

Université de Montréal

Simulation numérique de feux de forêt avec
réinitialisation et contournement d'obstacles

par

Alexandre Desfossés Foucault

Département de mathématiques et de statistique

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en mathématiques

Orientation mathématiques appliquées

février 2010

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé

**Simulation numérique de feux de forêt avec
réinitialisation et contournement d'obstacles**

présenté par

Alexandre Desfossés Foucault

a été évalué par un jury composé des personnes suivantes :

Jacques Bélair

(président-rapporteur)

Anne Bourlioux

(directeur de recherche)

Paul Arminjon

(membre du jury)

Mémoire accepté le:

25 janvier 2010

SOMMAIRE

Ce travail présente une technique de simulation de feux de forêt qui utilise la méthode *Level-Set*. On utilise une équation aux dérivées partielles pour déformer une surface sur laquelle est imbriqué notre front de flamme. Les bases mathématiques de la méthode *Level-set* sont présentées. On explique ensuite une méthode de réinitialisation permettant de traiter de manière robuste des données réelles et de diminuer le temps de calcul. On étudie ensuite l'effet de la présence d'obstacles dans le domaine de propagation du feu. Finalement, la question de la recherche du point d'ignition d'un incendie est abordée.

Mots-clés : Méthode *Level-Set*, simulation, feux de forêt, équations aux dérivées partielles.

SUMMARY

This work presents a forest fire simulation model which uses the *Level-Set* method. We use a partial differential equation to deform a surface on which our flame front is inscribed. The mathematical foundations of the *Level-set* method are presented. We then explain a reinitialization method that allows us to treat in a robust way real data and to reduce the calculation time. The effect of the presence of barriers in the fire propagation domain is also studied. Finally, we make an attempt to find the ignition point of a forest fire.

Keywords : Level-set method, simulation, forest fire, partial differential equations.

TABLE DES MATIÈRES

Sommaire	iii
Summary	iv
Liste des figures	vii
Liste des tableaux	x
Liste des sigles et abbréviations	xi
Remerciements	1
Introduction	2
Chapitre 1. Méthode <i>Level-Set</i> et simulation de feux de forêt ...	4
1.1. Méthode <i>Level-Set</i>	4
1.2. Application aux feux de forêt	11
Chapitre 2. Réinitialisation et méthode narrow-band	16
2.1. Réinitialisation avec EDP	19
2.1.1. Algorithme	20
2.1.2. Illustration de la réinitialisation.....	23
2.1.3. Comparaison des méthodes de réinitialisation possibles.....	25
2.1.4. Retour sur l'exemple problématique du début du chapitre	29
2.2. Méthode narrow-band.....	32
Chapitre 3. Contournement d'obstacles	37
3.1. Modélisation des obstacles	37

3.2.	Obstacles entourés	43
3.3.	Obstacles, réinitialisation et vitesses de propagation variables - Un exemple avec données “réelles”	45
Chapitre 4.	Recherche du point d’ignition d’un feu	48
4.1.	Inversion du temps dans l’EDP	49
4.1.1.	Présentation de l’algorithme	49
4.1.2.	Un exemple de résultat pour le cas où le processus de propagation de feu est réversible.....	50
4.1.3.	Exemple 2 - Un front en forme de marguerite	53
4.1.4.	Exemple 3 - Un front asymétrique.....	56
4.1.5.	Exemple 4 - Un front se brisant en deux en remontant le temps	58
4.1.6.	En présence d’obstacle	60
4.1.7.	Inversion du temps dans l’EDP - Conclusion	60
Chapitre 5.	Conclusion	62
Bibliographie		63

LISTE DES FIGURES

1.1	Illustration de la méthode des marqueurs	4
1.2	Inconvénient de la méthode des marqueurs	5
1.3	Front circulaire imbriqué dans un cône	6
1.4	Discrétisation des dérivées spatiales	9
1.5	Illustration des conditions au bord	11
1.6	Illustration du principe d'Huygens pour le cas sans vent	12
1.7	Illustration du principe d'Huygens pour le cas avec vent	14
2.1	Front de flamme au temps $t = 0$ accompagné des courbes de niveau de la fonction <i>Level-Set</i> au même temps	17
2.2	Front de flamme accompagné des courbes de niveau de la fonction <i>Level-Set</i> correspondante après 10 itérations pour le cas des vitesses aléatoires	18
2.3	Front de flamme accompagné des courbes de niveau de la fonction <i>Level-Set</i> correspondante après 20 itérations pour le cas des vitesses aléatoires	18
2.4	Fonction <i>signe</i> (x) pour $h = 0.5$ et x entre -5 et 5	21
2.5	Fonction <i>Level-Set</i> à des pseudo-temps différents lors du processus de réinitialisation	24
2.6	Solutions de l'équation (2.1.11) obtenues des 3 manières différentes au temps $t = 8.33$	27
2.7	Solution au cas problématique en utilisant la réinitialisation au temps $t = 0.5$	29

2.8	Solution au cas problématique en utilisant la réinitialisation au temps $t = 0.5$ pour le maillage raffiné	30
2.9	Points à l'intérieur des bandes	33
2.10	Fonction <i>Level-Set</i> correspondante	33
2.11	Comparaison des solutions obtenues des deux manières différentes....	34
3.1	Premier exemple de contournement d'obstacle.....	39
3.2	Premier exemple de contournement d'obstacle.....	40
3.3	Exemple de contournement d'obstacle en présence de vent.....	41
3.4	Exemple de contournement d'obstacle en présence de vent.....	42
3.5	Exemple de contournement d'obstacle en présence de vent.....	43
3.6	Régions brûlées à chaque 100 itérations	46
3.7	Régions brûlées à deux temps différents.....	47
4.1	Collision de deux fronts et perte d'information	49
4.2	Fronts de flamme obtenus vers le futur et vers le passé à des instants donnés.....	51
4.3	Front correspondant obtenu en 300 itérations	53
4.4	Résultat de l'inversion de l'EDP pour ces vitesses de propagation	54
4.5	Résultat de l'inversion de l'EDP pour ces vitesses de propagation	54
4.6	Erreur commise en inversant l'EDP en fonction du nombre d'itérations utilisées pour simuler la propagation du feu	55
4.7	Front correspondant obtenu en 300 itérations	56
4.8	Résultat de l'inversion de l'EDP pour ces vitesses de propagation	57
4.9	Résultat de l'inversion de l'EDP pour ces vitesses de propagation	57
4.10	Erreur commise en inversant l'EDP en fonction du nombre d'itérations utilisées pour simuler la propagation du feu vers le futur	58

4.11	Front correspondant obtenu en 300 itérations	59
4.12	Résultat de l'inversion de l'EDP pour ces vitesses de propagation	59
4.13	Fronts avec obstacle obtenus vers le futur	60
4.14	Fronts avec obstacle obtenus en remontant le temps.....	60

LISTE DES TABLEAUX

2.1	Paramètres utilisés lors du premier test effectué avec des vitesses variables.....	17
2.2	Paramètres utilisés lors du test de validation de la réinitialisation	23
2.3	Paramètres utilisés lors de la comparaison des méthodes de réinitialisation 26	
2.4	Classement des méthodes de réinitialisation en fonction de la précision du résultat.....	28
2.5	Classement des méthodes de réinitialisation en fonction du temps de calcul.....	29
2.6	Erreur au temps $t = 0.5$ entre la solution obtenue avec différentes valeurs de h et la solution de référence.....	31
2.7	Paramètres utilisés pour le tests de temps de calcul.....	35
2.8	Résultats des tests de temps de calcul pour plusieurs tailles de domaine différentes.....	35
3.1	Paramètres utilisés pour les tests de contournement d'obstacle.....	41
4.1	Paramètres utilisés lors de cette expérience.....	51

LISTE DES SIGLES ET ABBRÉVIATIONS

BROS : taux de propagation vers l'arrière(de l'anglais back rate of spread)

EDP : Équation aux dérivées partielles

FROS : taux de propagation latéral(de l'anglais flank rate of spread)

ROS : taux de propagation (de l'anglais rate of spread)

REMERCIEMENTS

Je voudrais remercier Anne Bourlioux pour avoir guidé mes recherches. Merci aussi à Jessica Hovnanian, qui a contribué à une partie de ce travail lors d'un stage d'été, et à tous ceux qui m'ont supporté.

INTRODUCTION

Partout à travers le monde, les feux de forêt coûtent des millions de dollars en dommages à chaque année, en plus de causer la mort de plusieurs personnes. Une grave décision à prendre pour le gouvernement est de décider quand évacuer une communauté qui se trouve près d'un feu de forêt, ou même s'il faut vraiment l'évacuer.

Afin de faciliter cette décision, plusieurs techniques de simulation de feux de forêt ont été élaborées. Certains tentent d'utiliser des automates cellulaires [7]. Ces automates sont construits en divisant une forêt en cellules, en général carrées, puis en choisissant des règles pour la propagation du feu d'une cellule à l'autre en se basant sur des principes physiques. D'autres ont essayé de représenter le phénomène en établissant des équations de réaction-diffusion [8] en appliquant les principes de conservation de masse et d'énergie.

L'approche considérée dans ce mémoire est basée sur la méthode *Level-Set*. Cette méthode repose sur l'idée qu'un front de flamme peut être représenté par la courbe de niveau d'une surface et qu'il est possible d'obtenir une équation aux dérivées partielles pour la déformer, simulant ainsi la propagation du feu. Pour un choix possible d'équation, on réfère le lecteur à [9]. L'équation étudiée ici sera basée sur un modèle physique différent.

Le modèle physique utilisé dans ce travail est celui du projet *Prometheus*. Le gouvernement albertain finance ce projet, qui consiste en un logiciel de simulation de feux de forêt. Ce programme utilise une méthode de marqueurs pour suivre un front de flamme. Ceci revient à placer des marqueurs sur notre front de flamme, déplacer chacun de ces marqueurs en fonction de la vitesse de propagation du front au point où ils sont, puis les reconnecter pour obtenir la position du front

à un temps futur. Cependant, cette manière de procéder est source de quelques difficultés techniques. Le but de ce travail est de prendre le même modèle physique que celui utilisé par le projet *Prometheus* pour les vitesses de propagation, mais d'utiliser la méthode *Level-Set* au lieu de la méthode de marqueurs pour suivre le front.

Ce mémoire poursuit le travail déjà accompli [3] [4] sur la simulation de feux de forêt avec la méthode *Level-Set*. Mon objectif premier était d'améliorer la méthode existante dans le but de pouvoir considérer des données réelles. Le présent mémoire décrit en quatre chapitres le cheminement que j'ai suivi pour répondre à cet objectif. Le premier chapitre explique les bases de la méthode *Level-Set*, et la façon de l'appliquer pour obtenir une équation aux dérivées partielles décrivant la propagation d'un feu.

Le second montre comment rajouter à l'algorithme de simulation de feux de forêt un processus appelé *réinitialisation*, qui permet de traiter des cas réels où les vitesses de propagation sont très hétérogènes. En effet, la principale faiblesse du travail effectué précédemment est que les cas où la vitesse de propagation du feu est considérée comme variable n'ont pas été étudiés en détail. Ensuite, on explique comment utiliser ce processus pour réduire significativement le temps de calcul de la méthode *Level-Set*.

Le troisième chapitre rajoute au modèle la notion de points infranchissables, qui peuvent représenter par exemple des lacs, des falaises et d'autres obstacles naturels. Ce chapitre se conclut en montrant comment appliquer les améliorations du modèle des chapitres 2 et 3 à un cas assez proche de la réalité.

Le dernier sujet abordé est la recherche du point d'ignition d'un feu. Au chapitre 4, on tente de répondre à la question : étant donné un front de flamme quelconque, est-il possible de retrouver mathématiquement sa source ?

Chapitre 1

MÉTHODE *LEVEL-SET* ET SIMULATION DE FEUX DE FORÊT

Ce chapitre présente les outils mathématiques utilisés dans ce mémoire pour simuler la propagation d'un feu de forêt. Les notions de bases sur la méthode *Level-Set*, les fonctions de distance signée et les schémas amont (upwind) pour la discrétisation des équations aux dérivées partielles sont présentés.

1.1. MÉTHODE *Level-Set*

Un problème de propagation de feu peut être ramené à un problème de suivi d'interface. Il existe plusieurs méthodes traditionnelles pour résoudre ce genre de problème. Par exemple, le modèle dans *Prometheus* utilise une méthode de marqueurs. Cette technique consiste à placer plusieurs marqueurs sur l'interface qui nous intéresse, puis à les faire évoluer dans le temps. Prenons l'exemple du feu de forêt. La figure 1.1 montre un cercle représentant un front de flamme, ainsi que quelques marqueurs choisis sur ce front.

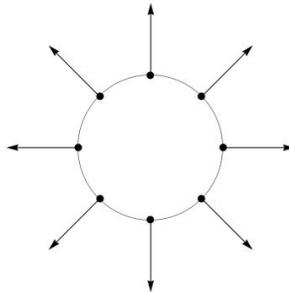


FIG. 1.1. Illustration de la méthode des marqueurs

Si on connaît les vitesses de propagation du feu en tout point de la région qui nous intéresse, on peut déplacer ces marqueurs pour un temps donné en fonction de la vitesse au point où ils sont. Ensuite, on peut les reconnecter pour avoir la position du front. Cependant, cette méthode a plusieurs inconvénients :

- (1) Il peut être difficile de reconnecter les marqueurs si leurs trajectoires se croisent ou si le front change de topologie.
- (2) Des collisions entre marqueurs sont possibles.

Pour se convaincre des inconvénients de cette méthode, on peut considérer le cas où deux fronts de flamme se rencontrent, tel qu'illustré à la figure 1.2.

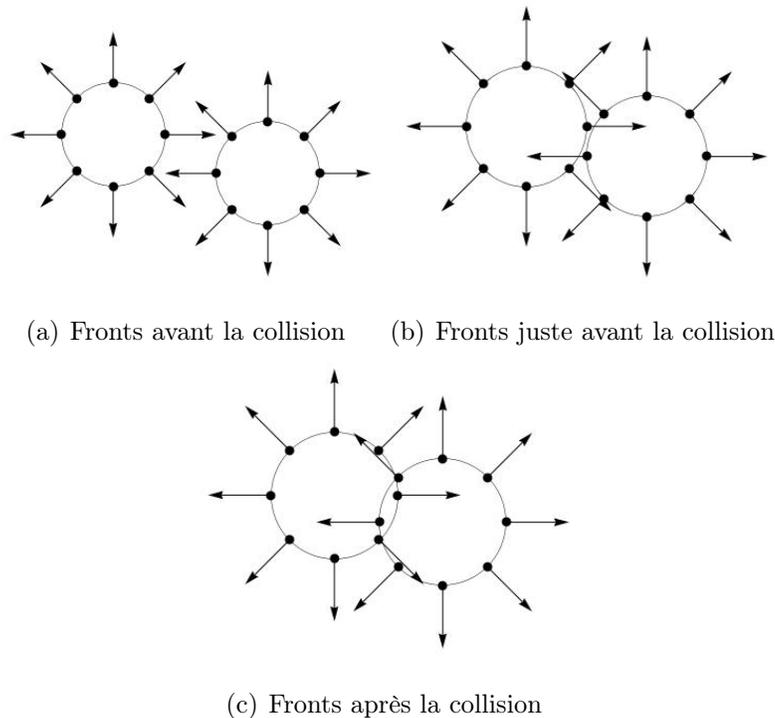
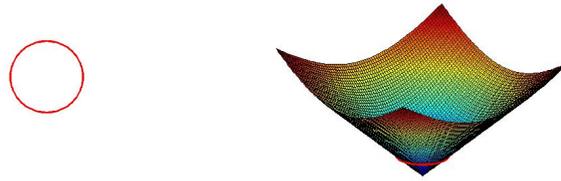


FIG. 1.2. Inconvénient de la méthode des marqueurs

Lors de la rencontre des deux fronts, notre interface change de topologie et il peut y avoir collision de marqueurs. Il sera difficile de décider où est vraiment le front de flamme. Aussi, comme on voit bien dans cet exemple, il faut éliminer les marqueurs qui sont maintenant à l'intérieur du front et il peut être difficile d'obtenir un critère pour les identifier.

Pour résoudre ces problèmes, la méthode Level-Set a été inventée par Sethian et Osher [1]. L'idée principale derrière cette méthode est qu'on peut représenter implicitement une interface comme une courbe de niveau d'un objet de dimension plus élevée. Par exemple, considérons un front de flamme circulaire satisfaisant l'équation $x^2 + y^2 = 1$. La figure 1.3 montre qu'on peut voir ce front comme la courbe de niveau zéro d'un cône d'équation $z = \sqrt{x^2 + y^2} - 1$.



(a) Front de flamme circulaire (b) Surface ayant le front en (a) comme courbe de niveau

FIG. 1.3. Front circulaire imbriqué dans un cône

La prochaine étape est de trouver une équation qui décrit la déformation de cette surface en fonction du temps, de telle sorte que son niveau zéro se déforme selon le modèle de propagation. Une approche pour dériver cette équation est celle de Sethian dans [2] :

Soit ϕ une fonction ayant notre front de flamme comme courbe de niveau 0. Considérons une particule sur le front avec trajectoire $\vec{x}(t)$. On exige qu'une telle particule demeure sur le niveau zéro de la fonction ϕ en tout temps :

$$\phi(\vec{x}(t), t) = 0 \quad (1.1.1)$$

En dérivant par rapport à t et en appliquant la règle de dérivation en chaîne, on obtient

$$\phi_t + \vec{x}'(t) \cdot \nabla \phi = 0 \quad (1.1.2)$$

Pour un t fixé, $\vec{x}'(t)$ nous donne donc la vitesse de propagation en un point du niveau zéro. On peut répéter l'argument pour les autres niveaux pour obtenir

l'équation suivante :

$$\phi_t + \vec{v} \cdot \nabla \phi = 0 \quad (1.1.3)$$

Ainsi, le niveau zéro de ϕ (le front de flamme) se déplacera selon les vitesses de propagation \vec{v} prédites par notre modèle physique et les autres courbes de niveaux de ϕ seront déformées en utilisant la vitesse de propagation d'un feu fictif en tout point (x, y) (le modèle utilisé pour déterminer les vitesses de propagation n'exige pas que le feu soit présent en un point pour pouvoir y calculer une vitesse).

On résoud donc l'équation aux dérivées partielles (1.1.3). Il reste à choisir une condition initiale et des conditions aux bords. Notre condition initiale doit être la représentation implicite d'un front de flamme donné. Il y a plusieurs choix de condition initiale puisqu'il existe une infinité de représentations implicites pour une courbe. Pour plusieurs raisons données dans [2], le choix traditionnel est de prendre ϕ comme la fonction de distance signée à notre front. Cette fonction associe à chaque point du domaine la distance entre ce point et le front de flamme, puis multiplie cette distance par -1 si le point est brûlé. Par exemple, pour un front circulaire centré en 0 de rayon 1, la fonction de distance signée est $\phi(x, y) = \sqrt{x^2 + y^2} - 1$.

Ce choix encode beaucoup d'informations utiles dans la fonction ϕ . Par exemple, on peut facilement trouver les points où le feu est déjà passé puisqu'en ces points, la fonction ϕ sera négative. Un autre avantage est qu'une fonction de distance signée est assez régulière, ce qui sera utile pour discrétiser notre équation.

Cependant, en résolvant l'équation (1.1.3) la fonction ϕ ne restera pas une fonction de distance signée. On présente au chapitre 2 un processus nous permettant de ramener ϕ à une telle fonction.

Pour choisir les conditions aux bords, remarquons qu'en réalité les domaines étudiés sont énormément grands et qu'on ne devrait pas avoir à considérer l'effet des bords sur les feux simulés. Pour faire comme si notre domaine s'étendait à l'infini, on peut faire une extrapolation linéaire pour rajouter des points tout autour du domaine étudié. On s'assurera aussi d'avoir un domaine assez grand pour que notre front initial ne grandisse pas jusqu'à sa frontière. C'est ce choix de conditions aux bords qui sera utilisé dans tous les tests subséquents.

Une fois nos conditions aux bords et notre condition initiale choisie, notre problème est bien posé. Sous certaines conditions [5], il y a existence et unicité de la solution. Cependant, dans la réalité les champs de vitesses de propagation sont trop compliqués pour qu'on puisse résoudre l'équation (1.1.3) analytiquement. On peut toutefois résoudre numériquement cette équation. La résolution numérique d'équations de ce type a été étudiée en détail (par exemple voir [5]). On présente ici des arguments intuitifs pour justifier la discrétisation, pour un cas qui reviendra souvent dans les chapitres ultérieurs. Supposons que la vitesse de propagation du feu est normale au front. C'est-à-dire, la vitesse est de la forme :

$$\vec{v} = F \frac{\nabla \phi}{\|\nabla \phi\|} \quad (1.1.4)$$

où F est une constante positive. Si on remplace ceci dans l'équation 1.1.3, on obtient :

$$\phi_t + F \|\nabla \phi\| = 0 \quad (1.1.5)$$

On veut ensuite résoudre numériquement cette équation. Quadrillons tout d'abord notre domaine pour avoir un maillage uniforme de $(N \times M)$ points avec un pas en espace égal à h dans les deux directions. On peut ensuite discrétiser la dérivée temporelle en utilisant un schéma Euler-avant. L'approximation numérique de la solution au i -ième point du quadrillage en x et au j -ième point en y à l'itération $n + 1$ sera donnée par :

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + F \Delta t \sqrt{\phi_{x|i,j}^2|_n + \phi_{y|i,j}^2|_n} \quad (1.1.6)$$

Pour discrétiser les dérivées spatiales, on utilise la méthode de Godunov [5]. Le choix de l'approximation des dérivées doit dépendre du sens de propagation de l'information tel qu'illustré à la figure 1.4.

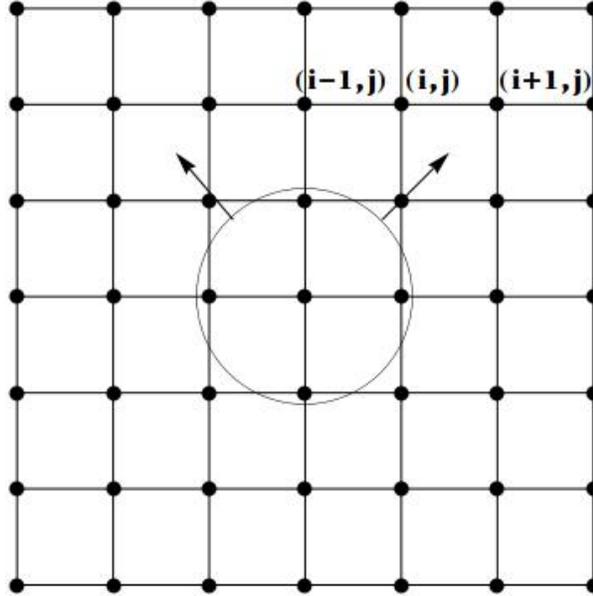


FIG. 1.4. Discrétisation des dérivées spatiales

On souhaite approximer les dérivées spatiales au point (i, j) à l'aide des points voisins sachant que l'information doit se propager dans la direction du gradient de ϕ , tel qu'illustré par les flèches. C'est la méthode de Godunov qui permet de choisir adéquatement l'approximation des dérivées spatiales pour refléter ce fait. Elle nous dit que pour une équation de type :

$$\phi_t + H(\nabla\phi) = 0 \quad (1.1.7)$$

on peut écrire la version discrète de $H(\nabla\phi)$ comme :

$$\hat{H} = ext_x ext_y H(\phi_x, \phi_y) \quad (1.1.8)$$

Pour définir ext_x et ext_y , on doit d'abord définir la différence avant ϕ_x^+ et la différence arrière ϕ_x^- en (i, j) à l'itération n comme :

$$\phi_x^+|_{i,j}^n = \frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{h} \quad (1.1.9)$$

$$\phi_x^-|_{i,j}^n = \frac{\phi_{i,j}^n - \phi_{i-1,j}^n}{h} \quad (1.1.10)$$

Définissons aussi pour chaque point (i, j) les intervalles suivants :

$$I_x = [\min(\phi_x^-, \phi_x^+), \max(\phi_x^-, \phi_x^+)] \quad (1.1.11)$$

$$I_y = [\min(\phi_y^-, \phi_y^+), \max(\phi_y^-, \phi_y^+)] \quad (1.1.12)$$

où les approximations des dérivées sont calculées en (i, j) . Avec ces définitions, on a que si $\phi_x^- < \phi_x^+$, $ext_x H$ est la valeur minimale de H pour ϕ_x dans I_x . Si $\phi_x^- > \phi_x^+$, $ext_x H$ est la valeur maximale de H pour ϕ_x dans I_x . Si les deux dérivées sont égales, $ext_x H$ est la valeur obtenue en évaluant H en $\phi_x = \phi_x^- = \phi_x^+$. En appliquant ce schéma à l'équation (1.1.5), on obtient la formule suivante, due à Rouy et Tourin [10] :

$$\phi_x^2 = \max \left(\max (\phi_x^-, 0)^2, \min (\phi_x^+, 0)^2 \right) \quad (1.1.13)$$

On peut répéter l'argument pour obtenir une formule similaire pour la dérivée en y . Un autre point à mentionner sur la discrétisation de l'équation est le choix du pas de temps. Il faut le choisir pour respecter une condition *CFL* (Courant-Friedrichs-Levy) pour que notre schéma numérique soit stable. Dans ce cas, la vitesse de propagation de l'information est de F et on doit prendre

$$\Delta t < \frac{h}{F} \quad (1.1.14)$$

Pour plus de sécurité, on divisera ce nombre par deux dans tous les tests présentés subséquentment.

Écrivons maintenant explicitement les conditions aux bords choisies plus haut. On souhaite faire une extrapolation linéaire pour rajouter des points tout autour de notre domaine. Considérons les points $\phi_{1,j}$ sur la première colonne du domaine. On rajoute une rangée de points à gauche $\phi_{0,j}$ avec la formule suivante :

$$\phi_{0,j} = 2 \cdot \phi_{1,j} - \phi_{2,j} \quad (1.1.15)$$

Supposons que notre domaine est de taille $(N \times M)$. On rajoute similairement les rangées de points en haut $\phi_{i,0}$, à droite $\phi_{N+1,j}$ et en bas $\phi_{i,M+1}$ avec les formules :

$$\phi_{i,0} = 2 \cdot \phi_{i,1} - \phi_{i,2} \quad (1.1.16)$$

$$\phi_{N+1,j} = 2 \cdot \phi_{N,j} - \phi_{N-1,j} \quad (1.1.17)$$

$$\phi_{i,M+1} = 2 \cdot \phi_{i,M} - \phi_{i,M-1} \quad (1.1.18)$$

La figure 1.5 illustre la situation. Sur cette figure, les points noirs représentent les points à l'intérieur de notre domaine et les points blancs ceux qui sont rajoutés.

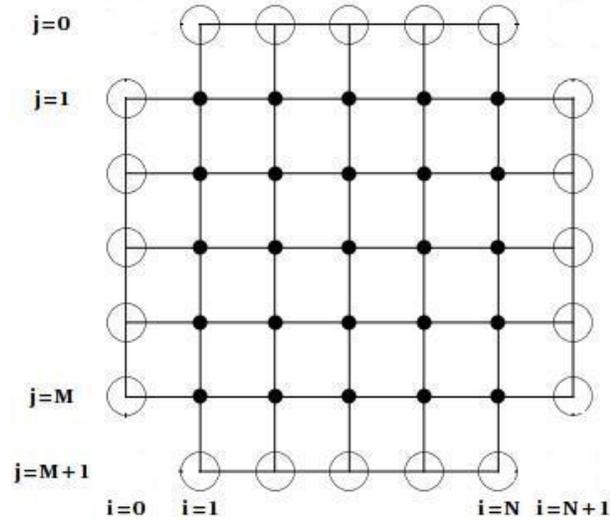


FIG. 1.5. Illustration des conditions au bord

1.2. APPLICATION AUX FEUX DE FORÊT

Dans la section précédente, on a établi que l'évolution d'un front de flamme peut être décrite en résolvant l'équation suivante :

$$\phi_t + \vec{v} \cdot \nabla \phi = 0 \quad (1.2.1)$$

où \vec{v} est un champ vectoriel défini à chaque point du plan représentant les vitesses de propagation de feu et où ϕ est une surface ayant le front de flamme comme courbe de niveau. Il est possible de partir de principes physiques pour déterminer ce champ vectoriel ([3], [4]).

L'idée générale est d'utiliser le principe d'Huygens pour la propagation des ondes. Pour commencer, considérons le cas sans vent. Prenons un front de flamme quelconque. Pour voir comment il se propagera, on suppose que chaque point du front est un nouveau point d'ignition d'un feu et que ces nouveaux feux croissent comme des cercles, comme le montre la figure 1.6.

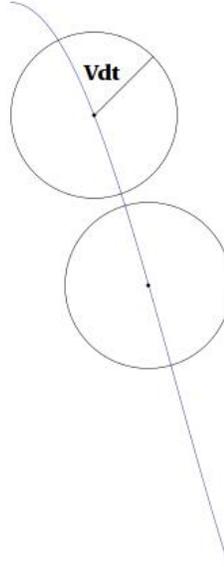


FIG. 1.6. Illustration du principe d'Huygens pour le cas sans vent

La valeur du rayon Vdt du cercle dépend de paramètres physiques à ce point (comme par exemple, la végétation et la température) et de la durée de la combustion dt . Pour trouver \vec{v} , on exige que le vecteur normal au front à un temps $t_0 + dt$ soit le même que le vecteur normal au front en t_0 . Donc, on cherche le point sur le cercle satisfaisant cette propriété. Le vecteur normal au front en t_0 est donné par $\vec{n} = \frac{\nabla\phi}{\|\nabla\phi\|}$. Pour trouver le vecteur normal au temps $t_0 + dt$, considérons une paramétrisation du cercle de la figure 1.6 :

$$x(s) = Vdt \cos(s) \quad (1.2.2)$$

$$y(s) = Vdt \sin(s) \quad (1.2.3)$$

Le vecteur normal unitaire de cette courbe est $\vec{w} = (\cos(s), \sin(s))$. On cherche donc la valeur du paramètre s rendant ces deux vecteurs normaux identiques. Ceci nous donnera la nouvelle position du centre du cercle de la figure 1.6 après un temps dt et donc nous pourrons écrire la vitesse de propagation \vec{v} . On peut trouver s en comparant les carrés des quotients des composantes x et y des deux expressions pour les vecteurs normaux :

$$\frac{\phi_x^2}{\phi_y^2} = \frac{\cos^2(s)}{\sin^2(s)} \quad (1.2.4)$$

Après quelques manipulations, on obtient :

$$\sin^2(s) = \frac{\phi_y^2}{\|\nabla\phi\|^2} \quad (1.2.5)$$

La nouvelle position du centre du cercle est donc :

$$x = V dt \frac{\phi_x}{\|\nabla\phi\|} \quad (1.2.6)$$

$$y = V dt \frac{\phi_y}{\|\nabla\phi\|} \quad (1.2.7)$$

Ceci correspond au déplacement du centre du cercle en un temps dt . Pour avoir la vitesse cherchée, on divise ce déplacement par dt . Finalement, on arrive à l'équation suivante pour les vitesses de propagation en (x, y) :

$$\vec{v} = \frac{V\nabla\phi}{\|\nabla\phi\|} \quad (1.2.8)$$

Ceci nous permet de réécrire l'équation d'évolution du feu comme :

$$\phi_t + V\|\nabla\phi\| = 0 \quad (1.2.9)$$

Il est aussi possible d'obtenir une équation tenant compte du vent. On suppose alors que pour simuler la propagation d'un front de flamme quelconque, chaque point du front est un nouveau point d'allumage d'un feu et que ces feux croissent comme des ellipses avec leurs grands axes orientés dans la direction du vent. Ceci changera la paramétrisation pour du nouveau front et donc l'expression pour la vitesse de propagation ne sera pas la même. La figure 1.7 illustre le phénomène avec un vent qui souffle en direction $\theta = \pi/4$.

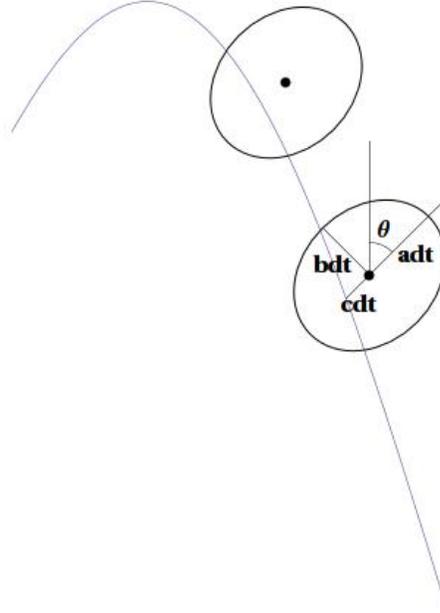


FIG. 1.7. Illustration du principe d'Huygens pour le cas avec vent

Définissons le taux de propagation (rate of spread) dans la direction du vent comme $ROS = a + c$, le taux de propagation dans la direction perpendiculaire au vent (flank rate of spread) comme $FROS = b$ et le taux de propagation contre le vent (back rate of spread) comme $BROS = a - c$. Les valeurs a et b serviront à définir les axes de l'ellipse et la valeur c donnera le déplacement de l'ellipse dû au vent.

Définissons aussi la matrice de rotation :

$$R_{\theta} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (1.2.10)$$

Définissons ensuite la matrice suivante :

$$\tilde{A} = \begin{pmatrix} b & 0 \\ 0 & a \end{pmatrix} \cdot R_{\theta} \quad (1.2.11)$$

Avec ces définitions, la vitesse d'advection correspondant à la propagation du feu est :

$$\vec{v} = \frac{\tilde{A}^t \tilde{A} \nabla \phi}{\|\tilde{A} \nabla \phi\|} + R_{\theta} \begin{pmatrix} 0 \\ c \end{pmatrix} \quad (1.2.12)$$

L'équation représentant la propagation du feu en présence de vent peut ensuite s'écrire sous la forme [4] :

$$\phi_t + \|\tilde{A}\nabla\phi\| + \vec{C} \cdot \nabla\phi = 0 \quad (1.2.13)$$

où $\vec{C} = R_\theta \begin{pmatrix} 0 \\ c \end{pmatrix}$. Le schéma numérique pour résoudre cette équation est expliqué dans [4].

Dans les travaux précédents [4], les paramètres a et b étaient dans la plupart des tests considérés constants partout dans le domaine. Dans ce travail, on a étudié comment améliorer le modèle pour prendre en compte des paramètres qui varient spatialement. Le prochain chapitre présente quelques outils qui permettent de prendre en compte l'hétérogénéité des paramètres de notre modèle et de réduire le temps de calcul de la méthode *Level-Set*.

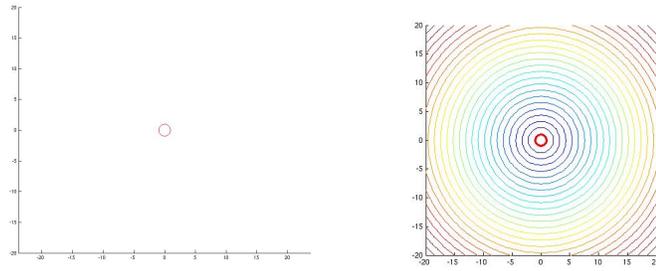
Chapitre 2

RÉINITIALISATION ET MÉTHODE NARROW-BAND

Une étape importante à franchir pour rendre le modèle de propagation de feu plus réaliste est d'utiliser des vitesses de propagation qui varient d'un point à l'autre du domaine. Cependant, avoir des vitesses variables peut nuire à l'exactitude de la solution numérique, surtout lorsque ces vitesses ne sont pas continues. Si c'est le cas, ou si les vitesses oscillent rapidement, la surface sur laquelle est inscrit notre front de flamme deviendra assez rapidement irrégulière, et ce même si le front demeure lisse. Cette irrégularité dans une direction transverse au front peut causer des problèmes lors de l'utilisation des différences finies. Ceci nous amène à parler d'un sujet important lié à la méthode *Level-Set* : la *réinitialisation*. C'est un processus qui nous assure que la surface reliée à notre front de flamme demeure régulière au fil des itérations. On rappelle qu'on résoud numériquement l'équation aux dérivées partielles suivante :

$$\phi_t + \|A\nabla\phi\| + \vec{C}(x, y) \cdot \nabla\phi = 0 \quad (2.0.14)$$

où notre front de flamme $\Gamma(x, y)$ est la courbe de niveau 0 de la fonction ϕ et où la valeur initiale de ϕ est la fonction de distance signée à ce front. Voici un exemple de résultats avec des paramètres $a(x, y)$ et $b(x, y)$ qui varient spatialement et un front initial en forme de cercle. Le front initial (en rouge) et ses courbes de niveaux sont montrés à la figure 2.1.



(a) Front circulaire en $t = 0$ (b) Courbes de niveau de la fonction ϕ en $t = 0$

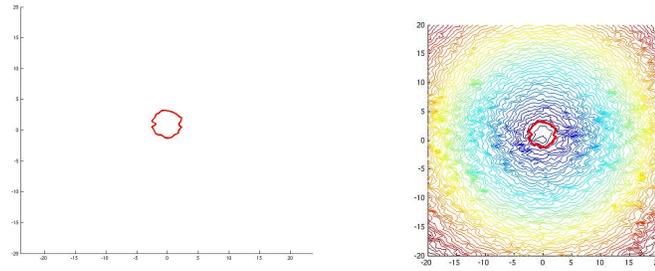
FIG. 2.1. Front de flamme au temps $t = 0$ accompagné des courbes de niveau de la fonction *Level-Set* au même temps

Prenons aléatoirement avec une distribution uniforme les valeurs de $a(x, y)$ entre 2 et 7 et $b(x, y)$ entre 2 et 10. Posons $c(x, y) = 4$. Les autres paramètres utilisés sont donnés au tableau 2.1.

Paramètres	Valeurs
h	0.5
Δt	$0.5 \cdot \frac{\Delta x}{\max(a,b,c)}$
Taille du domaine en x	$[-20, 20]$
Taille du domaine en y	$[-20, 20]$

TAB. 2.1. Paramètres utilisés lors du premier test effectué avec des vitesses variables

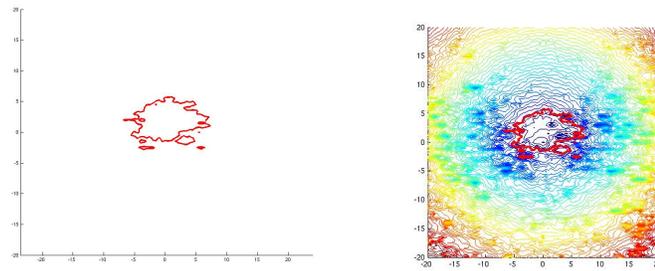
La figure 2.2 présente le résultat après 10 itérations.



(a) Front en $t = 0.25$ (b) Courbes de niveau de la fonction ϕ en $t = 0.25$

FIG. 2.2. Front de flamme accompagné des courbes de niveau de la fonction *Level-Set* correspondante après 10 itérations pour le cas des vitesses aléatoires.

On voit que la fonction *Level-Set* est maintenant très irrégulière. Si on continue l'expérience, on observe que des contours de niveau zéro apparaissent en plein milieu du domaine (voir figure 2.3), ce qui ne peut pas être la véritable solution au problème.



(a) Front au temps $t = 0.5$ (b) Courbes de niveau de la fonction ϕ en $t = 0.5$

FIG. 2.3. Front de flamme accompagné des courbes de niveau de la fonction *Level-Set* correspondante après 20 itérations pour le cas des vitesses aléatoires.

On verra plus loin qu'utiliser la réinitialisation nous permet d'éliminer les erreurs numériques flagrantes et d'obtenir la véritable solution pour cet exemple. Deux méthodes de réinitialisation seront expliquées et comparées. Une manière de réduire grandement le temps de calcul de la méthode *Level-Set*, rendue possible par la réinitialisation, sera ensuite présentée. Le chapitre se terminera par un exemple qui combine les différentes améliorations présentées jusqu'à maintenant et qui montre que la réinitialisation permet de bien traiter le cas des vitesses discontinues.

2.1. RÉINITIALISATION AVEC EDP

La méthode présentée ici est due à Rouy et Tourin [10] et à Sussman, Smereka et Osher [6]. L'idée est que, pour une surface $\phi_0(x, y, \tau)$ donnée, il existe une équation aux dérivées partielles qui a pour solution à l'état stationnaire la fonction de distance signée à la courbe de niveau 0 de $\phi_0(x, y, \tau)$:

$$\frac{\partial \phi}{\partial \tau} + \text{signe}(\phi_0)(\|\nabla \phi\| - 1) = 0 \quad (2.1.1)$$

où $\text{signe}(\phi_0)$ vaut 1 où ϕ_0 est positive, -1 où elle est négative et 0 où elle est nulle.

Cette équation revient à faire évoluer notre surface $\phi_0(x, y, \tau)$ dans le temps (le temps ici n'a rien à voir avec le temps réel correspondant à la propagation du feu) jusqu'à un état stationnaire. Il reste à vérifier qu'on atteint bien un état stationnaire, que cet état correspond à l'état cherché (une fonction de distance signée) et que le niveau 0 n'a pas été modifié. Cette dernière affirmation n'est pas toujours vraie, mais l'utilisation de $\text{signe}(\phi_0)$ au lieu de $\text{signe}(\phi)$ tend à corriger les légers déplacements de l'interface [6]. Aussi, aux points où ϕ est nulle, $\text{signe}(\phi_0)$ est nulle et l'équation vérifiée est :

$$\frac{\partial \phi}{\partial \tau} = 0 \quad (2.1.2)$$

et donc il n'y a pas de mouvement. Si un état stationnaire est atteint, la dérivée est nulle et il nous reste alors $\text{signe}(\phi_0)(\|\nabla \phi\| - 1) = 0$. Ceci sera vérifié si la fonction ϕ satisfait le critère suivant :

$$\|\nabla \phi\| = 1 \quad (2.1.3)$$

On a que si une fonction est une fonction de distance signée, elle satisfait ce critère. Aussi, on peut montrer qu'une fonction satisfaisant ce critère est, sauf dans certaines situations pathologiques, une fonction de distance signée. Pour un argument justifiant que le signe est le bon et pour une discussion sur la convergence à un état stationnaire, on réfère le lecteur à [6] et [10].

Pour comprendre comment agit l'équation de réinitialisation, on peut considérer le terme $\text{signe}(\phi_0)\|\nabla\phi\|$ comme un terme de déplacement dans une direction parallèle à la normale au front (la fonction $\frac{\text{signe}(\phi_0)\nabla\phi}{\|\nabla\phi\|}$ représente alors la vitesse d'advection). La fonction $\text{signe}(\phi_0)$ reflète donc le sens de propagation de l'information. Les seuls points où on est sûr que la fonction de distance signée est bien définie sont les points du front. On veut donc que l'information parte du front pour arriver aux autres points du domaine. La fonction $\text{signe}(\phi_0)$ est positive à l'extérieur du front de flamme, ce qui implique que la vitesse de propagation est positive et en direction de la normale, alors l'information part du front pour arriver aux autres points à l'extérieur du front. Elle est négative à l'intérieur, dans ce cas la vitesse de propagation est négative et en direction de la normale et l'information part de l'interface pour arriver aux points intérieurs.

Trouver la solution à l'état stationnaire de cette équation après chaque itération de la résolution de l'équation de propagation de feu nous assurera que notre fonction *Level-Set* demeurera suffisamment proche d'une fonction distance signée en tout temps.

2.1.1. Algorithme

La discrétisation de l'équation utilisée ici est celle suggérée dans [5]. L'idée générale est d'utiliser un schéma amont. On choisit les dérivées en utilisant le fait que l'information se propage dans une direction normale au front de flamme pour les points à l'extérieur du front, et dans la direction inverse à la normale pour les points intérieurs.

Soient :

- $\Delta\tau$ le pas en pseudo-temps (le temps fictif utilisé dans l'équation de réinitialisation)

- h le pas en espace
- $\tilde{\phi}^0$ notre condition initiale pour l'EDP, obtenue après une itération de propagation du front de flamme.

Discrétisons la dérivée pseudo-temporelle de l'équation (2.1.1) avec un schéma Euler-avant. La valeur de $\tilde{\phi}_{i,j}$ à l'itération $m+1$ dans le pseudo-temps sera donnée par :

$$\tilde{\phi}_{i,j}^{m+1} = \tilde{\phi}_{i,j}^m - \Delta\tau \cdot \text{signe}(\tilde{\phi}_{i,j}^0) \cdot \left(\sqrt{\left(\frac{\partial\tilde{\phi}}{\partial x}\right)^2 \Big|_{i,j}^m + \left(\frac{\partial\tilde{\phi}}{\partial y}\right)^2 \Big|_{i,j}^m} - 1 \right) \quad (2.1.4)$$

Notons qu'on utilise le même maillage que pour la résolution de l'équation de propagation de feu. Il reste à considérer la fonction *signe* et les dérivées spatiales. Il est préférable de prendre une fonction *signe* lissée pour la résolution numérique de l'équation. Un choix possible est :

$$\text{signe}(\tilde{\phi}^0) = \frac{\tilde{\phi}^0}{\sqrt{(\tilde{\phi}^0)^2 + h^2}} \quad (2.1.5)$$

La figure 2.4 montre que cette fonction vaut 1 ou -1 pour les points loin de zéro, mais que la discontinuité en zéro est lissée.

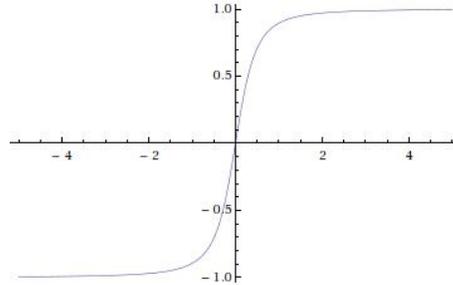


FIG. 2.4. Fonction $\text{signe}(x)$ pour $h = 0.5$ et x entre -5 et 5

Pour les dérivées spatiales, on utilise les équations suivantes (voir chapitre 1 section 1.1) :

$$\left(\frac{\partial\tilde{\phi}}{\partial x}\right)^2 \Big|_{i,j}^m = \begin{cases} \max(\max(\tilde{\phi}_x^-, 0)^2, \min(\tilde{\phi}_x^+, 0)^2) & \text{si } \text{signe}(\tilde{\phi}^0) \geq 0 \\ \max(\min(\tilde{\phi}_x^-, 0)^2, \max(\tilde{\phi}_x^+, 0)^2) & \text{si } \text{signe}(\tilde{\phi}^0) < 0 \end{cases} \quad (2.1.6)$$

avec $\tilde{\phi}_x^-$ la différence arrière en x de $\tilde{\phi}_{i,j}^m$ et $\tilde{\phi}_x^+$ la différence avant en x de $\tilde{\phi}_{i,j}^m$. On définit ensuite $\left(\frac{\partial \tilde{\phi}}{\partial y}\right) \Big|_{i,j}^m$ de façon similaire.

Pour choisir un pas de pseudo-temps $\Delta\tau$, on utilise les mêmes arguments qu'à la section 1.1 du chapitre 1.

Il reste à fixer un critère d'arrêt sur les itérations de réinitialisation. Une possibilité est de fixer arbitrairement un nombre d'itérations. Sinon, dans l'article [6] les auteurs suggèrent le critère d'arrêt suivant. Définissons :

$$c = \sum_{i,j \in N} |\tilde{\phi}_{i,j}^{m+1} - \tilde{\phi}_{i,j}^m| - N \cdot \Delta\tau \cdot h^2 \quad (2.1.7)$$

où N représente le nombre de points dans le domaine.

On considère être à l'état stationnaire lorsque $c < 0$. Dans la plupart des cas, cette condition est satisfaite au bout de quelques itérations (moins de 5).

2.1.2. Illustration de la réinitialisation

Pour se convaincre que cette méthode de réinitialisation avec EDP fonctionne, on considère un cas élémentaire :

Soit

$$\phi(x, y) = \begin{cases} 1 & \text{si } x^2 + y^2 > 16 \\ 0 & \text{si } x^2 + y^2 = 16 \\ -1 & \text{si } x^2 + y^2 < 16 \end{cases} \quad (2.1.8)$$

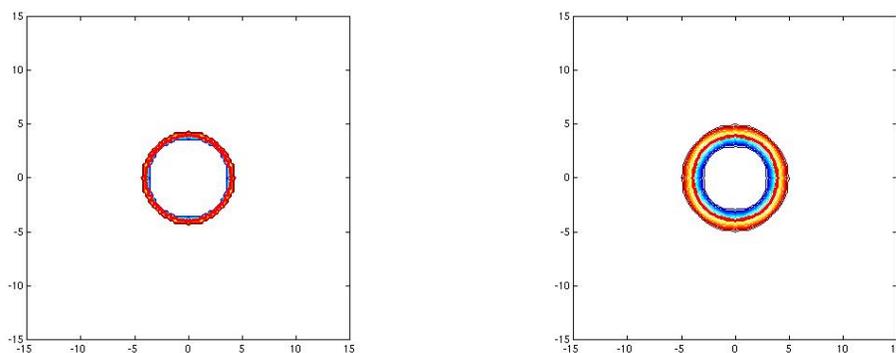
On a comme courbe de niveau zéro de cette fonction le cercle de rayon 4 centré à l'origine. Cette fonction est très loin d'être une fonction de distance signée à cette courbe.

Appliquons notre algorithme de réinitialisation en utilisant les paramètres donnés au tableau 2.2.

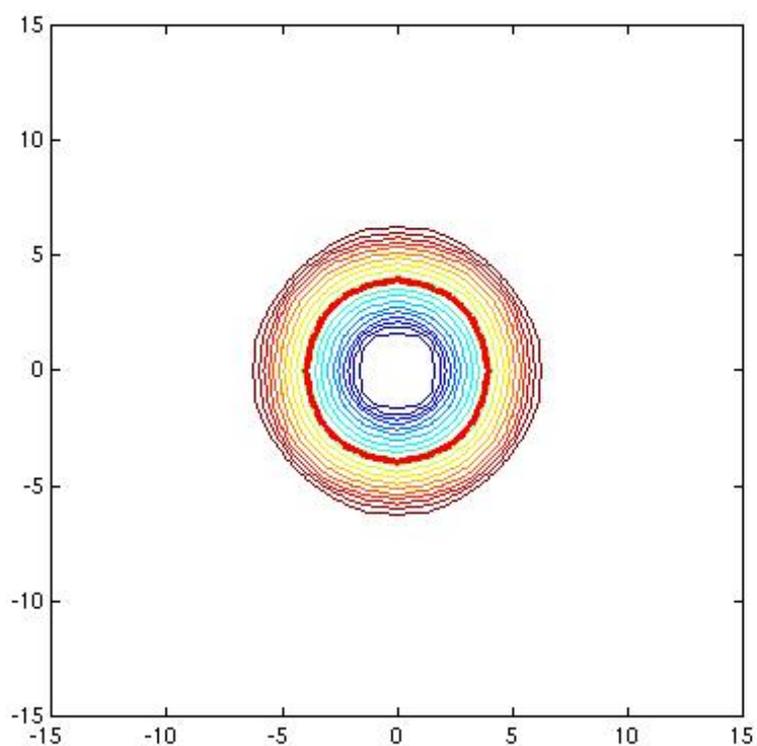
Paramètres	Valeurs
h	0.25
$\Delta\tau$	0.125
Taille du domaine en x	$[-15, 15]$
Taille du domaine en y	$[-15, 15]$

TAB. 2.2. Paramètres utilisés lors du test de validation de la réinitialisation

La figure 2.5 montre les résultats obtenus si on prend cette condition initiale pour résoudre l'équation (2.1.1) (la courbe de niveau zéro est indiqué en gros en rouge).



(a) Fonction *Level-Set* après 1 itération de ré-initialisation (b) Fonction *Level-Set* après 5 itérations de ré-initialisation



(c) Fonction *Level-Set* après 15 itérations de ré-initialisation

FIG. 2.5. Fonction *Level-Set* à des pseudo-temps différents lors du processus de réinitialisation

Au fil des itérations, la fonction ressemble de plus en plus à une fonction de distance signée (une telle fonction est caractérisée par des courbes de niveaux équidistantes).

2.1.3. Comparaison des méthodes de réinitialisation possibles

Un autre test à faire est de comparer le résultat de la propagation d'un feu en utilisant la réinitialisation au résultat obtenu sans la réinitialisation. On peut aussi comparer la méthode de réinitialisation avec EDP à une méthode purement géométrique. Pour obtenir la fonction de distance signée à notre front géométriquement, on peut mesurer explicitement la distance entre chaque point du domaine et notre front de flamme. Un algorithme simple pour faire ceci est :

- (1) Extraire le front de flamme de la fonction *Level-Set*. On prend une liste de points qui sont sur le front.
- (2) Prendre cette liste de points et obtenir une approximation linéaire par morceaux du front.
- (3) Pour chaque point (x, y) , mesurer la distance de ce point à tous les segments de droites. La plus petite distance trouvée sera la distance du point au front de flamme.

On va donc résoudre l'EDP de propagation de feu et après chaque itération de propagation, réinitialiser avec chacune des 3 manières suivantes :

- (1) En utilisant la réinitialisation avec EDP
- (2) En utilisant la réinitialisation basée sur la méthode géométrique décrite ci-dessus
- (3) En ne réinitialisant pas

Prenons le cas d'un front de flamme circulaire centré à l'origine au temps initial, qui se propage avec les vitesses de propagation suivantes :

$$a = \left| \sin \left(3 \cdot \arctan \left(\frac{y}{x + \varepsilon} \right) \right) \right| + 0.5 \quad (2.1.9)$$

$$b = \left| \sin \left(3 \cdot \arctan \left(\frac{y}{x + \varepsilon} \right) \right) \right| + 0.5 \quad (2.1.10)$$

L'équation résolue est donc

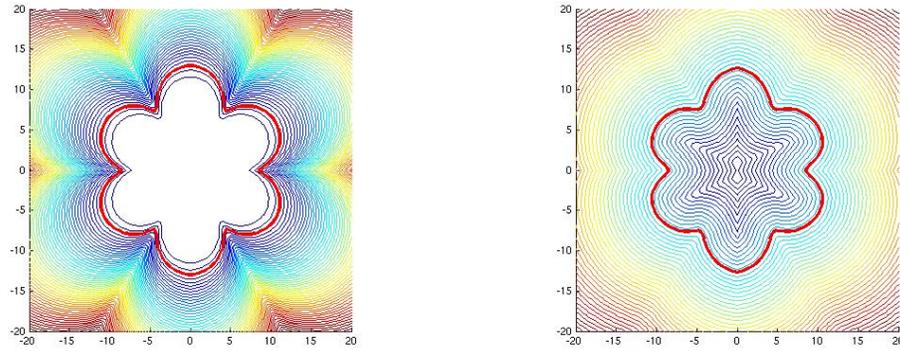
$$\phi_t + \|A\nabla\phi\| = 0 \quad (2.1.11)$$

Voici le résultat obtenu par les 3 méthodes après 10 itérations. Le tableau 2.3 donne les paramètres utilisés.

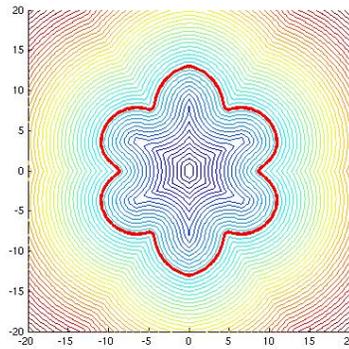
Paramètres	Valeurs
h	0.5
Δt	$\frac{0.5h}{\max(a , b , c)}$
$\Delta \tau$	0.25
Taille du domaine en x	$[-20, 20]$
Taille du domaine en y	$[-20, 20]$
Itérations de propagation de feu	50
Itérations de réinitialisations	4

TAB. 2.3. Paramètres utilisés lors de la comparaison des méthodes de réinitialisation

Les solutions obtenues sont présentées à la figure 2.6.



(a) Solution obtenue en ne réinitialisant pas (b) Solution obtenue en réinitialisant avec EDP



(c) Solution obtenue en réinitialisant avec la méthode géométrique

FIG. 2.6. Solutions de l'équation (2.1.11) obtenues des 3 manières différentes au temps $t = 8.33$.

On voit que la solution obtenue sans réinitialisation présente les contours les plus éloignés des contours d'une fonction de distance signée. On peut aussi remarquer que l'approximation linéaire par morceaux utilisée par la réinitialisation géométrique a une influence sur les contours de la fonction *Level-Set*, qui sont moins lisses que dans le cas de la réinitialisation avec EDP.

On peut aussi estimer le temps de calcul nécessaire à chacune des méthodes de réinitialisation. Bien entendu, ne pas faire de réinitialisation n'augmente pas le temps de calcul. Pour la méthode avec EDP, il faut faire à chaque itération de réinitialisation un calcul en chaque point du domaine. Si le domaine est de taille $n \times n$, et qu'on a besoin de k itérations en pseudo-temps, le temps de calcul sera

de l'ordre de kn^2 . Cependant, on introduit à la prochaine section une méthode basée sur la réinitialisation qui permet de diminuer beaucoup le temps de calcul de la méthode *Level-Set*, et alors réinitialiser nous coûtera moins cher en bout de ligne que ne pas réinitialiser.

Pour la méthode géométrique, il faut comparer chaque point du domaine à chaque segment utilisé pour représenter le front (ce nombre de segment est proportionnel à n). Le nombre d'opérations est donc d'ordre n^3 . Donc, la réinitialisation avec EDP coûte moins cher si on a besoin de moins que n itérations de réinitialisation (c'est vrai dans la plupart des cas).

Un dernier argument contre la réinitialisation géométrique est qu'elle n'utilise pas toute l'information disponible pour ramener notre surface à une fonction de distance signée. En effet, elle n'utilise que le niveau zéro de notre surface, ignorant qu'en tous les points du domaine notre fonction est déjà assez proche d'une fonction de distance signée. De plus, les contours de la solution ont tendance à être moins lisse, à cause de l'approximation linéaire par morceaux du front. Il est vrai qu'on peut utiliser une meilleure représentation du front (par exemple, des splines cubiques), mais alors le temps de calcul devient prohibitif.

En conclusion, on peut résumer la situation avec les tableaux 2.4 et 2.5.

Position	Méthode
1	Réinitialisation avec EDP
2	Réinitialisation géométrique
3	Sans réinitialisation

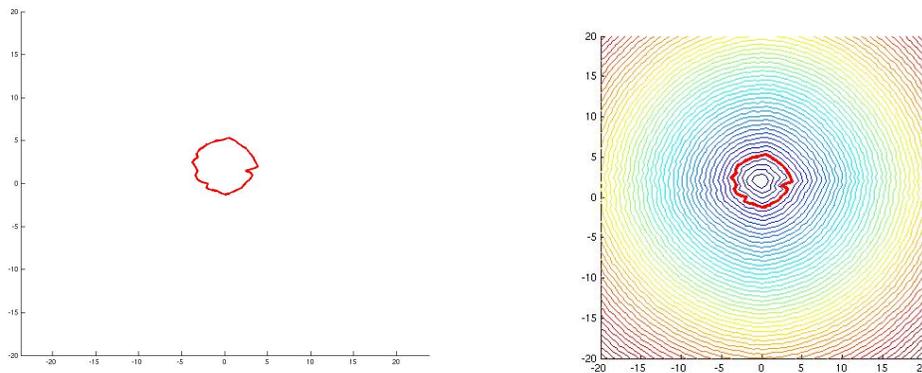
TAB. 2.4. Classement des méthodes de réinitialisation en fonction de la précision du résultat

Position	Méthode
1	Sans réinitialisation
2	Réinitialisation avec EDP
3	Réinitialisation géométrique

TAB. 2.5. Classement des méthodes de réinitialisation en fonction du temps de calcul

2.1.4. Retour sur l'exemple problématique du début du chapitre

Revenons maintenant à l'exemple problématique du début du chapitre. La figure 2.7 illustre ce qu'on obtient dans ce cas en utilisant la réinitialisation après 20 itérations.



(a) Front final

(b) Courbes de niveau de la fonction *Level-Set* correspondante

FIG. 2.7. Solution au cas problématique en utilisant la réinitialisation au temps $t = 0.5$

Ce cas, qui n'était pas traitable sans réinitialisation, est maintenant résoluble. La réinitialisation nous a permis de conserver une fonction *Level-Set* assez régulière en tout point du domaine et il n'y a maintenant plus rien à craindre lors de l'approximation des dérivées.

Pour se convaincre que la solution trouvée est la véritable solution au problème, on peut raffiner le maillage et vérifier si on a convergence. Voici la solution obtenue en multipliant par deux le nombre de points en x et en y . Les valeurs

des vitesses aux nouveaux points ont été obtenues par interpolation bilinéaire à partir des vitesses originales. La figure 2.8 montre la position du front en $t=0.5$ ainsi que les courbes de niveau de la fonction ϕ correspondante.

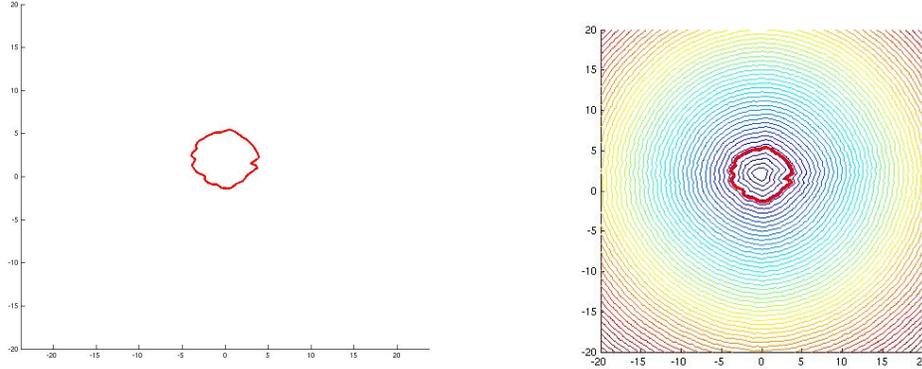
(a) Front au temps $t = 0.5$ (b) Courbes de niveau de la fonction *Level-Set* correspondante

FIG. 2.8. Solution au cas problématique en utilisant la réinitialisation au temps $t = 0.5$ pour le maillage raffiné

Graphiquement, les solutions se ressemblent beaucoup. Vérifions analytiquement qu'en diminuant le maillage, la solution converge. Définissons la solution de référence ϕ_{ref} obtenue en divisant en 8 dans les deux directions le maillage original et en utilisant l'interpolation bilinéaire pour obtenir les vitesses aux nouveaux points. Définissons ensuite ϕ_0 , ϕ_1 et ϕ_2 les solutions obtenues en divisant en 2 le maillage en x et y 0,1 et 2 fois. Définissons aussi N_k comme le nombre de points dans le domaine pour $k = 0, 1, 2$.

Si on a convergence, l'écart entre les solutions $\frac{\|\phi_k - \phi_{ref}\|_1}{N_k}$ devrait diminuer quand k augmente. Notons que pour faire cette comparaison, on échantillonne ϕ_{ref} pour qu'elle soit de la même taille que les ϕ_k . Le tableau 2.6 montre les résultats.

h	N-le nombre de points dans le domaine	$\frac{\ \phi_k - \phi_{ref}\ _1}{N}$
0.5	6561	0.2779
0.25	25921	0.1500
0.125	103041	0.0615

TAB. 2.6. Erreur au temps $t = 0.5$ entre la solution obtenue avec différentes valeurs de h et la solution de référence

Les résultats indiquent qu'en raffinant le maillage, on converge vers une solution. Ceci montre qu'utiliser la réinitialisation nous permet de trouver la solution dans ce cas, qui était autrement impossible à traiter.

2.2. MÉTHODE NARROW-BAND

Un des désavantages de la méthode *Level-Set* par rapport à une méthode de marqueurs est le coût en temps de calcul. En effet, pour une méthode de marqueur, ce coût est proportionnel à la longueur du front de flamme. Pour simuler la propagation d'un feu avec la méthode *Level-Set*, il faut résoudre une EDP sur tout notre domaine. Ceci exige qu'à chaque itération, un calcul soit fait en chaque point de notre maillage. Pour l'application aux feux de forêt, les tailles de domaines sont très grandes et la méthode *Level-Set* peut devenir très coûteuse en temps de calcul. Il est cependant possible de réduire le nombre de points où il faut faire des calculs, pour obtenir un coût en temps de calcul de même ordre que le coût relié à la méthode des marqueurs.

Puisqu'on ne s'intéresse qu'à la position du front, on peut s'en tenir à ne mettre ϕ à jour qu'aux points qui en sont suffisamment proches. La réinitialisation nous fournit un critère pour trouver ces points : plus un point (x, y) est près du front, plus $|\phi(x, y)|$ est petit, puisqu'on réinitialise toujours ϕ à une fonction de distance signée. Ceci est l'idée derrière la méthode "narrow-band" (Chopp [11]). À chaque itération de la méthode *Level-Set*, on doit :

- (1) choisir les points où on calcule l'EDP. On peut prendre les points :

$$\{(x, y) : |\phi(x, y)| \leq kh\} \tag{2.2.1}$$

où k est un entier.

- (2) calculer une itération de méthode *Level-Set*
- (3) choisir une largeur de bande à l'intérieur de laquelle on réinitialise. On doit prendre cette bande plus grande que la bande pour le calcul de l'EDP pour s'assurer que la fonction *Level-Set* est lisse où on calcule l'EDP. On peut prendre les points $\{(x, y) : |\phi(x, y)| \leq 2kh\}$
- (4) poser comme valeurs de ϕ aux points (x, y) à l'extérieur de la seconde bande comme : $\phi(x, y) = (\pm 1.1)2kh$ pour ne pas conserver des valeurs de ϕ qui ne correspondent plus à la distance d'un point au front (on prend "+" si (x, y) est non-brûlé et "-" si (x, y) est brûlé).

- (5) rajouter à la bande sur la réinitialisation tous les points voisins à un point dans la bande de réinitialisation. Ceci nous assure que la bande de calcul peut grandir à chaque itération. Sans cette étape, les points posés comme étant à l'extérieur de la bande à une itération ne pourraient jamais y pénétrer puisque la valeur de la fonction *Level-Set* en ces points est maintenant de $(\pm 1.1)2kh$ et cette valeur ne satisfait pas le critère sur les bandes.
- (6) réinitialiser la fonction *Level-Set* aux points de la deuxième bande.

La figure 2.9 montre une illustration graphique de la méthode. Les points à l'intérieur de la bande sur l'EDP sont indiqués en bleu et ceux à l'intérieur de la bande sur la réinitialisation sont indiqués en vert.

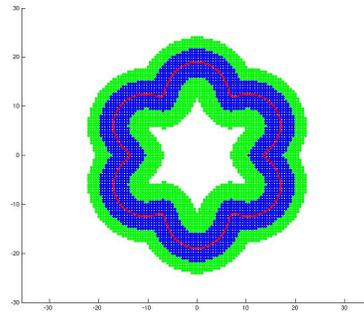


FIG. 2.9. Points à l'intérieur des bandes

Les courbes de niveau de la fonction *Level-Set* correspondante sont tracées à la figure 2.10.

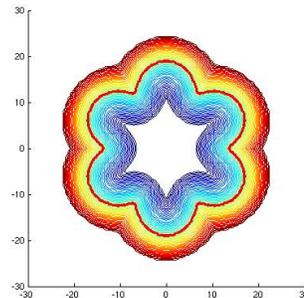
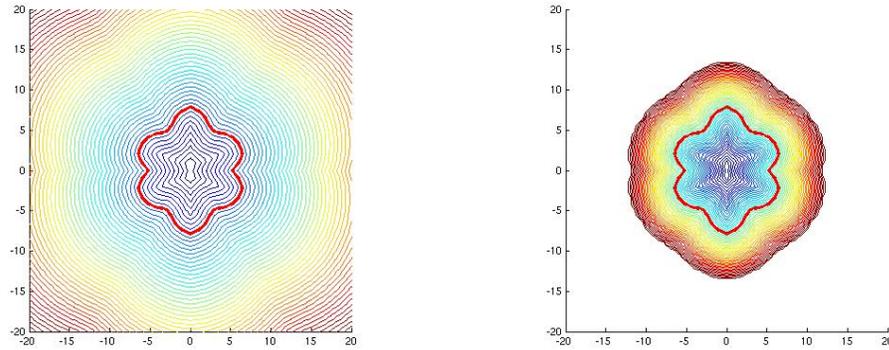


FIG. 2.10. Fonction *Level-Set* correspondante

Un test à effectuer serait de vérifier que le résultat est le même quand on utilise la méthode narrow-band que si on ne l'utilise pas. Reprenons l'exemple de la section 2.1.3. On avait trouvé la solution en réinitialisant avec EDP et en faisant les calculs partout sur le domaine. Reprenons les même paramètres, mais appliquons la méthode “narrow-band”.



(a) Courbes de niveau de la fonction *Level-Set* (b) Courbes de niveau de la fonction *Level-Set* au temps $t = 5$ en n'utilisant pas la méthode *Set* au temps $t = 5$ en utilisant la méthode “narrow-band”

FIG. 2.11. Comparaison des solutions obtenues des deux manières différentes

On voit graphiquement à la figure 2.11 que la position du niveau zéro (en rouge) est la même.

On peut ensuite se demander quel est le gain en temps apporté par la méthode narrow-band. Supposons qu'on a un domaine de taille $N \times N$. Le temps de calcul de la méthode *Level-Set* est $O(N^2)$. Le temps de calcul de la méthode “narrow-band” devrait être $O(N)$, puisque ce temps est proportionnel à la longueur du front, un objet fondamentalement unidimensionnel. Voici les temps de calcul mesurés dans le cas de la propagation d'un front circulaire avec une vitesse de propagation normale au front pour quelques tailles de domaine.

Les paramètres utilisés sont donnés au tableau 2.7.

Δx	0.5
Δt	0.25
Largeur de la bande sur l'EDP	$6 \cdot \Delta x$
Largeur de la bande sur la réinitialisation	$12 \cdot \Delta x$
Nombre d'itérations	100
Nombre d'itérations de réinitialisation	4

TAB. 2.7. Paramètres utilisés pour les tests de temps de calcul

L'équation résolue est la suivante :

$$\phi_t + \|\nabla\phi\| = 0 \quad (2.2.2)$$

avec comme condition initiale :

$$\phi_0 = \sqrt{x^2 + y^2} - 1 \quad (2.2.3)$$

Les temps de calcul totaux (tableau 2.8) ont été mesurés pour 3 situations : en utilisant la méthode “narrow-band”, sans l'utiliser et sans réinitialisation (quand on ne réinitialise pas, on ne peut employer la méthode narrow-band). Les calculs ont été effectués sur un des ordinateurs au département de mathématiques et statistique de l'Université de Montréal. Chaque temps écrit ici est obtenu grâce à la commande tic-toc de *Matlab*.

Nombre de points	Avec bandes(s)	Sans bandes(s)	Sans réinitialisation(s)
101×101	3.4	29.0	11.0
201×201	8.5	124.7	49.5
401×401	25.6	539.3	205.9

TAB. 2.8. Résultats des tests de temps de calcul pour plusieurs tailles de domaine différentes

On peut voir que la méthode narrow-band nous permet de sauver beaucoup de temps, et que ce gain est de plus en plus marqué lorsque la taille du domaine augmente. On a aussi que réinitialiser et utiliser la méthode “narrow-band” est

plus économique que ne pas réinitialiser et ne pas utiliser la méthode “narrow-band“, alors le coût relié à la réinitialisation est amplement justifié. L’utilisation de cette technique sera très importante lorsqu’on travaillera avec des données réelles, car dans ce cas les domaines seront trop grands pour pouvoir appliquer la méthode *Level-Set* originale.

Chapitre 3

CONTOURNEMENT D'OBSTACLES

Pour rendre le modèle de propagation plus réaliste, il est nécessaire de tenir compte d'endroits où le feu ne peut pas passer. Ce chapitre présente une manière d'introduire des obstacles infranchissables dans le domaine de propagation du feu.

3.1. MODÉLISATION DES OBSTACLES

Dans un premier temps, on représentera les obstacles comme des segments de droite. Pour empêcher le feu de traverser ces segments de droite, il faudra forcer les vitesses de propagation à être nulles aux mailles qui les entourent. On devra donc trouver toutes les mailles à une distance quelconque de tous les segments. Cette distance devra être assez grande pour qu'on soit sûr que le feu ne traverse pas les obstacles, mais assez petite pour que les mailles bloquées représentent le plus possible un objet unidimensionnel. On choisit dans les tests subséquents de mettre les vitesses de propagation à zéro à toutes les mailles qui sont à une distance de moins de $1.5h$ d'un segment représentant un obstacle (h est ici le pas en espace).

Le dernier exemple du chapitre utilisera comme représentation d'obstacles un ensemble de mailles infranchissables pour le feu.

Dans les deux cas, on représente les obstacles comme des mailles où les vitesses de propagation du feu sont nulles. Définissons $M(x, y)$ en chaque point du domaine :

$$M(x, y) = \begin{cases} 1 & \text{si } (x, y) \text{ n'est pas un obstacle} \\ 0 & \text{si } (x, y) \text{ est un obstacle} \end{cases} \quad (3.1.1)$$

On s'assurera aussi que la fonction *Level-Set* demeure positive sur les obstacles après chaque itération en prenant la valeur absolue de la fonction aux points où $M(x, y) = 0$. L'équation de propagation reflétera la présence d'obstacle en devenant :

$$\partial \frac{\phi(x, y, t)}{\partial t} + M(x, y) \|A \nabla \phi(x, y, t)\| + M(x, y) \vec{C}(x, y) \cdot \nabla \phi(x, y, t) = 0 \quad (3.1.2)$$

Maintenant, supposons qu'on a un front de flamme circulaire en absence de vent et avec des vitesses de propagations isotropes et constantes et égales à 1. On résoud donc l'équation suivante :

$$\phi_t + M(x, y) \cdot \|\nabla \phi\| = 0 \quad (3.1.3)$$

avec comme condition initiale :

$$\phi_0 = \sqrt{x^2 + y^2} - 1 \quad (3.1.4)$$

On utilise ici toutes les techniques décrites au chapitre 2 : on réinitialise la fonction level-set à une fonction de distance signée et on utilise la méthode narrow-band.

Un obstacle a été ajouté au domaine, tel que montré à la figure 3.1. Les points infranchissables sont indiqués en noir et les obstacles sont tracés en bleu.

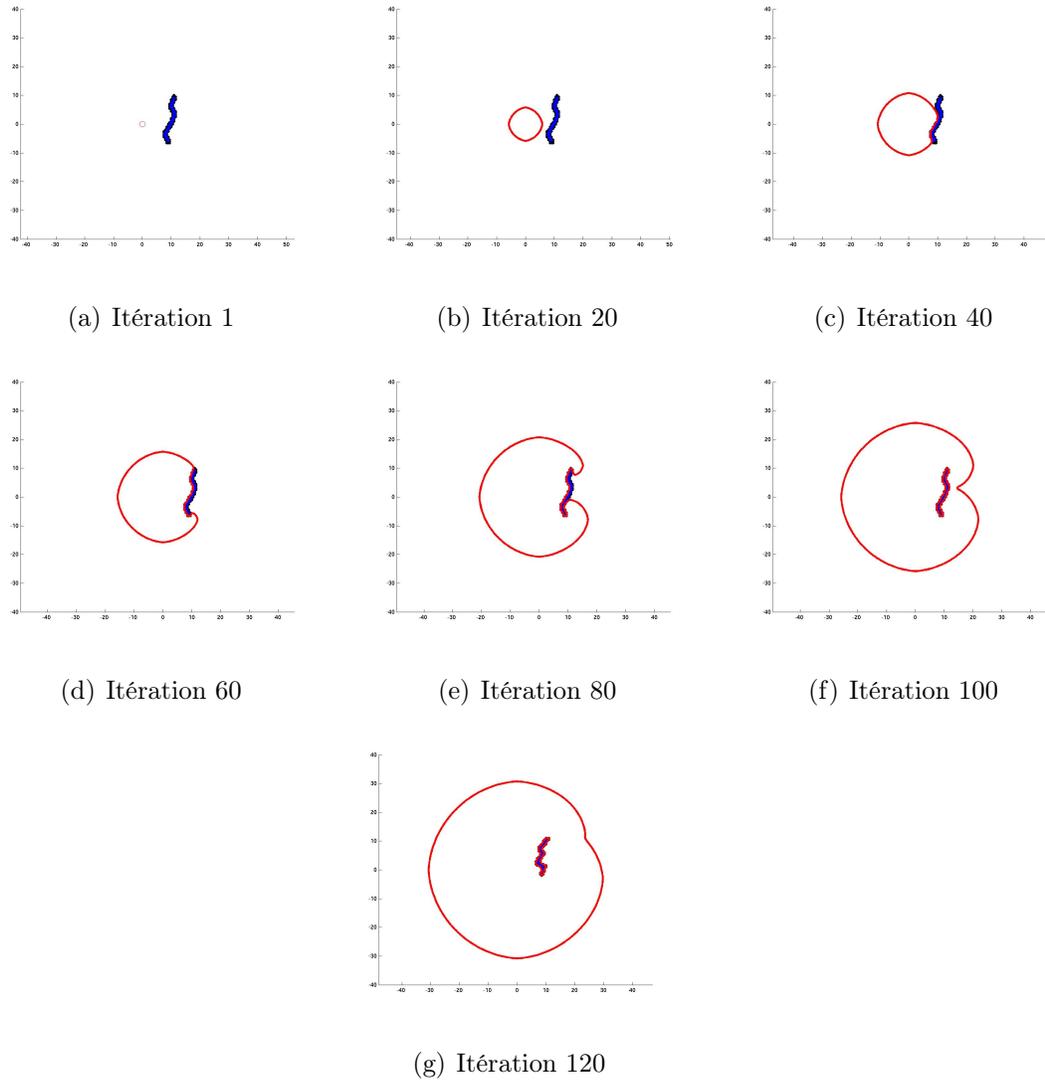


FIG. 3.1. Premier exemple de contournement d'obstacle

On a pu observer que le front contourne l'obstacle sans le traverser. On peut voir que la fonction ϕ reste une fonction de distance signée grâce à la réinitialisation à la figure 3.2.

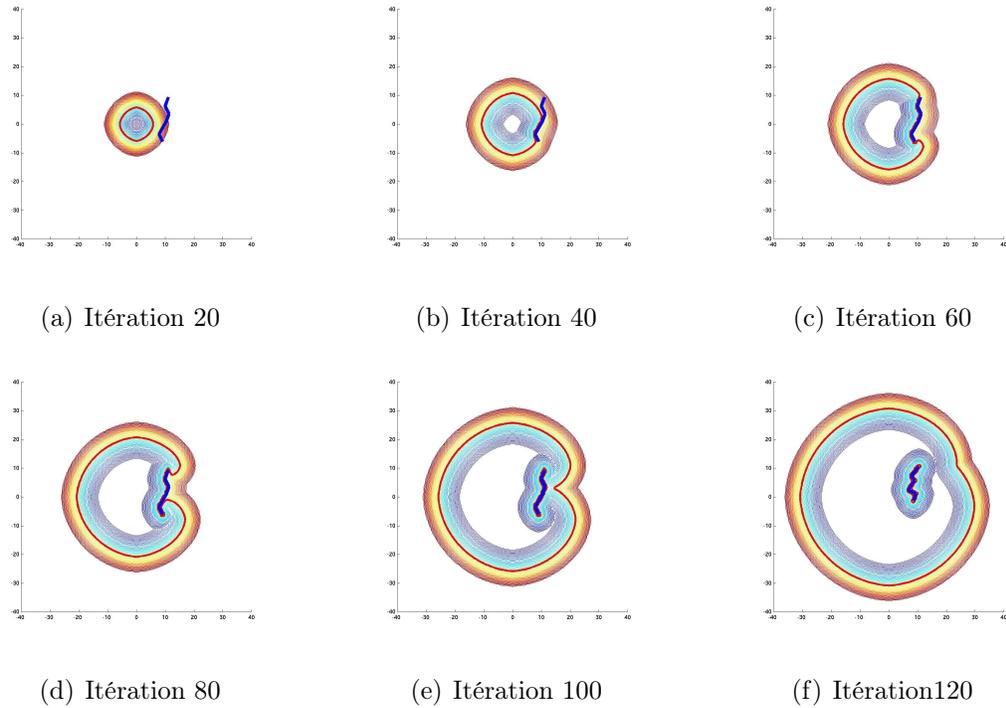


FIG. 3.2. Premier exemple de contournement d'obstacle

Voici un autre exemple avec vitesses de propagation variables, vent et plusieurs obstacles. On résoud alors l'équation 3.1.2 avec le paramètre $c = 1$ en direction $\theta = \frac{\pi}{4}$. Prenons

$$a = 3 + 2 \cdot \sin\left(\frac{\pi \cdot y}{40}\right) \quad (3.1.5)$$

$$b = 2 + 1.5 \cdot \cos\left(\frac{\pi \cdot x}{30}\right) \quad (3.1.6)$$

On rappelle que

$$A = \begin{pmatrix} b & 0 \\ 0 & a \end{pmatrix} \cdot R_{\frac{\pi}{4}} \quad (3.1.7)$$

où $R_{\frac{\pi}{4}}$ est la matrice de rotation d'angle $\frac{\pi}{4}$.

Prenons la même condition initiale que pour l'exemple précédent et les paramètres donnés au tableau 3.1.

Paramètres	valeurs
domaine	$[-60,60] \times [-60,60]$
h	0.5
Δt	$\frac{0.5 \cdot h}{\max(a, b, c)}$
Largeur de la bande sur l'EDP	$6 \cdot h$
Largeur de la bande sur la réinitialisation	$12 \cdot h$
Nombre d'itérations	200

TAB. 3.1. Paramètres utilisés pour les tests de contournement d'obstacle

La figure 3.3 présente les résultats.

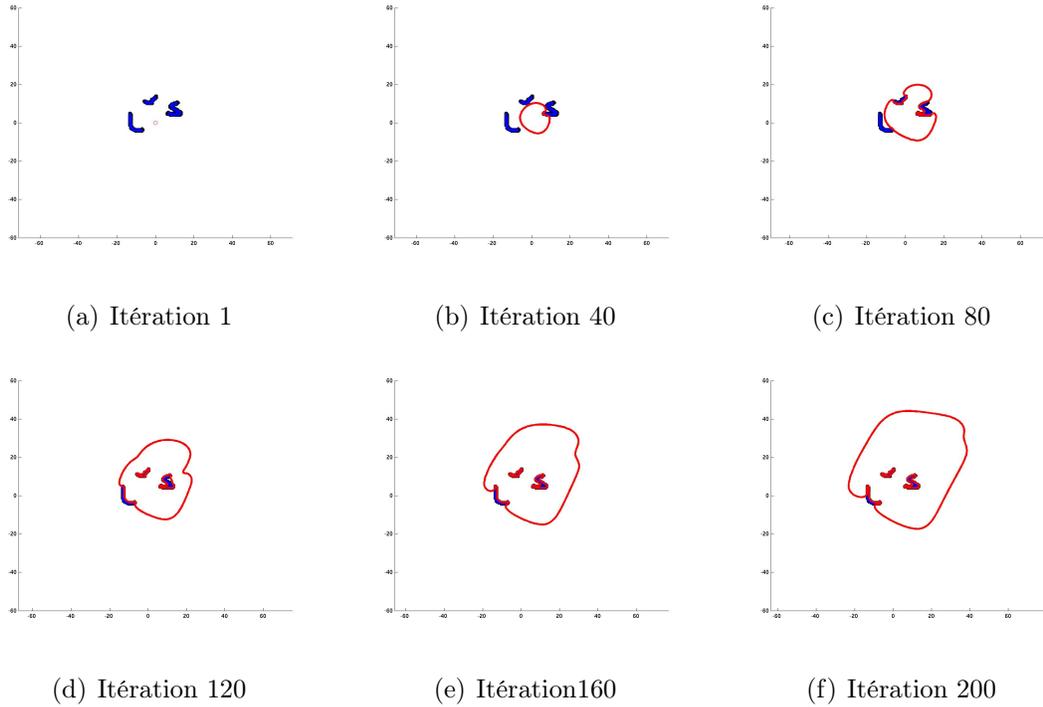
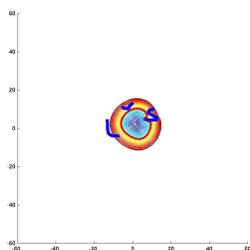
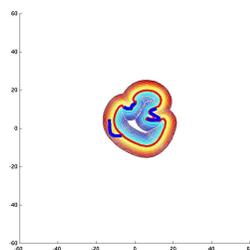


FIG. 3.3. Exemple de contournement d'obstacle en présence de vent

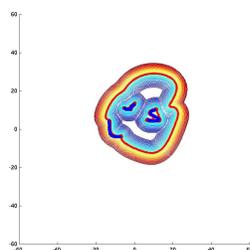
La figure 3.4 montre les courbes de niveau de la fonction *Level-Set* aux temps correspondants.



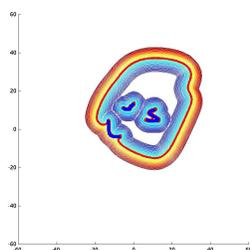
(a) Itération 40



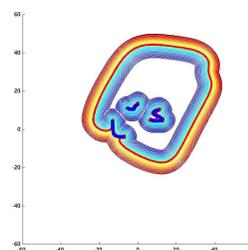
(b) Itération 80



(c) Itération 120



(d) Itération 160



(e) Itération 200

FIG. 3.4. Exemple de contournement d'obstacle en présence de vent

On remarque que dans ce cas aussi, la manière de traiter les obstacles est adéquate.

3.2. OBSTACLES ENTOURÉS

Au fil des itérations, des obstacles dans le domaine peuvent devenir complètement entourés par un front de flamme. Il restera toujours pour la suite de la simulation un front de flamme entourant ces obstacles, comme le montre la figure 3.4. On voit aussi sur cette figure que la fonction de distance signée trouvée par la réinitialisation est influencée par ces fronts. Ceci rajoute des calculs inutiles pour déterminer la véritable position du front et il est donc souhaitable d'éliminer les fronts entourant des obstacles. On peut s'en débarrasser en forçant la fonction *Level-Set* à être négative aux points entourés et en réinitialisant. Donc, après chaque itération de résolution de l'EDP de propagation du feu, on doit trouver ces fronts inutiles. Une manière de les détecter est de considérer chaque front de flamme présent dans le domaine. Pour chacun de ces fronts, on peut compter le nombre de points qu'il contient qui font partie d'un obstacle. Si suffisamment de points font partie d'un obstacle, on élimine le front. Reprenons l'exemple précédent et éliminons les fronts entourant des obstacles, tel qu'illustré à la figure 3.5.

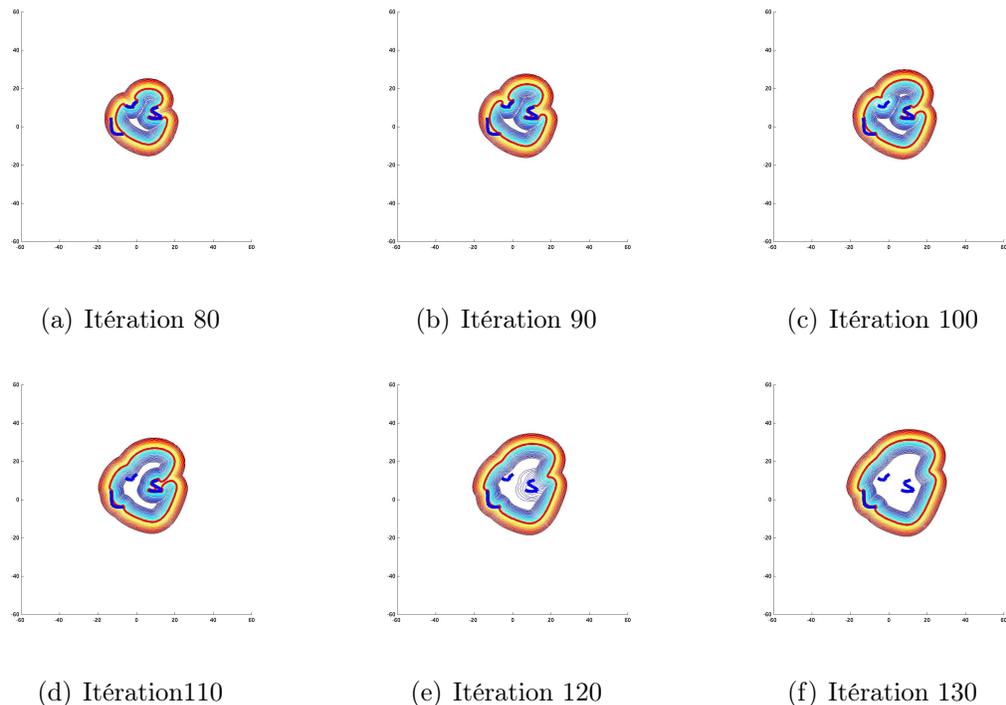


FIG. 3.5. Exemple de contournement d'obstacle en présence de vent

Une fois qu'on a détecté qu'un obstacle est entouré par un front et que ce front est suffisamment collé sur l'obstacle, on a forcé la fonction *Level-Set* à être négative aux points entourés, ce qui nous a permis d'éliminer le front intérieur. La réinitialisation nous permet ensuite d'effacer l'influence des fronts éliminés, comme on peut le voir à la figure correspondant à l'itération 130.

3.3. OBSTACLES, RÉINITIALISATION ET VITESSES DE PROPAGATION VARIABLES - UN EXEMPLE AVEC DONNÉES “RÉELLES”

Toutes les améliorations à la méthode suggérées aux chapitres 2 et 3 nous permettent d’aborder un cas réaliste. La réinitialisation nous permet de traiter le cas des vitesses discontinues sans trop de crainte et la méthode “narrow-band” nous laisse espérer que le temps de calcul sera acceptable. Les mailles où le feu ne peut pas passer sont maintenant aussi bien traitées.

Le gouvernement albertain rend disponible avec le logiciel *Prometheus* des données sur les paramètres physiques qui influent sur la propagation du feu (tels que la végétation en un point donné et la topographie) pour une partie de l’Alberta. Ce logiciel nous permet aussi de calculer les vitesses de propagation a et b correspondants à ces données en tout point d’un domaine. Voici un exemple utilisant ces données. L’équation (3.1.2) est résolue pour les coefficients a et b et c venant des données de *Prometheus*, avec un vent soufflant dans la direction $31\pi/16$. Pour le calcul des coefficients de vitesse, on n’utilise ici que les données sur la végétation. En chaque point de la grille, la sorte d’arbre présente est connue et on rattache à chaque variété une vitesse de propagation différente. Des points où le feu ne peut pas passer sont aussi connus. Le domaine est de taille environ $12km$ par $16km$ et les vitesses sont définies à intervalle de $25m$.

Sur les graphiques qui suivent (figure 3.6), la couleur du fond représente les vitesses de propagation (plus la couleur est pâle, plus le feu brûle vite). Les points noirs représentent des points infranchissables et la région brûlée est indiquée en rouge.

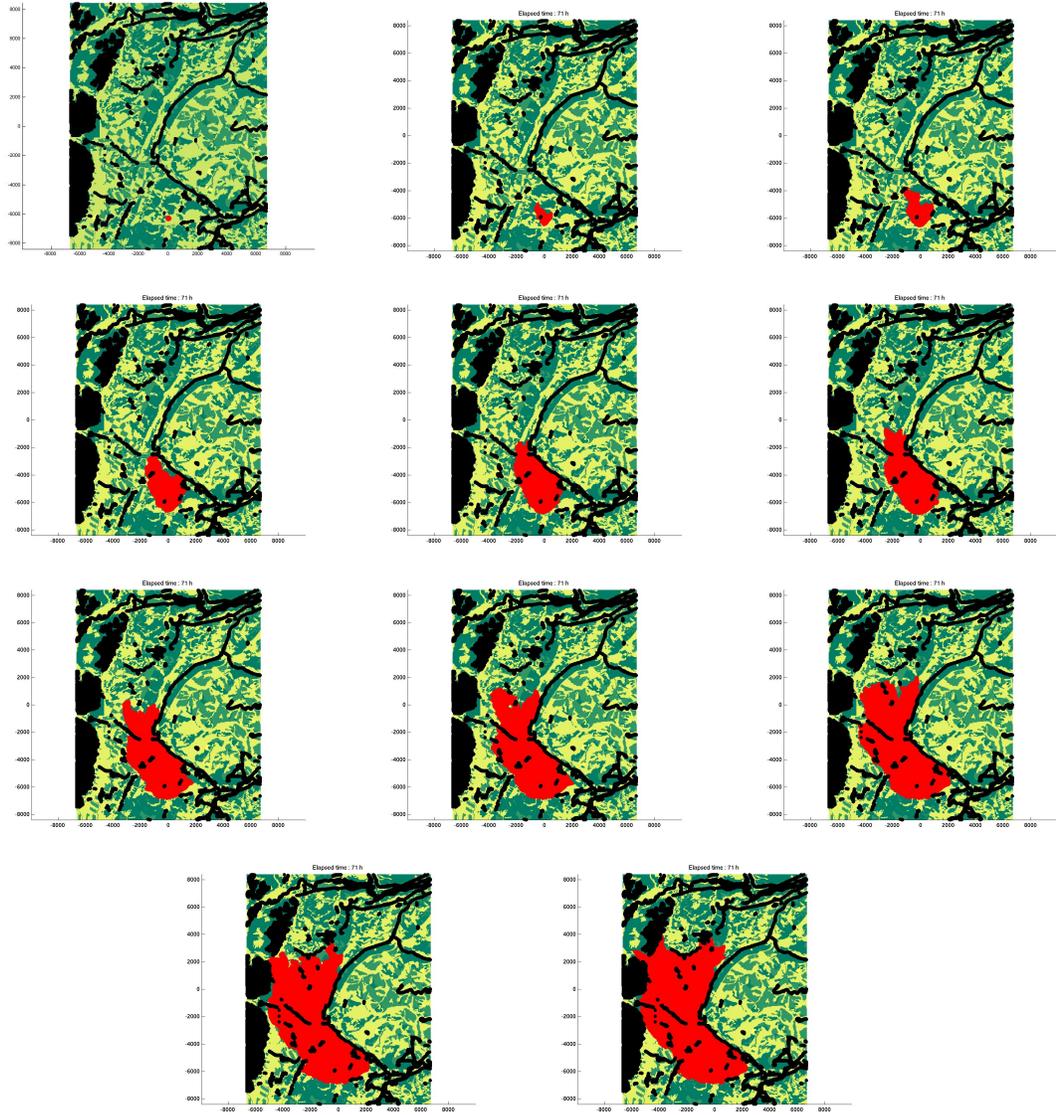
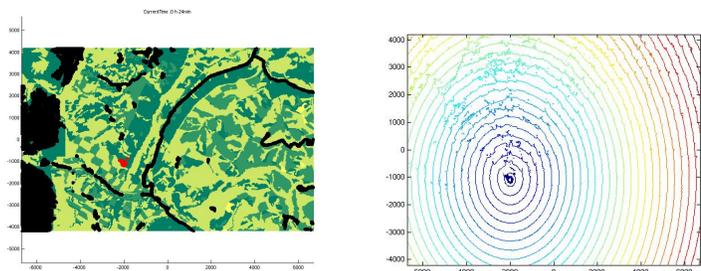


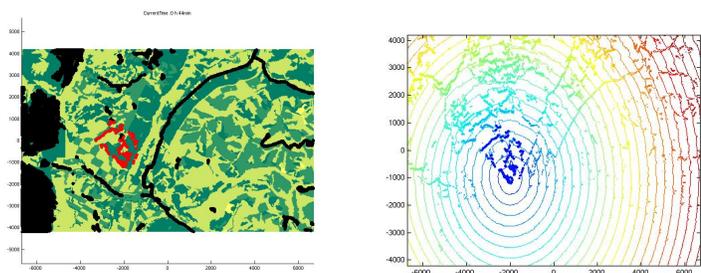
FIG. 3.6. Régions brûlées à chaque 100 itérations

Les résultats sont encourageants. L'algorithme a été capable de bien traiter le cas des vitesses de propagation de feu réalistes et les points infranchissables. Le calcul s'est fait dans un temps raisonnable, malgré le nombre élevé de points dans le domaine. Sans utiliser la méthode narrow-band, on a besoin de plus d'une heure pour faire le calcul tandis qu'en l'utilisant, le temps de calcul se compte en minutes. La prochaine étape sera de comparer les résultats de notre modèle avec ceux de *Prometheus* et avec les données provenant d'un vrai feu.

Pour se convaincre que les techniques introduites au chapitre 2 sont indispensables, on peut regarder ce qu'on obtient comme résultat en ne réinitialisant pas, tel qu'illustré à la figure 3.7.



(a) Résultats à l'itération 5 (b) Courbes de niveau correspondantes



(c) Résultats à l'itération 10 (d) Courbes de niveau correspondantes

FIG. 3.7. Régions brûlées à deux temps différents

On voit que les variations de vitesse rendent le problème intraitable sans utiliser la réinitialisation. Aussi, le gain en temps de calcul en utilisant la méthode narrow-band est vraiment nécessaire pour pouvoir étudier les tailles de domaine de propagation de feux réels.

Chapitre 4

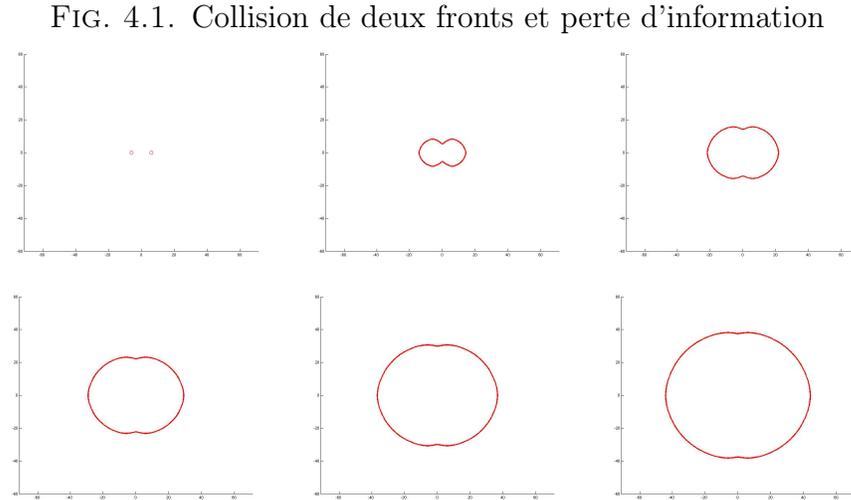
RECHERCHE DU POINT D'IGNITION D'UN FEU

Retrouver le point d'allumage d'un feu est un problème important pour l'industrie forestière, car connaître sa position peut aider les pompiers forestiers à trouver les causes d'un incendie et le cas échéant, à faire payer les responsables. Les gens travaillant sur le projet Prometheus ont tenté de retrouver ce point d'ignition en utilisant leur méthode de marqueurs (voir chapitre 1). L'idée était de placer des marqueurs sur le front dont ils cherchent la source, puis de déplacer ces marqueurs vers l'intérieur du front en fonction des vitesses de propagation. Tous les désavantages inhérents à une méthode de marqueurs jouaient alors un grand rôle. En effet, pour un front brûlant vers l'intérieur, les trajectoires des marqueurs vont forcément se croiser et ces croisements seront très difficiles à gérer.

La méthode *Level-set* ne présente pas ces inconvénients, c'est pourquoi les créateurs de Prometheus espèrent qu'inverser le temps dans l'EDP de propagation de feu correspondant à la méthode *Level-set* nous permettrait de retrouver le point d'ignition. Le but de ce chapitre est de les convaincre que, pour plusieurs raisons, cette méthode ne pourra pas fonctionner. On inversera le temps dans l'EDP de propagation de feu pour des cas de plus en plus complexes, pour finalement se convaincre que cette méthode ne fonctionnera pas.

4.1. INVERSION DU TEMPS DANS L'EDP

On peut voir rapidement qu'en général, on ne pourra pas retrouver le point d'ignition d'un feu avec cette méthode. Considérons le cas de la propagation de deux feux circulaires qui entrent en collision (voir figure 4.1).



On voit que le résultat de cette expérience est un front qui est de plus en plus circulaire au fil du temps. On peut démontrer qu'asymptotiquement, le front obtenu sera presque exactement un cercle. Pour un temps assez grand, il sera donc impossible de déterminer que le front de flamme provient de deux foyers différents. Le même genre de phénomène se produit à chaque fois que les caractéristiques provenant d'un front se croisent. Ceci implique qu'il y a une perte d'information et que le processus est irréversible. Même s'il est clair qu'en général, on ne pourra pas retrouver la forme du front à un instant quelconque dans le passé, on peut se demander si dans des cas moins extrêmes (par exemple, quand le front étudié ne résulte pas de la collision de deux fronts distincts), le front se contracte à son point d'allumage. Les prochaines sections présentent les résultats de l'inversion du temps pour différentes vitesses de propagation.

4.1.1. Présentation de l'algorithme

On rappelle que l'EDP pour la propagation d'un feu sans vent est donnée par :

$$\partial \frac{\phi(x, y, t)}{\partial t} + \|A \nabla \phi(x, y, t)\| = 0 \quad (4.1.1)$$

où $A = \begin{pmatrix} b(x, y) & 0 \\ 0 & a(x, y) \end{pmatrix}$ avec $a(x, y)$ et $b(x, y)$ les vitesses de propagation du feu en direction y et x en un point donné.

Supposons qu'on a une condition initiale ϕ_0 qui est la fonction de distance signée à un front de flamme. Pour faire remonter le temps à ce front, il suffit de prendre $-\phi_0$ comme condition initiale à l'équation (4.1.1). Ceci revient à inverser le temps et le schéma pour l'équation originale fonctionne.

4.1.2. Un exemple de résultat pour le cas où le processus de propagation de feu est réversible

Pour cet exemple, on suppose que le front initial a la forme d'un cercle et qu'il se propage à une vitesse constante et égale à 1 dans une direction normale à lui-même. L'équation qui gouverne la propagation du feu est donc :

$$\partial \frac{\phi(x, y, t)}{\partial t} + \|\nabla \phi(x, y, t)\| = 0 \quad (4.1.2)$$

Ceci revient à choisir $a = b = 1$ partout dans le domaine.

On prend comme condition initiale la fonction de distance signée à un cercle de rayon 4 centré en (2,2) :

$$\phi^0 = \sqrt{(x-2)^2 + (y-2)^2} - 4 \Delta x \quad (4.1.3)$$

Ce cercle représente notre point d'allumage. On choisit un rayon suffisamment petit pour qu'il représente le plus possible un point, mais assez grand pour ne pas nuire à la méthode *Level-set*. L'étape suivante est de résoudre numériquement cette EDP. Les calculs sont restreints à une bande et la fonction ϕ est réinitialisée à chaque itération en utilisant l'équation de réinitialisation suivante :

$$\frac{\partial \phi}{\partial t} - \text{sign}(\phi) (\|\nabla \phi\| - 1) = 0 \quad (4.1.4)$$

Le tableau 4.1 donne les paramètres numériques utilisés.

Une fois le front final obtenu, on tente de retrouver sa source en inversant l'EDP qui gouverne sa propagation. On pose $\phi' = -\phi$ et on résoud numériquement

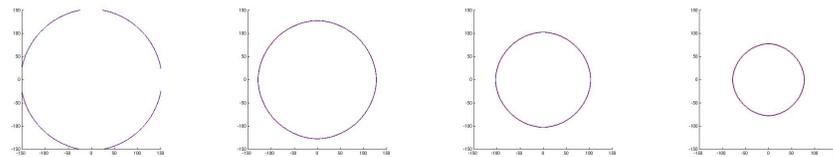
TAB. 4.1. Paramètres utilisés lors de cette expérience

Paramètres	Signification	Valeurs
x	Domaine selon l'axe x	$[-300, 300]$
y	Domaine selon l'axe y	$[-300, 300]$
Δx	Pas en espace selon l'axe x	1
Δy	Pas en espace selon l'axe y	1
Δt	Pas de temps	0.5
dw	Largeur de Bande pour l'équation d'évolution	$6 \cdot \Delta x$
dw'	Largeur de Bande pour l'équation de réinitialisation	$12 \cdot \Delta x$

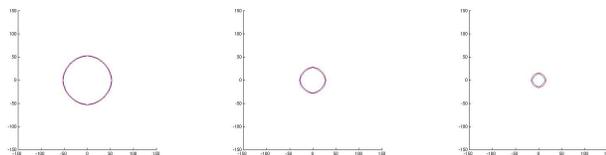
l'équation (4.1.2) pour ϕ' . On prend comme condition initiale la dernière fonction ϕ trouvée vers l'avant multipliée par -1.

Voici un exemple de résultat où le front final est obtenu avec 300 itérations. Le temps final est alors 150 (le feu se propage d'une maille par unité de temps). La séquence d'images à la figure 4.2 montre les fronts à des itérations différentes obtenus vers l'avant (en rouge) et vers l'arrière (en bleu).

FIG. 4.2. Fronts de flamme obtenus vers le futur et vers le passé à des instants donnés



(a) Itération 300 (b) Itération 250 (c) Itération 200 (d) Itération 150



(e) Itération 100 (f) Itération 50 (g) Itération 25

On voit que dans ce cas, le processus est complètement réversible.

Donnons-nous maintenant une procédure pour mesurer l'erreur de cette méthode pour retrouver le point d'allumage d'un feu, et ce pour n'importe quel front de flamme.

- (1) On résoud l'équation (4.1.1) jusqu'à la disparition totale du front.
- (2) On prend comme point d'ignition retrouvé la position moyenne des N points intérieurs (x_i, y_i) à la dernière itération où il y a encore des points intérieurs. Le centre (x_c, y_c) retrouvé est :

$$x_c = \sum_{i=1}^N \frac{x_i}{N} \quad (4.1.5)$$

$$y_c = \sum_{i=1}^N \frac{y_i}{N} \quad (4.1.6)$$

- (3) On définit ensuite l'erreur sur la position du point d'ignition trouvé comme :

$$E = \sqrt{(2 - x_c)^2 + (2 - y_c)^2} \quad (4.1.7)$$

- (4) On vérifie si cette erreur augmente en fonction du nombre d'itérations utilisé pour propager le feu vers le futur.

Appliquons maintenant cette procédure à des vitesses de propagation plus compliquées.

4.1.3. Exemple 2 - Un front en forme de marguerite

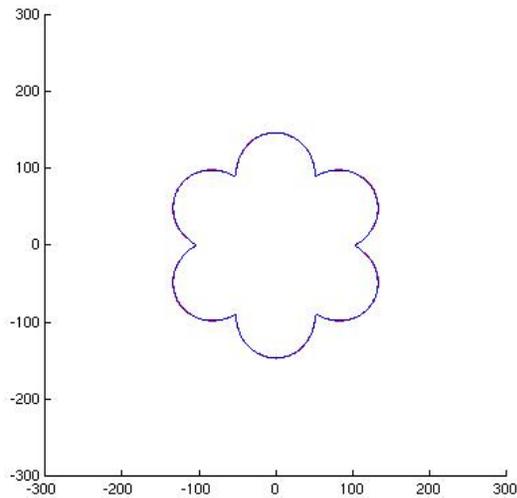
Voici un autre exemple en prenant comme vitesses de propagation :

$$a = \left| \sin \left(3 \cdot \arctan \left(\frac{y}{x + \varepsilon} \right) \right) \right| + 0.5 \quad (4.1.8)$$

$$b = \left| \sin \left(3 \cdot \arctan \left(\frac{y}{x + \varepsilon} \right) \right) \right| + 0.5 \quad (4.1.9)$$

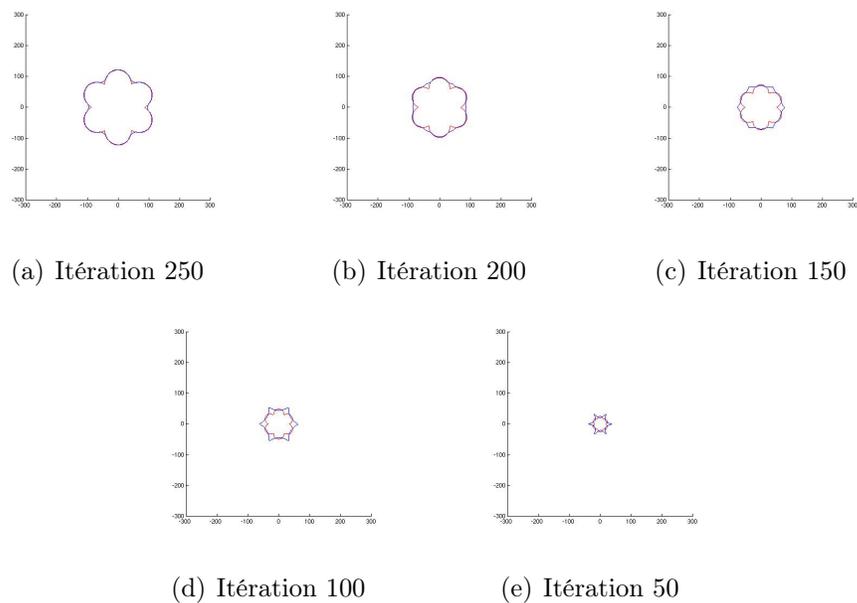
La figure 4.3 présente le front final obtenu en utilisant les mêmes paramètres que dans le cas précédent, sauf qu'on prend $\Delta t = \frac{0.5\Delta x}{\max(|a|,|b|)} = \frac{1}{3}$. Le temps final est alors 100 et pour chaque unité de temps, le front se déplace d'au plus une maille.

FIG. 4.3. Front correspondant obtenu en 300 itérations



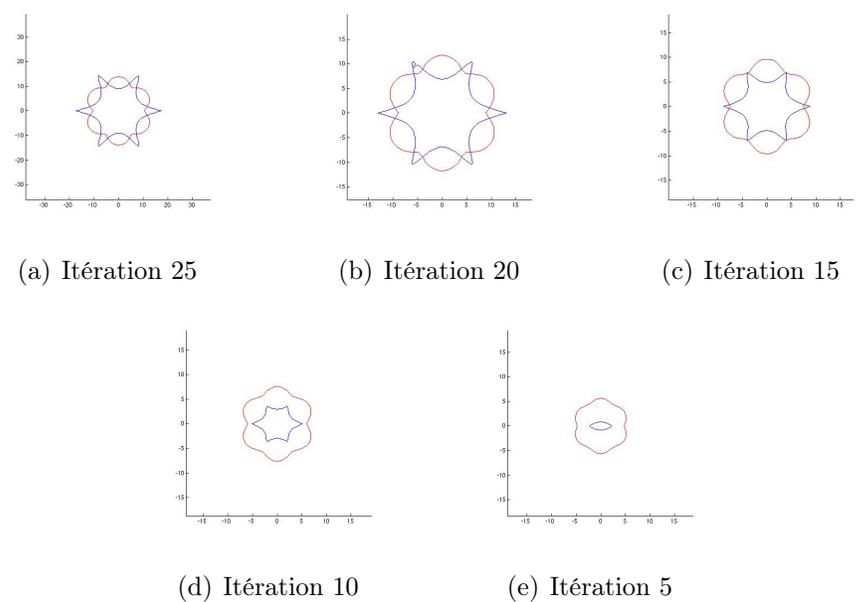
La figure 4.4 montre les fronts obtenus vers le futur (en rouge) et vers le passé (en bleu) correspondant aux itérations 250 à 50.

FIG. 4.4. Résultat de l'inversion de l'EDP pour ces vitesses de propagation



La séquence d'image à la figure 4.5 correspond aux itérations 25 à 5 par sauts de 5.

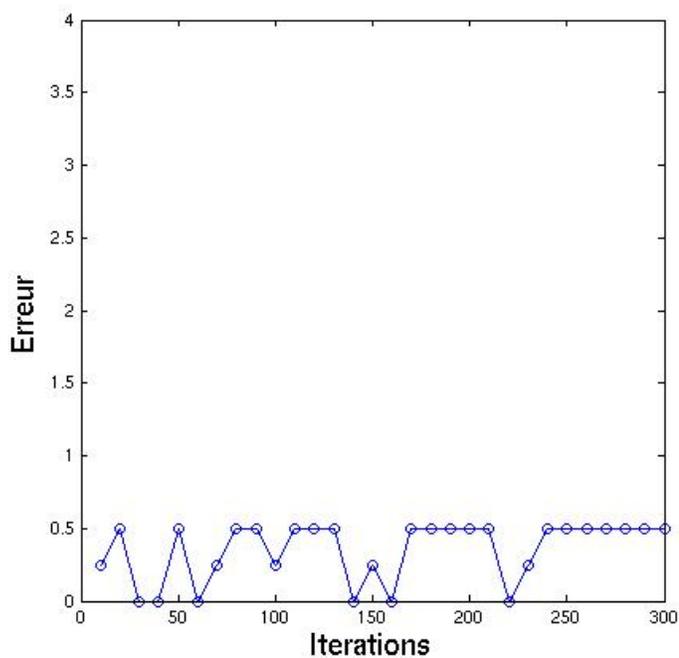
FIG. 4.5. Résultat de l'inversion de l'EDP pour ces vitesses de propagation



On voit que dans ce cas, on ne retrouve pas la véritable forme du front en remontant le temps, mais que le front se contracte autour de son point d'ignition.

La figure 4.6 trace un graphique de l'erreur sur la position du point d'ignition en fonction du nombre d'itérations.

FIG. 4.6. Erreur commise en inversant l'EDP en fonction du nombre d'itérations utilisées pour simuler la propagation du feu



On conclut que dans ce cas, inverser le temps dans l'EDP nous permet de retrouver le point d'ignition du feu puisque l'erreur est très petite par rapport au rayon du cercle correspondant au front initial (ce rayon est 4).

4.1.4. Exemple 3 - Un front asymétrique

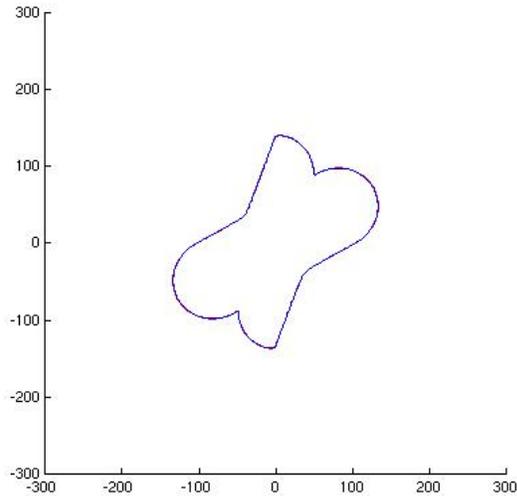
Voici un exemple avec les vitesses suivantes :

$$a = \begin{cases} |\sin(3 \cdot \arctan(\frac{y}{x+\varepsilon}))| + 0.5 & \text{si } \arctan(\frac{y}{x+\varepsilon}) > 0 \\ 0.5 & \text{sinon} \end{cases} \quad (4.1.10)$$

$$b = \begin{cases} |\sin(3 \cdot \arctan(\frac{y}{x+\varepsilon}))| + 0.5 & \text{si } \arctan(\frac{y}{x+\varepsilon}) > 0 \\ 0.5 & \text{sinon} \end{cases} \quad (4.1.11)$$

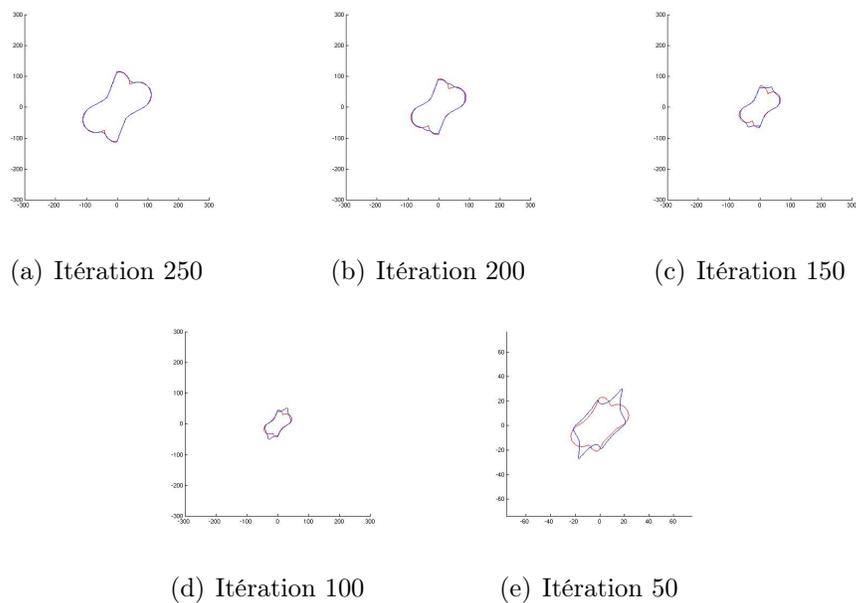
La figure 4.7 montre le front final obtenu en utilisant les mêmes paramètres que dans le cas précédent, sauf qu'on prend $\Delta t = \frac{0.5\Delta x}{\max(|a|,|b|)} = \frac{1}{3}$. Comme dans l'exemple précédent, le temps final est alors 100 et pour chaque unité de temps, le feu se déplace d'au plus une maille.

FIG. 4.7. Front correspondant obtenu en 300 itérations



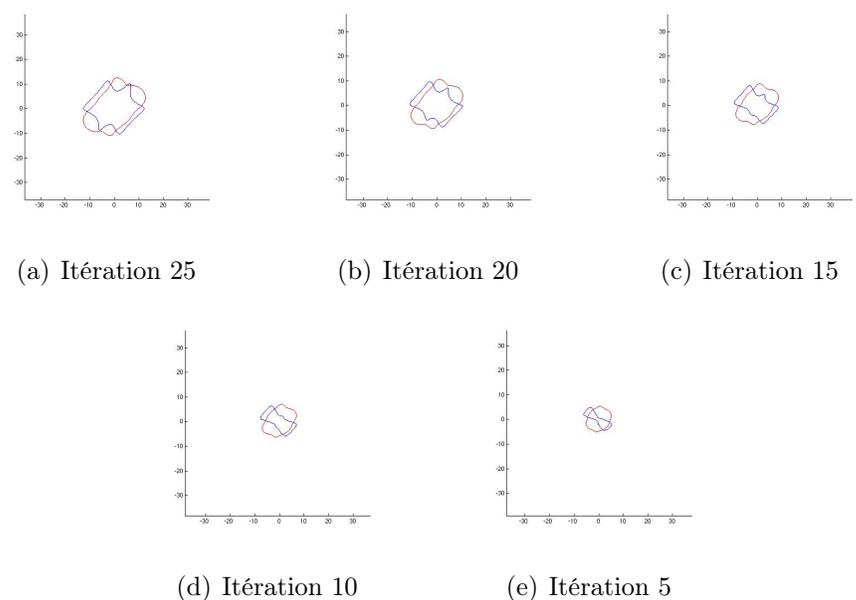
La figure 4.8 donne les fronts obtenus vers le futur (en rouge) et vers le passé (en bleu) correspondant aux itérations 250 à 50.

FIG. 4.8. Résultat de l'inversion de l'EDP pour ces vitesses de propagation



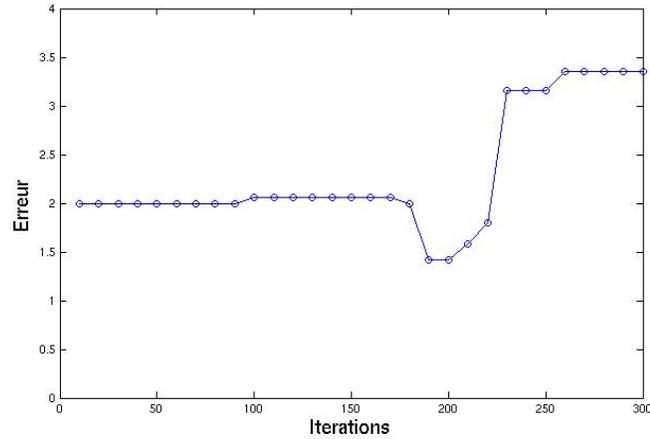
La séquence d'images de la figure 4.9 correspond aux itérations 25 à 5 par sauts de 5.

FIG. 4.9. Résultat de l'inversion de l'EDP pour ces vitesses de propagation



La figure 4.10 donne un graphique de l'erreur obtenue.

FIG. 4.10. Erreur commise en inversant l'EDP en fonction du nombre d'itérations utilisées pour simuler la propagation du feu vers le futur



Dans ce cas, le front se contracte autour de son point d'ignition car l'erreur demeure plus petite que le rayon du cercle correspondant au front initial. Cependant, l'erreur est plus grande que dans les cas précédents.

4.1.5. Exemple 4 - Un front se brisant en deux en remontant le temps

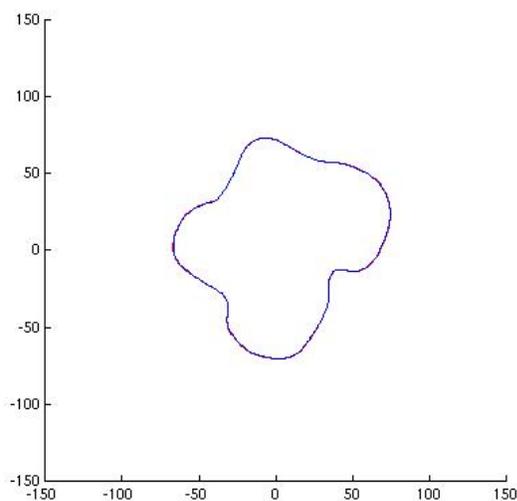
Dans les exemples précédents, le front se contracte autour du point d'ignition même si la forme du front n'est pas préservée. Cependant, ce n'est pas toujours le cas. Dans cette section, on présente un exemple de front qui se brise en plusieurs morceaux lorsqu'on essaie de remonter le temps. Prenons comme vitesses de propagation :

$$b = 2 + 1.5 \cdot \cos\left(\frac{\pi \cdot x}{30}\right) \quad (4.1.12)$$

$$a = 3 + 2 \cdot \sin\left(\frac{\pi \cdot x}{40}\right) \quad (4.1.13)$$

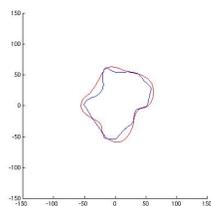
Le front final (figure 4.11) est obtenu en utilisant les mêmes paramètres que dans le cas précédent.

FIG. 4.11. Front correspondant obtenu en 300 itérations

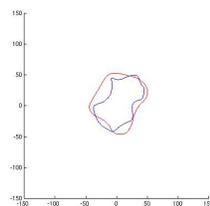


Les fronts à différentes itérations sont tracés à la figure 4.12.

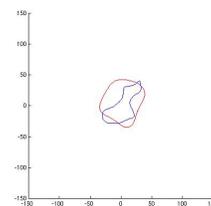
FIG. 4.12. Résultat de l'inversion de l'EDP pour ces vitesses de propagation



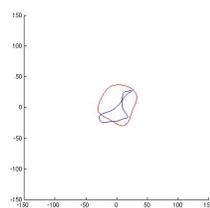
(a) Itération 250



(b) Itération 200



(c) Itération 150



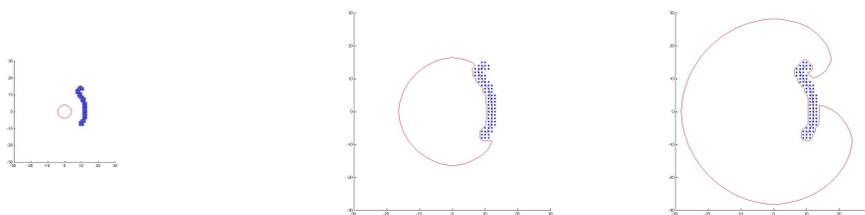
(d) Itération 125

Ici, le front s'est brisé en deux morceaux distincts. Il ne se contractera donc pas autour de son point d'ignition. Cet exemple illustre qu'il existe des cas où inverser le temps dans l'EDP ne nous permettra pas de retrouver le point d'ignition du feu. La prochaine section énonce une autre objection à cette méthode.

4.1.6. En présence d'obstacle

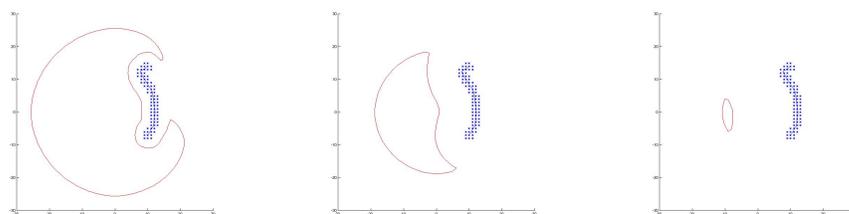
Les obstacles empêchent le feu de franchir certains points du domaine. Cependant, lorsqu'on tente d'inverser la propagation en présence d'obstacles, rien n'empêche le front de s'en décoller immédiatement. La figure 4.13 donne des images obtenues à des temps successifs du front dont on essaie de retrouver la source.

FIG. 4.13. Fronts avec obstacle obtenus vers le futur



Des images du front obtenu en inversant l'EDP sont montrées à la figure 4.1.6.

FIG. 4.14. Fronts avec obstacle obtenus en remontant le temps



On voit que le front se décolle trop tôt des obstacles et que dans de telles situations, inverser le temps dans l'EDP ne nous permet pas de retrouver le point d'ignition.

4.1.7. Inversion du temps dans l'EDP - Conclusion

On a montré qu'en général, on ne peut pas inverser le temps dans l'EDP pour retrouver le point d'ignition correspondant à un front de flamme donné. En effet, même si le problème d'inversion du temps dans l'EDP est bien posé,

la non-linéarité de l'équation de propagation rend le processus irréversible. La solution obtenue en inversant le temps ne correspond pas en général à la solution obtenue avec le temps allant vers le futur. On a aussi vu que même dans des cas en apparence simples, il est possible que le front se brise en plusieurs morceaux lorsqu'on inverse le temps et qu'alors on n'a aucune chance de retrouver le point d'ignition. Une dernière objection importante est qu'il est compliqué de traiter les obstacles lors de l'inversion du temps.

Il faudra donc trouver d'autres méthodes pour retrouver le point d'ignition d'un feu. Ce sera un des sujets étudiés lors de mon doctorat. Une façon de faire possible serait de prendre un front de flamme dont on cherche la source et placer plusieurs points d'ignition sur ce front pour faire brûler de nouveaux feux jusqu'à ce qu'il y ait rencontre des nouveaux fronts. Puisqu'un feu devrait prendre autant de temps pour brûler du point A au point B que pour faire le chemin inverse (en absence de vent), on peut espérer que le point d'ignition original se trouve au point de rencontre des nouveaux feux.

Une autre idée serait de prendre un point du domaine comme point d'ignition d'un feu et calculer le temps d'arrivée du feu partant de ce point et allant aux autres points du domaine. On pourrait ensuite utiliser le temps d'arrivée à partir de ce point pour déterminer le temps d'arrivée à partir d'un autre point en résolvant une EDP. Si on trouve une manière économique de calculer le temps d'arrivée à partir de plusieurs points dans le domaine, on pourra identifier le point d'ignition.

Chapitre 5

CONCLUSION

Le but principal de ma maîtrise était de continuer le travail déjà accompli sur la simulation de feux de forêt avec la méthode *Level-Set* pour pouvoir appliquer ce modèle à un cas réaliste. Ceci a été rendu possible grâce aux diverses améliorations présentées aux chapitres 2 et 3. La prochaine étape sera de tester le modèle avec des données réelles, et ceci constituera une partie de mon doctorat. Finalement, le dernier chapitre nous montrait qu'inverser l'EDP ne nous permettra pas de retrouver le point d'ignition et que d'autres méthodes sont à envisager.

BIBLIOGRAPHIE

- [1] OSHER S, J.A.SETHIAN, *Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys. 79 : 12-49,(1988)
- [2] J.A. SETHIAN, *Level Set Methods and Fast Marching Methods :Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press,1999
- [3] JARED BARBER,CHRISE BOSSE,ANNE BOURLIOUX, JOHN BRAUN, ERIC BRUNELLE, TANYA GARCIA, THOMAS HILLEN, CLEMENT LAM, BEN ONG, CHRISTIAN POSCHL,ROBERT BRYCE,CORDY TYMSTRA, *Prometheus - , Canada's Wildfire Growth Simulator*, Chapitre 1
- [4] NEDA ZARE NEYNEY , *A Numerical Approach for Modeling Forest Fire Spread*, Rapport de stage de maîtrise,Université de Montréal, Juillet 2008
- [5] STANLEY OSHER, RONALD FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Applied Mathematical Sciences Volume 153, Springer-Verlag New York, 2002
- [6] MARK SUSSMAN, PETER SMEREKA ET STANLEY OSHER, *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*, Journal of Computational Physics, volume 114, issue 1, p.145-159, 1994
- [7] L. HERNANDEZ ENCINAS, S. HOYA WHITE, A. MARTIN DEL REY, G. RODRIGUEZ SANCHEZ, *Modelling forest fire spread using hexagonal cellular automata*, Applied Mathematical Modelling, Volume 31, Issue 6, June 2007, Pages 1213-1227, ISSN 0307-904X
- [8] O. SERO-GUILLAUME, S. RAMEZANI, J. MARGERIT, D. CALOGINE, *On large scale forest fires propagation models*, International Journal of Thermal Sciences, Volume 47, Issue 6, June 2008, Pages 680-694, ISSN 1290-0729

- [9] V. MALLET, D.E. KEYES, F.E. FENDELL, *Modeling wildland fire propagation with level set methods*, Computers and Mathematics with Applications, Volume 57, Issue 7, April 2009, Pages 1089-1101, ISSN 0898-1221
- [10] ROUY E. AND TOURIN A., *A viscosity solutions approach to shape-from-shading*, SIAM J. Num. Anal. 29 867-884 (1992)
- [11] CHOPP,D.L., *Computing minimal surfaces via level set curvature flow*, Jour. of comp. phys., 106,pp. 77-91,1993