model with a detection skill of 11 % is of very low value compared to the 41 % of the non-stationary Balbi model.

Overall, the performance may not be good enough for a model to be reliable in operational applications. The performance spread among the models suggests that further development on the model has the potential to immensely improve the results. Nonetheless, it is clear from the simulation results that the input data play a key role. In particular, an erroneous wind direction spoiled so many simulations that good local meteorological forecasts alone would be likely to dramatically improve performance.

5 Conclusions

The objective of this paper was to evaluate fire spread models and their relevance in a realistic operational context with limited information. We considered a database of 80 fires whose final burned surfaces were observed. We simulated these fires in a purely automated manner, using only a poor data set that may be available in this operational context. The meteorological values were taken at the closest observation station, even though the actual wind direction at the exact fire location may be significantly different. The vegetation cover and the associated fuel load were not tuned by any means for any cases or models.

Despite the crude application setting, we were able to rank these fire spread models. Overall, the non-stationary version of the Balbi model gave the best results. The higher relative performance of the non-stationary Balbi model was observed across most of the 80 cases. The stationary version of the Balbi model showed a significantly lower performance. This suggests that the numerical treatment of the fire front is a key aspect, just like the rate of spread. The Rothermel model (Andrews et al., 2013), with a new imposed wind speed limit function, was ranked second. Finally, we observed that the most empirical model, taking 3 percents of the wind velocity, is clearly not a good option. Overall, it is estimated that current fire spread models may benefit from a better physical description, even with poor data quality. In order to strengthen this conclusion, we can add that the performance of the original Rothermel model (whose results were not reported here) is significantly lower than that of the version used in this work. The original Rothermel model consistently performs significantly worse than the stationary Balbi model.

We evaluated the skill of the models to forecast that a certain location will be burned. We also evaluated whether the locations burned in a simulation are likely to be burned in reality. In both cases, there is a wide spread in the models’ skills. In addition, the skill of the best model may not be high enough for a reliable use in operational context. Therefore, we conclude that further work on the physical models is still needed, and improved input data, especially for the wind direction, are obviously required. For instance, largest gains in performance may be obtained from local micro-meteorological forecasts that properly forecast wind direction.

Considering the high uncertainties in both models and the input data, the simulation context is barely deterministic. There is a need for probabilistic approaches, such as those developed by Finney et al. (2011). With model results being so variable, a promising direction may be the use of ensembles of models, together with perturbed input data. Such developments could be used for uncertainty quantification, risk assessment and ensemble-based forecasting.

Appendix A: Simulations for all fires

This section shows the simulation results of the four models for all considered fires. Information about the fires is collected in Table A1. The results are displayed in Figs. A1–A3.

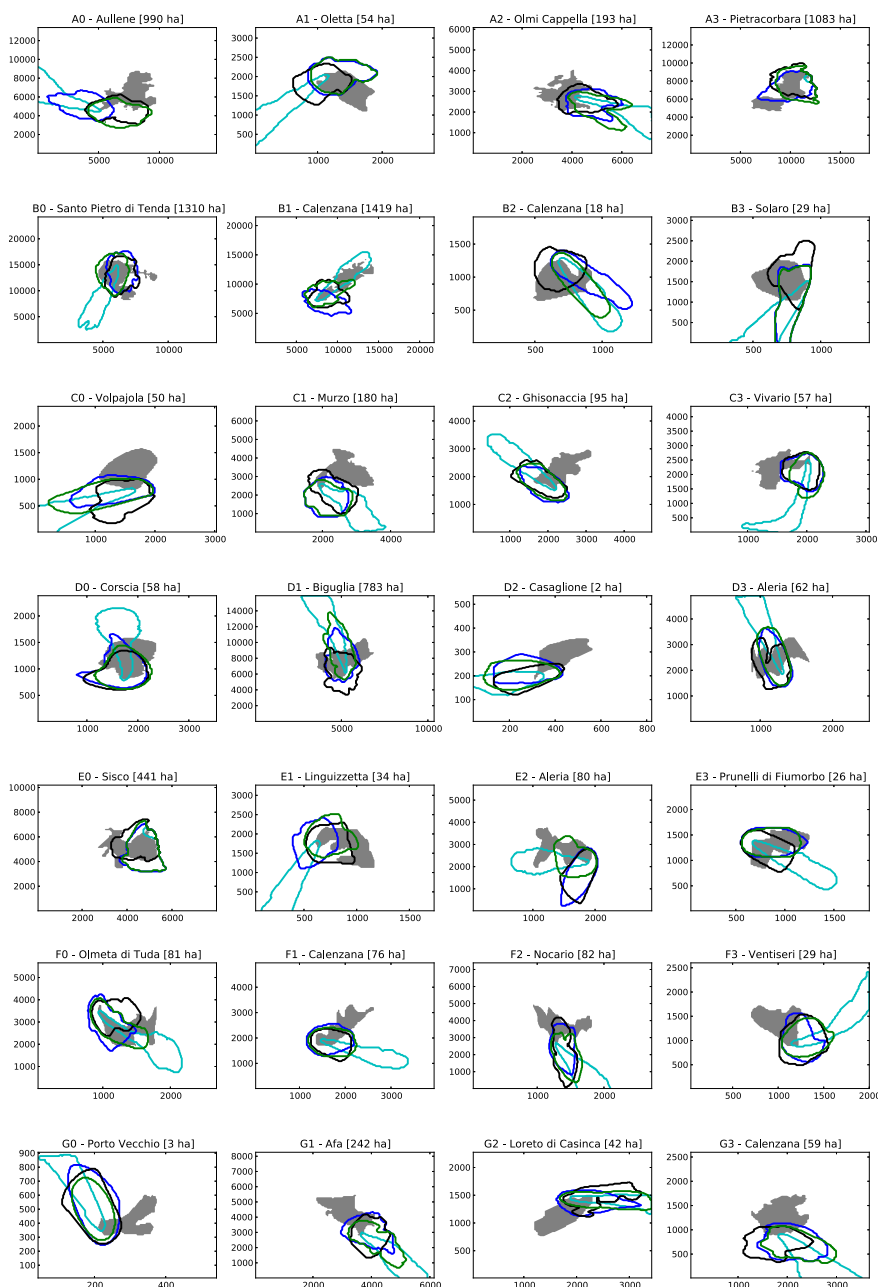


Figure A1. Simulation for all fires (1/3). The non-stationary Balbi model is in blue; the stationary Balbi model is in green; the Rothermel model is in black; the 3 % model is in cyan. The grey area is the observed final burned area. The coordinates are in metres.

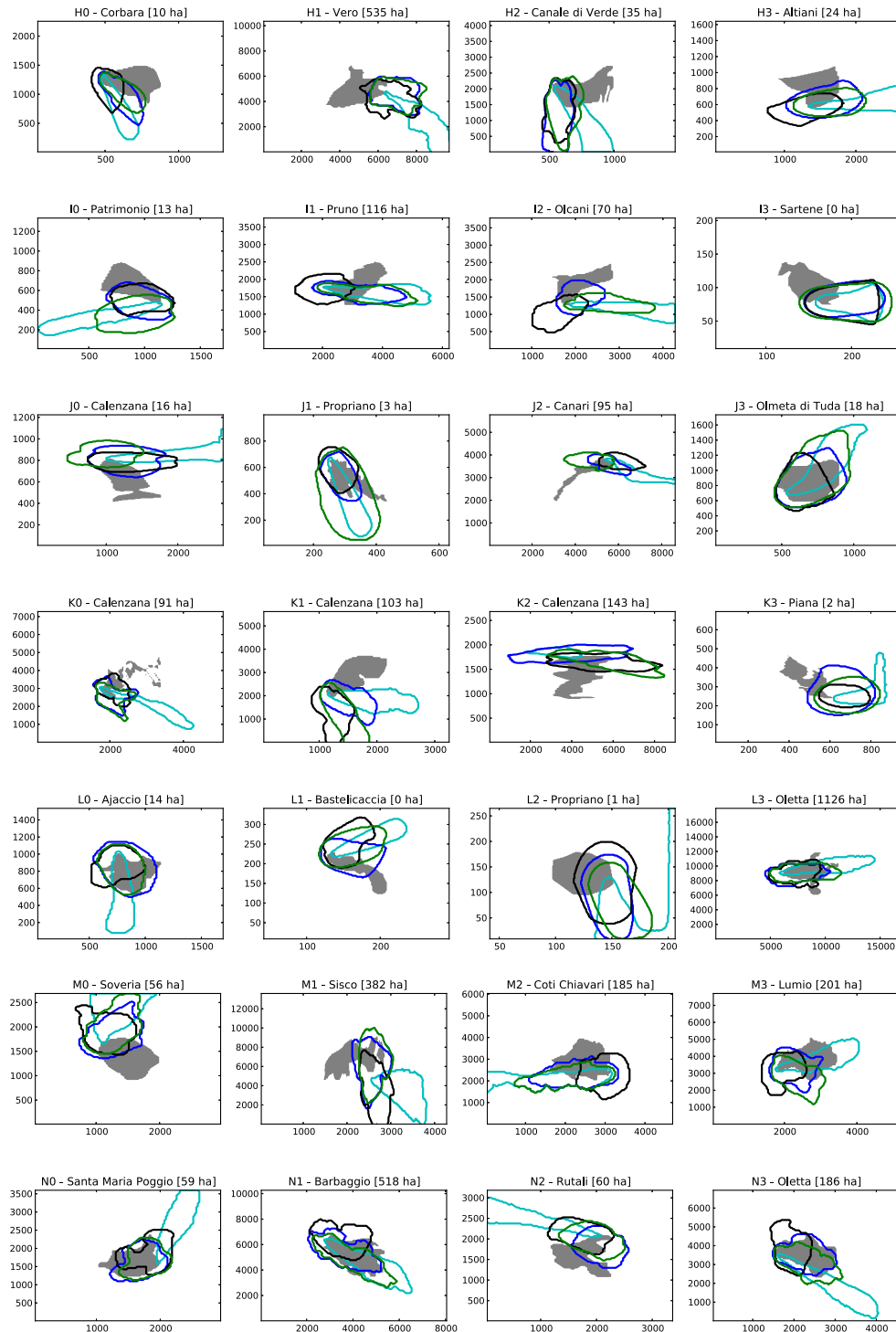


Figure A2. Simulation for all fires (2/3). The non-stationary Balbi model is in blue; the stationary Balbi model is in green; the Rothermel model is in black; the 3 % model is in cyan. The grey area is the observed final burned area. The coordinates are in metres.

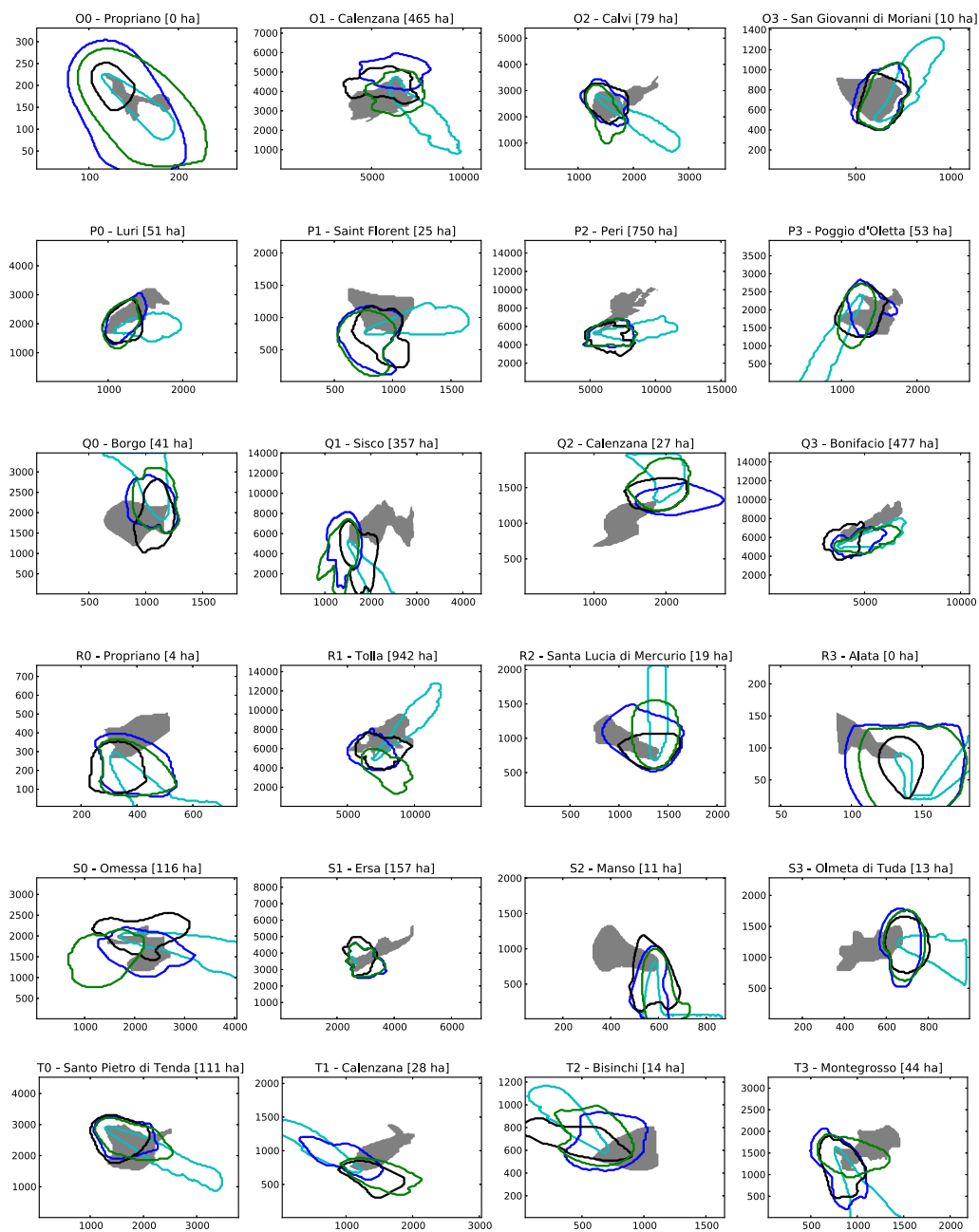


Figure A3. Simulation for all fires (3/3). The non-stationary Balbi model is in blue; the stationary Balbi model is in green; the Rothermel model is in black; the 3 % model is in cyan. The grey area is the observed final burned area. The coordinates are in metres.

Table A1. Information about all the 80 fires shown in Figs. A1–A3. The size is the final burned area in hectares. WS stands for wind speed, in m s^{-1} . WD is the wind direction in degrees, defined clockwise, with 0 corresponding a westerly wind (90 southerly). The resolution in metres is in column “Res.”. The final three columns represent the best scores (among the four models) found for Jaccard similarity coefficient (BJ), arrival time agreement (BA) and shape agreement (BS).

#	Name	Date	Size	WS	WD	Res.	BJ	BA	BS
A0	Aullene	7 Aug 2003	990 ha	9	60	133 m	0.16	0.64	0.66
A1	Oletta	9 Aug 2003	53 ha	5	130	27 m	0.29	0.76	0.64
A2	Olmi Cappella	29 Aug 2003	192 ha	6	229	60 m	0.28	0.71	0.59
A3	Pietracorbara	13 Aug 2004	1082 ha	19	260	139 m	0.47	0.8	0.63
B0	Santo Pietro di Tenda	1 Jul 2003	1310 ha	3	150	136 m	0.53	0.88	0.95
B1	Calenzana	30 Jun 2005	1418 ha	6	320	214 m	0.42	0.85	0.77
B2	Calenzana	19 Jul 2005	17 ha	5	220	13 m	0.27	0.74	0.7
B3	Solaro	13 Jan 2004	28 ha	2	129	13 m	0.46	0.78	0.6
C0	Volpajola	21 Jun 2003	49 ha	5	130	23 m	0.15	0.69	0.3
C1	Murzo	9 Aug 2003	180 ha	5	210	52 m	0.076	0.76	0.39
C2	Ghisonaccia	2 Aug 2003	95 ha	7	40	45 m	0.15	0.73	0.77
C3	Vivario	9 Jul 2007	56 ha	4	140	30 m	0.13	0.62	0.44
D0	Corscia	1 Jan 2009	57 ha	1	0	23 m	0.42	0.78	0.7
D1	Biguglia	2 Aug 2003	782 ha	7	20	103 m	0.42	0.72	0.89
D2	Casaglione	26 May 2008	2 ha	5	110	5 m	0.084	0.67	0.22
D3	Aleria	5 Sep 2004	62 ha	5	30	24 m	0.41	0.76	0.57
E0	Sisco	24 Jul 2003	441 ha	1	276	79 m	0.53	0.85	0.79
E1	Linguizzetta	9 Aug 2003	33 ha	4	140	16 m	0.32	0.8	0.64
E2	Aleria	20 Jun 2004	80 ha	4	90	28 m	0.29	0.79	0.53
E3	Prunelli di Fiumorbo	4 Jul 2003	26 ha	2	242	18 m	0.53	0.8	0.77
F0	Olmata di Tuda	21 Aug 2004	80 ha	7	240	26 m	0.4	0.77	0.77
F1	Calenzana	19 Jul 2005	75 ha	8	250	38 m	0.28	0.79	0.83
F2	Nocario	11 Apr 2006	81 ha	4	240	26 m	0.15	0.87	0.47
F3	Ventiseri	21 Aug 2004	28 ha	5	310	19 m	0.19	0.71	0.48
G0	Porto Vecchio	21 Aug 2006	3 ha	3	40	5 m	0.085	0.6	0.39
G1	Afa	22 Jul 2003	242 ha	5	220	61 m	0.16	0.7	0.67
G2	Loreto di Casinca	20 Apr 2005	41 ha	7	260	22 m	0.21	0.95	0.74
G3	Calenzana	17 Oct 2008	59 ha	7	210	25 m	0.092	0.7	0.2
H0	Corbara	1 Jul 2003	10 ha	8	210	12 m	0.28	0.81	0.88
H1	Vero	15 Jul 2003	535 ha	3	210	96 m	0.13	0.72	0.63
H2	Canale di Verde	4 Jan 2003	34 ha	3	240	14 m	0.25	0.88	0.76
H3	Altiani	12 Aug 2003	23 ha	3	280	16 m	0.21	0.73	0.56
I0	Patrimonio	11 Feb 2005	12 ha	5	120	13 m	0.24	0.74	0.41
I1	Pruno	20 Apr 2005	116 ha	7	260	37 m	0.25	0.67	0.64
I2	Olcani	1 Jan 2006	69 ha	12	260	37 m	0.23	0.66	0.4
I3	Sartene	14 Jul 2008	0.25 ha	10	270	1 m	0.13	0.69	0.35
J0	Calenzana	8 Aug 2003	16 ha	2	275	12 m	0.17	0.6	0.37
J1	Propriano	24 Aug 2008	2.8 ha	4	210	5 m	0.5	0.8	0.69
J2	Canari	1 Jul 2003	94 ha	7	239	57 m	0.2	0.66	0.76
J3	Olmata di Tuda	21 Aug 2004	18 ha	8	320	12 m	0.66	0.85	0.89
K0	Calenzana	31 Jul 2005	91 ha	8	240	50 m	0.3	0.78	0.77
K1	Calenzana	15 Jul 2004	102 ha	4	260	32 m	0.085	0.78	0.41
K2	Calenzana	30 Aug 2006	143 ha	5	40	26 m	0.26	0.69	0.67
K3	Piana	22 Aug 2008	2.2 ha	3	280	6 m	0.086	0.69	0.052
L0	Ajaccio	22 Jul 2006	13 ha	6	180	15 m	0.55	0.79	0.81
L1	Bastelicaccia	13 Jul 2008	0.3 ha	5	300	2 m	0.27	0.67	0.46
L2	Propriano	19 Jul 2006	0.5 ha	3	190	1 m	0.27	0.59	0.51
L3	Oletta	2 Jul 2003	1126 ha	7	280	168 m	0.47	0.8	0.81
M0	Soveria	29 Aug 2003	56 ha	5	350	26 m	0.17	0.72	0.41
M1	Sisco	14 Aug 2003	382 ha	1	258	42 m	0.24	0.62	0.31
M2	Coti Chiavari	6 May 2003	185 ha	7	110	46 m	0.31	0.67	0.65
M3	Lumio	12 Aug 2003	201 ha	3	280	51 m	0.37	0.95	0.86

Table A1. Continued.

#	Name	Date	Size	WS	WD	Res.	BJ	BA	BS
N0	Santa Maria Poggio	3 Aug 2003	58 ha	1	321	29 m	0.55	0.77	0.88
N1	Barbaggio	1 Jul 2003	517 ha	7	239	80 m	0.56	0.78	0.83
N2	Rutali	14 Jan 2004	59 ha	1	70	31 m	0.23	0.65	0.83
N3	Oletta	21 Aug 2004	186 ha	7	240	44 m	0.72	0.83	0.93
O0	Propriano	27 Aug 2008	0.25 ha	5	230	2 m	0.29	0.74	0.56
O1	Calenzana	25 Jul 2004	465 ha	6	200	72 m	0.41	0.77	0.76
O2	Calvi	8 Jul 2009	78 ha	9	240	36 m	0.48	0.86	0.83
O3	San Giovanni di Moriani	4 Aug 2003	10 ha	4	320	10 m	0.53	0.88	0.82
P0	Luri	13 Jan 2004	50 ha	2	270	26 m	0.46	0.75	0.67
P1	Saint Florent	5 Aug 2003	25 ha	3	284	17 m	0.25	0.84	0.48
P2	Peri	23 Jul 2009	749 ha	2	280	152 m	0.066	0.76	0.6
P3	Poggio d'Oletta	11 Sep 2004	53 ha	4	140	26 m	0.38	0.82	0.51
Q0	Borgo	20 Jul 2003	40 ha	1	20	17 m	0.26	0.75	0.65
Q1	Sisco	14 Aug 2003	357 ha	4	238	43 m	0.089	0.72	0.32
Q2	Calenzana	6 Sep 2004	27 ha	10	20	19 m	0.055	0.7	0.17
Q3	Bonifacio	28 Jul 2003	477 ha	8	280	104 m	0.19	0.64	0.65
R0	Propriano	19 Jul 2006	3.5 ha	4	210	7 m	0.14	0.73	0.29
R1	Tolla	9 Aug 2003	942 ha	2	325	146 m	0.26	0.7	0.7
R2	Santa Lucia di Mercurio	6 Sep 2004	19 ha	7	0	20 m	0.27	0.74	0.63
R3	Alata	21 Jul 2008	0.12 ha	6	230	1 m	0.092	0.78	0.18
S0	Omessa	31 Aug 2003	115 ha	9	255	33 m	0.42	0.53	0.63
S1	Ersa	17 Oct 2003	156 ha	3	90	69 m	0.24	0.87	0.79
S2	Manso	8 Jul 2004	11 ha	2	180	8 m	0.12	0.62	0.39
S3	Olmata di Tuda	4 Aug 2003	13 ha	1	260	9 m	0.16	0.79	0.52
T0	Santo Pietro di Tenda	1 Jul 2003	110 ha	7	239	37 m	0.49	0.81	0.83
T1	Calenzana	6 Sep 2004	28 ha	10	40	20 m	0.14	0.74	0.37
T2	Bisinchi	2 Aug 2005	13 ha	4	30	12 m	0.28	0.75	0.56
T3	Montegrosso	1 Jul 2003	43 ha	8	210	21 m	0.27	0.74	0.55

Acknowledgements. This research is supported by the Agence Nationale de la Recherche, project no. ANR-09-COSI-006, IDEA <http://anridea.univ-corse.fr/>.

Edited by: R. Lasaponara

Reviewed by: two anonymous referees

References

- Alexander, M. E. and Cruz, M. G.: Are the applications of wildland fire behaviour models getting ahead of their evaluation again?, *Environ. Modell. Softw.*, 41, 65–71, doi:10.1016/j.envsoft.2012.11.001, 2013.
- Anderson, H.: Heat transfer and fire spread, USDA Forest Service research paper INT, Intermountain Forest and Range Experiment Station, Forest Service, US Dept. of Agriculture, 1969.
- Anderson, H., Forest, I., and Range Experiment Station (Ogden, U.): Aids to determining fuel models for estimating fire behavior, General technical report INT, US Dept. of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, 1981.
- Andrews, P. L.: BEHAVE: fire behavior prediction and fuel modeling system – BURN subsystem, Tech. rep., USDA Forest Service, Intermountain Research Station, 1986.
- Andrews, P. L., Cruz, M. L., and Rothermel, R.: Examination of the wind speed limit function in the Rothermel surface fire spread model, *Int. J. Wildland Fire*, 22, 959–969, doi:dx.doi.org/10.1071/WF12122, 2013.
- Balbi, J., Morandini, F., Silvani, X., Filippi, J., and Rinieri, F.: A Physical Model for Wildland Fires, *Combust. Flame*, 156, 2217–2230, doi:10.1016/j.combustflame.2009.07.010, 2009.
- Byram, G. and Fons, W.: Thermal conductivity of some common forest fuels, Tech. rep., USDA Forest Service, Division of Fire Research, 1952.
- Cruz, M. G. and Alexander, M. E.: Uncertainty associated with model predictions of surface and crown fire rates of spread, *Environ. Modell. Softw.*, 47, 16–28, doi:10.1016/j.envsoft.2013.04.004, 2013.
- Filippi, J.-B., Morandini, F., Balbi, J. H., and Hill, D. R.: Discrete Event Front-tracking Simulation of a Physical Fire-spread Model, *Simulation*, 86, 629–646, doi:10.1177/0037549709343117, 2009.
- Filippi, J.-B., Mallet, V., and Nader, B.: Representation and evaluation of wildfire propagation simulations, *Int. J. Wildland Fire*, doi:10.1071/WF12202, 2013.
- Finney, M., Grenfell, I., McHugh, C., Seli, R., Trethewey, D., Stratton, R., and Brittain, S.: A Method for Ensemble Wildland Fire Simulation, *Environ. Model. Assess.*, 16, 153–167, doi:10.1007/s10666-010-9241-3, 2011.
- Forthofer, J.: Modeling Wind in Complex Terrain for Use in Fire Spread Prediction, Thesis, Colorado State University, 2007.
- Frandsen, W. H.: Fire spread through porous fuels from the conservation of energy, *Combust. Flame*, 16, 9–16, doi:10.1016/S0010-2180(71)80005-6, 1971.
- Fujioka, F. M.: A new method for the analysis of fire spread modeling errors, *Int. J. Wildland Fire*, 11, 193–203, 2002.
- GDAL development team: GDAL – Geospatial Data Abstraction Library, Version 1.10.0, Open Source Geospatial Foundation, available at: <http://www.gdal.org>, 2013.
- McArthur, A.: Weather and Grassland Fire Behaviour, Leaflet, Forest Research Institute, Forestry and Timber Bureau, 1966.
- Rothermel, R.: A mathematical model for predicting fire spread in wildland fuels, Tech. rep., USDA Forest Service, 1972.
- Rothermel, R. and Anderson, H.: Fire spread characteristics determined in the laboratory, US Forest Service research paper INT, Intermountain Forest & Range Experiment Station, Forest Service, US Dept. of Agriculture, 1966.
- Santoni, P., Filippi, J. B., Balbi, J.-H., and Bosseur, F.: Wildland Fire Behaviour Case Studies and Fuel Models for Landscape-Scale Fire Modeling, *J. Combust.*, pp. Article ID 613 424, 12 pp., doi:10.1155/2011/613424, 2011.
- Sullivan, A. L.: Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models, *Int. J. Wildland Fire*, 18, 349–386, doi:10.1071/WF06143, 2009a.
- Sullivan, A. L.: Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models, *Int. J. Wildland Fire*, 18, 369–386, doi:10.1071/WF06142, 2009b.

CONCLUSION GENERALE

Le travail de recherche présenté dans ce manuscrit est consacré à l'évaluation des modèles de simulation de propagation des incendies de forêt. Une revue des travaux, formats et approches, présentée en chapitre 1, nous a montré qu'aucune solution normalisée et automatisable n'était proposée dans ce champ applicatif. Nous avons ainsi décliné une approche formelle développée dans le cadre de la théorie de la modélisation et simulation DEVS [Ziegler et.al. 2002], afin d'obtenir un cadre applicatif utilisable directement dans le contexte de la simulation d'incendies. Le travail a, à cette fin, été décomposé en quatre points principaux.

- La définition d'un format de données standard, un conteneur et une bibliothèque permettant de normaliser les évènements incendies en vue de leur traitement automatisé.
- Le développement d'un cadre expérimental pour formaliser l'évaluation des modèles de simulation d'incendies.
- L'évaluation de métriques de calcul d'erreurs de simulation, la proposition de deux nouvelles méthodes et une bibliothèque de calcul de ces erreurs.
- Enfin, une application des méthodes et codes développés par l'étude des performances des modèles sur un grand nombre de cas d'incendies.

Le premier point a permis de répondre à la problématique de la normalisation des données disponibles pour les incendies de forêt. Cette étape était un préalable nécessaire au traitement de données en masse, aucun standard de fichier ni même de nomenclature n'étant disponible et/ou utilisé par les modélisateurs ou ingénieurs en charge des données incendies (observations ou simulation). Les résultats de ce premier point sont notamment le développement

CONCLUSION GENERALE

d'un ensemble de noms, d'une notation, et d'un encodage des données dans un conteneur NetCDF. Une API a aussi été proposée pour la création et la manipulation des données à ce format, facilitant ainsi l'interfaçage avec les outils existants.

Le second point, le cadre expérimental, a consisté à décliner les approches formelles d'évaluation générique de modèles dans le cadre des incendies de forêt. Il a permis de définir comment cette évaluation devait se réaliser, quels composants devaient être développés, quelles méthodes étaient développées et comment les interconnecter. Ce cadre a mis en évidence, en particulier, la nécessité de séparer clairement la génération des paramètres de simulation à partir d'une observation (objet générateur), le contrôle et la récupération des résultats de simulation (objet transducteur) et leur évaluation (objet accepteur).

Le troisième point, correspondant à l'objet accepteur, a nécessité un état de l'art spécifique afin de déterminer les métriques nécessaires à quantifier les erreurs de simulation (score de simulation). Si quatre méthodes standard ont pu être identifiées dans la littérature, nous avons pu montrer qu'elles se limitaient à la comparaison à un instant donné, ne pouvant donc rendre compte de la performance de la dynamique d'une simulation incendie. Cette problématique a été traitée en proposant deux nouvelles méthodes de calcul de scores, spécifiquement développées pour évaluer la performance d'une simulation à bien représenter l'évolution d'un feu (et non pas seulement son état final ou transitoire). Un format bien défini de données ayant déjà été développé, ces différentes méthodes d'évaluation ont été implantées au sein d'une bibliothèque de calcul de scores « fireScore », disponible en licence de code ouvert et libre.

Enfin, le quatrième point est la réalisation d'une évaluation de modèles à l'aide d'une implantation du cadre défini précédemment. Nous avons, pour cela, utilisé une base de données d'observations, et effectué 80 simulations d'incendies réels de manière complètement

CONCLUSION GENERALE

automatique. Ces simulations ont été réalisées en utilisant les données librement accessibles (météo, terrain) non modifiées par un opérateur humain. Cette automatisation et le non-ajustement de paramètres ont ainsi permis de se rapprocher au plus près du contexte opérationnel où peu d'informations locales sont disponibles, peu de temps après l'alerte de l'éclosion d'un incendie.

Quatre formulations de modèles de vitesse de front de flammes ont été, par la suite, testés dans ce contexte expérimental, allant du modèle le plus empirique au modèle quasi-physique plus complexe. Les résultats de cette étude, l'utilisation de nombreux cas, et en particulier l'agrégation de leur score, a démontré que cette approche est de nature à laisser apparaître une certaine hiérarchie des performances entre les différentes formulations de modèles de vitesse incendie.

Le cadre proposé permet donc ainsi d'évaluer la performance d'une paramétrisation ou d'une formulation relativement à une autre, sans toutefois être en mesure de donner une mesure absolue de la performance.

Perspectives

La réalisation de ce cadre expérimental destiné à la simulation incendie est la première étape nécessaire à une évaluation plus approfondie des modèles. La principale perspective de recherche, suite directe de ces travaux, concerne l'évaluation de l'incertitude des modèles. Des approches stochastiques sont, pour cela, développées, dans lesquelles de très nombreuses simulations sont effectuées en utilisant des conditions initiales perturbées. Une fois cette incertitude quantifiée, il est possible de déterminer la fiabilité des résultats de simulation et donc de permettre une appréciation objective de la confiance que l'on peut mettre dans les résultats.

CONCLUSION GENERALE

Une seconde piste concerne aussi naturellement l'utilisation de ce cadre expérimental pour l'amélioration des modèles évalués. Ce cas d'utilisation est, d'ailleurs, la principale application des cadres expérimentaux. Mis à la disposition des modélisateurs, et associé à une base de données incendie plus conséquente, il pourra permettre l'évaluation objective des performances des innovations des modèles dans un processus de constante amélioration. Ces erreurs étant normées, il est aussi envisageable de les utiliser dans un processus de minimisation ou encore d'assimilation de données afin d'améliorer les paramétrisations de ces modèles.

Enfin, la dernière perspective de recherche concerne le développement de nouveaux indicateurs de risque incendie. Basés sur de très nombreuses simulations incendies, ces nouveaux indices devront donner, au-delà du risque d'ignition, une appréciation de la taille potentielle des incendies et de la fiabilité des simulations.

CONCLUSION GENERALE

REFERENCES

- [Albini, F.A., 1979] Albini, F. A.; Baughman, R. G. 1979. Estimating windspeeds for predicting wildland fire behavior. Research Paper INT-221. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. (1,179 KB; 12 pages)
- [Albini, F.A., 1979b] Albini, F. A. 1979. Spot fire distance from burning trees – a predictive model. General Technical Report INT-56. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. (2,850KB; 73 pages)
- [Alexander et al. 1992] Alexander, M. E, B.D. Lawson, B.J. Stocks, and C.E. Van Wagner, 1992, Development and structure of the Canadian forest fire behavior prediction system. Forestry Canada Fire Danger Group, Inf. Rep. ST-X-3.
- [Arca B. et al., 2007] Arca B, Duce P, Laconi M, Pellizzaro G, Salis M, Spano D (2007), "**Evaluation of {FARSITE} simulator in Mediterranean maquis.**", International Journal of Wildland Fire 16(5), 563–572. doi:10.1071/WF06070
- [Arnould P. et al., 2008] Arnould, P., and C. Calugaru (2008), "**Incendies de forêts en Méditerranée : le trop dit, le mal dit, le non dit = Wildfire around the Mediterranean : too much said, or too badly, and too much left unsaid.**", Forêt méditerranéenne 29 (3): 281–296.
- [Balbi et al. , 2009] Balbi, J. H., F. Morandini, X. Silvani, J. B. Filippi, and F. Rinieri. 2009. "A physical model for wildland fires". *Combustion and Flame* 156 (12): 2217--2230

REFERENCES

- [Banerjee et al., 1999] Banerjee M, Capozzoli M, McSweeney L, Sinha D (1999), "**Beyond kappa: a review of interrater agreement measures.**", The Canadian Journal of Statistics/La Revue Canadienne de Statistique 27(1), 3–23. doi:10.2307/3315487
- [Cheon, 2007] Cheon, Saehoon (2007), "**Experimental frame structuring for automated model construction: Application to simulated weather generation.**", Ph.D., THE UNIVERSITY OF ARIZONA, 2007, 154 pages; 3258332
- [Cohen, 1960] Cohen J (1960), "**A coefficient of agreement for nominal scales.**", Educational and Psychological Measurement 20(1), 37–46. doi:10.1177/001316446002000104
- [Diffenbaugh et al., 2003] Filippi, J.-B. 2003, "*Une Architecture Logicielle pour la Multi-Modélisation et la Simulation à évènements Discrets de Systèmes Naturels Complexes*", PhD thesis-Université de Corse.
- [Eaton et al., 2006] Eaton, B., J. Gregory, B. Drach, K. Taylor, and S. Hankin (2006), "**NetCDF Climate and Forecast (CF) Metadata Convention.**", <http://www.cgd.ucar.edu/cms/eaton/cf-metadata/index.html>.
- [Filippi et al., 2009] Filippi, J.B., Balbi, J.H. (2009) Validation of a physics-based fire spread model and simulation method
- [Filippi et al., 2014] J.-B. Filippi, V. Mallet, and B. Nader. (2014), Evaluation of forest fire models on a large observation database, Nat. Hazards Earth Syst. Sci. Discuss., 2, 3219-3249
- [Filippi et al., 2014] Filippi Jean-Baptiste, Mallet Vivien, Nader Bahaa (2014) Representation and evaluation of wildfire propagation simulations. International Journal of Wildland Fire 23, 46–

- 57.<http://dx.doi.org/10.1071/WF12202>
- [Filippi et al., 2010] Jean-Baptiste Filippi, Frédéric Morandini, Jacques Henri Balbi, David R. C. Hill (2010), "*Discrete Event Front-tracking Simulation of a Physical Fire-spread Model*", Simulation 86(10): 629-646.
- [Filippi, J.-B., 2003] Filippi, J.-B. 2003, "*Une Architecture Logicielle pour la Multi-Modélisation et la Simulation à évènements Discrets de Systèmes Naturels Complexes*", PhD thesis-Université de Corse.
- [Finney ,1999] Finney, M.A. 1999. Spatial Modeling of Post-Frontal Fire Behavior. Final Report RMRS-99557-RJVA, Systems for Environmental Management, Missoula, MT. 8 pages.
- [Finney, 1995]. Finney, M.A. and K.C. Ryan. 1995. Use of the FARSITE fire growth model for fire prediction in US National Parks. Proc. The International Emergency Mgt. and Engineering Conf. May 1995 Sofia Antipolis, France. pp. 183-189.
- [Finney, 2004] Finney, Mark A. 2004. FARSITE: Fire Area Simulator-model development and evaluation. Res. Pap. RMRS-RP-4, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 47 p
- [Fishwick, 1995] Paul A. Fishwick (1995), "*Simulation model design and execution - building digital worlds*", Prentice Hall 1995, ISBN 978-0-13-098609-2, pp. I-XVI, 1-448
- [Fujioka, 2002] Fujioka FM (2002) A new method for the analysis of fire spread modeling errors. International Journal of Wildland Fire 11(4), 193–203. doi:10.1071/WF02004
- [Hajer A. , 2011] Ayad HAJER, (2011), "Simulation numérique d'un ensemble

REFERENCES

- [Ayad H., 2011] de feux de forêt ", Rapport de stage de M2, INRIA – Paris, France.
- [Jaccard, 1901] Jaccard P (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 547–579.
- [Landis et al., 1977] Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics 33(1), 159–174. doi:10.2307/2529310
- [Nader et al. , 2011] Nader B, Filippi JB, Bisgambiglia P-A. 2011. An experimental frame for the simulation of forest fire spread. In ‘Proceedings of the 2011 Winter Simulation Conference’, 11–14 December 2011, Phoenix, AZ. (Eds S Jain, RR Creasey, J Himmelspach, KP White, M Fu) pp. 1010–1022. (IEEE)
- [Prentice et al., 1993] Prentice, I. C., Sykes, M. T., and Cramer, W.: A simulation model for the transient effects of climate change on forest landscapes, Ecol. Model., 65, 51–70, 1993.
- [Prentice et al., 1992] Prentice, I. C., Cramer, W., Harrison, S. P., Leemans, R., Monserud, R. A., and Solomon, A. M.: A global biome model based on plant physiology and dominance, soil properties and climate, J. Biogeogr., 19, 117–134, 1992.
- [Perry, 1999] Perry GLW, Sparrow AD, Owens IF (1999) A {GIS-supported} model for the simulation of the spatial structure of wildland fire, Cass Basin, New Zealand. Journal of Applied Ecology 36(4), 502–518. doi:10.1046/J.1365-2664.1999.00416.X
- [Rew and Davis, 1990] Rew, R. K., and G. P. Davis. 1990. “The unidata netCDF: Software for scientific data access”. In Sixth International

REFERENCES

- Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Anaheim, California, American Meteorology Society, 33–40.
- [Rothermel, R.C. ,1972] Rothermel, R.C. 1972. A mathematical model for predicting fire spread in wildland fuels. USDA For. Serv. Res. Pap. INT-115.
- [Rothermel et al. 1986] Rothermel R.C., Wilson R.A., Morris G.A., Sackett S.S. 1986. Modeling moisture content of fine dead wildland fuels input into the BEHAVE fire prediction system. USDA Forest Service Research Paper INT-359. Ogden, Utah.
- [Rothermel, R.C. ,1991] Rothermel, R.C. 1991. Predicting behavior and size of crown fires in the northern Rocky Mountains. USDA For. Serv. Res. Pap. INT-438.
- [S. Kanga et al. , 2014] Shruti Kanga, Laxmikant Sharma, Prem Chandra Pandey , Mahendra Singh Nathawat, (2014), "GIS MODELING APPROACH FOR FOREST FIRE RISK ASSESSMENT AND MANAGEMENT", International Journal of Advancement in Remote Sensing, GIS and Geography
- [Sorensen, 1948] Sørensen T (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter* 5, 1–34.
- [Van Wagner, C.E., 1977] C. E. Van Wagner, 1977, Conditions for the start and spread of crown fire, *Canadian Journal of Forest Research*, 7:23-34, 10.1139/x77-004
- [Van Wagner, C.E., 1993] Van Wagner, CE. (1993) Prediction of crown fire behaviour in

REFERENCES

- two stands of jack pine. Canadian Journal of Forest Research
23: 442-9
- [Zeigler et al., 2000] Zeigler, B.; H. Praehofer; T.G. Kim. (2000), "**Theory of Modeling and Simulation**", 2nd Edition. Academic Press, San Diego, CA.
- [Zeigler, 1976] Zeigler, B., (1984), "**Theory of Modelling and Simulation**", Wiley, New York
- Farsite <http://www.firemodels.org/index.php/national-systems/farsite>
- ForeFire <http://forefire.univ-corse.fr/websim2/>
- Lua www.lua.org/
- SERTIT <http://sertit.u-strasbg.fr/>
- WinfNinja <http://www.firemodels.org/index.php/research-systems/windninja>

ANNEXE

1. Formats de fichiers pour la simulation

Cette partie a été réalisée par Hajer AYAD (INRIA) [Ayad H., 2011]. Voici la liste des données qui interfèrent avec le cadre expérimental proposé.

- **Une liste des cas d'observations** : ces cas sont les différents incendies observés. La source de données est la base de données de Prométhée et ONF. Elles sont sous forme de fichiers NetCDF (format ASCII). Les fichiers sont les sorties de l'API - Fire2NetCDF .

Exemple : le chemin du répertoire dans lequel la liste des cas d'observations existe. Les fichiers des feux appartiennent au département de Haute-Corse (2B) et sur toute l'année (2003).

```
fire_case_list = "observation_base/2B01_01_2003/NC/"
```

- **Élévation** : présente l'élévation topographique de la localisation du feu de forêt ; format GIS.

Exemple : le chemin du fichier d'élévation :

```
élévation = "simulation_base/corsica/élévation/alti25.asc"
```

- **Végétation** : décrit les types de végétation existant à l'endroit où le feu a été déclenché au format TIFF.

Exemple : le chemin du fichier de la distribution de végétation :

```
vegetation_cover = "simulation_base/europe/cover/ASTER GDEM/GLC250.tif"
```

- **Données climatiques** : présentent les informations relatives au climat (spécialement le vent) de cette période. Elles sont divisées en données Climat et en données Stations. Une explication détaillée est faite prochainement.

Exemple: le chemin du fichier des données des données climatiques, le composant vent qui se compose d'une direction et d'une vitesse. Si le feu a été déclenché en 2003, les données du climat (wind) seront utilisées à partir d'un fichier NetCDF. S'il a été déclenché après 2003, les données des stations seront utilisées (fichiers csv) :

```
wind = data_directory .. "simulation_base/corsica/weather/ECMWF/output.nc"
```

```
stations = data_directory .. "simulation_base/corsica/weather/stations/stations.index"
```

- **Ortho-photographie** : est un fichier au format graphique Raster et l'extension est tiff. Ce fichier est une image aérienne ou satellite de la surface terrestre rectifiée géométriquement.

Exemple : le chemin du fichier d'ortho-photographie :

```
orthophoto = "simulation_base/corsica/orthophoto/Corse_osm_27574.tif"
```

- **Couverture du territoire** : est en .shp, shapefile, ou "fichier de formes" est un format de fichier issu du monde des Systèmes d'Informations Géographiques (ou SIG - GIS). Il décrit la topographie de l'endroit de la localisation du feu.

Exemple : le chemin du fichier qui décrit la couverture du territoire :

```
land_use_cover = "simulation_base/corsica/cover/IDEA/IFN_Corse_Proterina.shp"
```

Des paramètres de simulation sont aussi définis dans ce fichier de configuration. Ces paramètres sont :

- **Le rapport de forme** : c'est le rapport entre la plus longue extension et la plus petite extension de l'incendie. Il est utilisé pour déterminer le domaine de simulation de la surface brûlée.
- **Les perturbations du vent** : une liste de perturbations du vent (en degrés) si nous avons, entre]-360, 360].

Exemple : l'intervalle des perturbations de la direction du vent (wind) :

```
wind_direction_perturbation = { 0, 10, -10, 20, -20, 30, -30, 40, -40, 50, -50, 60, -60, 70,
-70, 80, -80, 90, -90}
```

- **Paramètres de WindNinja** : Ce sont les paramètres de configuration pour lancer le logiciel WindNinja. Il est responsable du calcul des variations spatiales du vent. Une explication détaillée sera réalisée plus tard.

Exemple de la configuration du logiciel Windninja qui est responsable de la création des fichiers du Vent, les paramètres sont : le chemin du logiciel, le type des mailles, la hauteur du vent, le type et la résolution, la sortie et le chemin du fichier de la végétation :

```
----- WINDNINJA -----
windninja = {
program_path = "... /bin/WindNinja-sandbox-cli/bin/WindNinja.exe",
wind_height = 10.,
mesh_choice = "coarse",
output_type = "fb",
output_resolution = 14,
Put either: "file" (data extracted from file "vegetation_cover"),
vegetation = "file"
}
```

- **Paramètres de ForeFire** : Ces sont les paramètres de configuration pour lancer ForeFire. Le simulateur qui est présenté et simule les propagations des feux de forêt.

Exemple de la commande du ForeFire qui exécute la simulation du feu. Cette commande se compose de tous les paramètres du ForeFire. Comme :

- La condition d'arrêt de la simulation (stop_criterion) qui est utilisée ici est la surface total brûlée du feu - observed_burned_area"- (en hectares). Cette condition peut être aussi la durée du feu - observed_duration – (en minutes).
- Le type de commande (forefire).
- Le fichier ASCII (.xml) qui décrit les combustibles.
- La résolution de la grille de la simulation (10 mètres par grille).
- Les noms des fichiers de sortie (result). Les sorties sont deux fichiers : un fichier NetCDF qui est contient toutes les données obtenues à la fin de la simulation d'un feu quelconque en ForeFire (nc) et un fichier KML qui est une présentation graphique de la simulation compatible avec Google Earth pour visualiser la sortie du ForeFire (kml).

----- FOREFIRE -----

Stop criterion: "observed_burned_area" or "observed_duration".

stop_criterion = "observed_burned_area"

forefire = {

Put the command to launch ForeFire.

command = "forefire",

fuel_description = "configuration/fuelsModelsNStat.xml",

stationary = true,

quant = 1,

Extension of the initial front.

initial_extension = 25.,

With background image?

with_background = false,

Grid resolution in meters (integer).

output_grid_resolution = 10.,

Put a file name (not a full path) without extension. This file will be written in the output directory for the fire case.

output_file_name = "result",

output_format = {"kml", "nc"}

}

La Figure 3 présente une capture d'écran qui montre les fichiers qui sont créés pour ForeFire afin de lancer une simulation en détaillant les paramètres du fichier de configuration (SA) et un extrait du fichier du modèle de combustible utilisé.

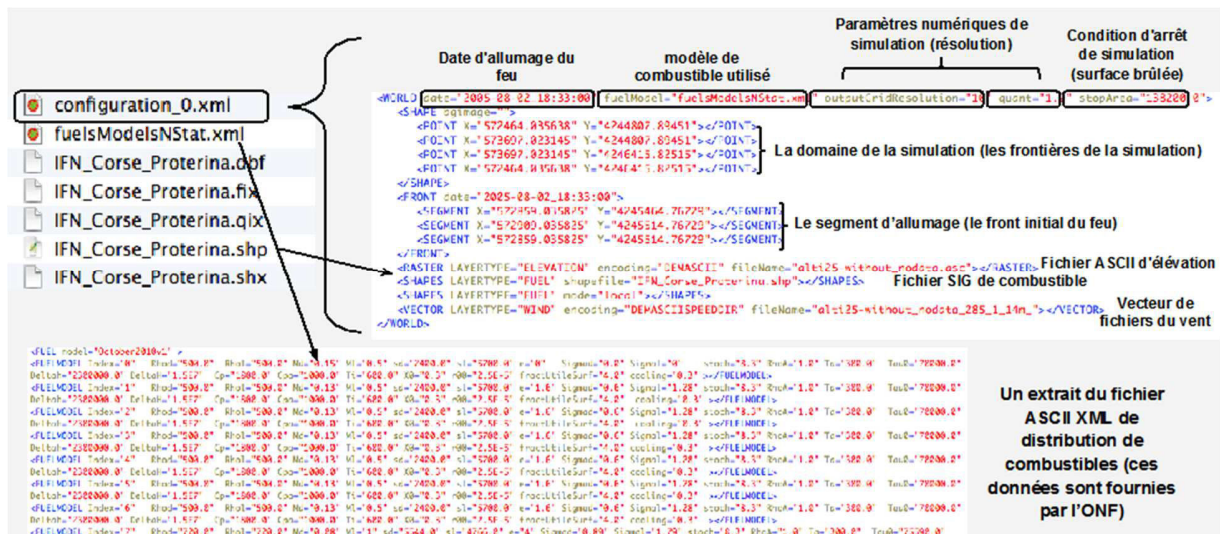


Figure 3. Capture d'écran des fichiers qui sont créés pour ForeFire