Université de Montréal

Méthodes rapides et efficaces pour la résolution numérique d'équations de type Hamilton-Jacobi avec application à la simulation de feux de forêt

par

Alexandre Desfossés Foucault

Département de mathématiques et de statistique
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en mathématiques

1er octobre 2015

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée

Méthodes rapides et efficaces pour la résolution numérique d'équations de type Hamilton-Jacobi avec application à la simulation de feux de forêt

présentée par

Alexandre Desfossés Foucault

a été évaluée par un jury composé des personnes suivantes :

Robert Owens
(président-rapporteur)

Anne Bourlioux
(directeur de recherche)

Jacques Bélair
(membre du jury)

Thomas Hillen
(examinateur externe)

Liliana Perez
(représentant du doyen de la FAS)

Thèse acceptée le
21 août 2015

SOMMAIRE

Cette thèse est divisée en trois chapitres. Le premier explique comment utiliser la méthode «level-set» de manière rigoureuse pour faire la simulation de feux de forêt en utilisant comme modèle physique pour la propagation le modèle de l'ellipse de Richards. Le second présente un nouveau schéma semi-implicite avec une preuve de convergence pour la solution d'une équation de type Hamilton-Jacobi anisotrope. L'avantage principal de cette méthode est qu'elle permet de réutiliser des solutions à des problèmes «proches» pour accélérer le calcul. Une autre application de ce schéma est l'homogénéisation. Le troisième chapitre montre comment utiliser les méthodes numériques des deux premiers chapitres pour étudier l'influence de variations à petites échelles dans la vitesse du vent sur la propagation d'un feu de forêt à l'aide de la théorie de l'homogénéisation.

Mots-clés : Équations aux dérivées partielles, Hamilton-Jacobi, méthode «level-set», homogénéisation, schéma semi-implicite, propagation anisotrope, feux de forêt, modèle de l'ellipse de Richards.

SUMMARY

This thesis is divided in three chapters. The first explains how to use the level-set method in a rigorous way in the context of forest fire simulation when the physical propagation model for firespread is Richards' ellipse model. The second chapter presents a new semi-implicit scheme with a proof of convergence for the numerical solution of an anisotropic Hamilton-Jacobi partial differential equation. The advantage of this scheme is it allows the use of approximative solutions as initial conditions which reduces the computation time. The third chapter shows how to use the tools introduced in the first two chapters to study the influence of small-scale variations on the wind speed on firespread using the theory of homogenization.

Keywords : Partial differential equations, Hamilton-Jacobi, Level-set method, homogenization, semi-implicit scheme, anisotropic firespread, forest fires, Richards' ellipse model.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

REMERCIEMENTS

Je suis un grand fan de trampoline, c'est donc naturel pour moi d'écrire mes remerciements en utilisant une image qui vient de ce sport. Les trampolinistes utilisent le terme technique «double-bounce» pour décrire un mouvement particulier effectué par deux sauteurs. Lors d'un double-bounce, un des deux athlètes sautille allègrement pendant que l'autre, aux aguets, attend son moment sur le rebord du trampoline. Au temps qu'il juge opportun, le deuxième s'élance dans les airs avec pour but d'atterrir en même temps que le sautilleur. Si le timing est bon, le sautilleur est propulsé dans les airs à une hauteur vertigineuse.

Je voudrais donc profiter de l'occasion pour remercier tous ceux qui m'ont double-bouncé sur le trampoline de la vie et qui m'ont permis de monter à ma hauteur présente. Un grand merci à ma directrice Anne Bourlioux pour avoir accepté de me prendre comme étudiant à la maîtrise et au doctorat, puis pour toutes les heures de discussion depuis 2008. Merci à ma femme Josyanne Sylvestre, qui a accepté d'unir son élan au mien. Merci à mes parents Jocelyne Desfossés et Daniel Foucault. Vous êtes le sol sur lequel mon trampoline repose. Merci aussi à tous mes amis à l'intérieur du DMS et à l'extérieur pour le support moral et un nombre incalculable d'heures de plaisir. Je ne nomme pas de noms de peur d'oublier quelqu'un, mais vous savez qui vous êtes ! Merci à mes deux chats Babou et Gizmo pour une bonne dose de ronronthérapie quotidienne. Miaw prrrRRrrourr prrrr Miaourrrr.

Encore une fois merci à tous ! J'espère que nos bonds futurs se synchroniseront pour que nous puissions mutuellement nous faire sauter de plus en plus haut.

Bonne lecture !

Alexandre

Chapitre 1

INTRODUCTION

Le but de cette thèse est de développer de nouvelles méthodes numériques pour la résolution d'équations de type Hamilton-Jacobi et de montrer leur utilité dans le contexte de la simulation de feux de forêt.

Chaque année, des dizaines de feux de forêt ravagent le territoire canadien. Il serait utile de pouvoir prédire les régions brûlées par des incendies forestiers. Être capable de prévoir l'étendue du désastre peut permettre aux autorités de bien planifier leurs interventions. Le SRD (Sustainable Resource Development, branche du gouvernement de l'Alberta) a développé un logiciel appelé "Prometheus" dans ce but. Ce logiciel tient compte de tous les paramètres pertinents (météo, vent, topographie, etc...) et utilise un modèle physique adéquat pour obtenir la vitesse de propagation d'un feu à partir de ces paramètres. Cette vitesse est ensuite utilisée dans une méthode numérique de marqueurs pour calculer la position du feu. Ce processus donne une bonne estimation de la position du feu à chaque instant. Cependant, les méthodes de marqueurs présentent quelques inconvénients. La recherche d'une meilleure formulation numérique pour calculer le déplacement du front est à l'origine de ce projet.

Le but principal du premier article de cette thèse est de présenter une manière mathématiquement rigoureuse d'utiliser la méthode «level-set» (appelée méthode iso-niveaux en français, mais ce terme est d'usage peu courant) dans le contexte de la simulation de feux de forêt quand le modèle physique pour la propagation est le modèle de l'ellipse de Richards. Ce projet est né lors d'un atelier de résolution de problèmes industriels MITACS. Lors de cet atelier, les participants ont obtenu une équation aux dérivées partielles (EDP) décrivant la propagation d'un feu de forêt avec la méthode level-set (Barber et al., 2007)).

Après l'atelier, le travail a été poursuivi par Neyney (2008). Le résultat principal de Neyney (2008) est une méthode numérique pour la résolution de l'équation

décrivant la propagation d'un feu. L'algorithme donné dans Neyney (2008) fonctionne pour des cas de propagation à paramètres spatialement homogènes. Mais, le schéma numérique utilisé pour résoudre l'équation est donné sans preuve de convergence.

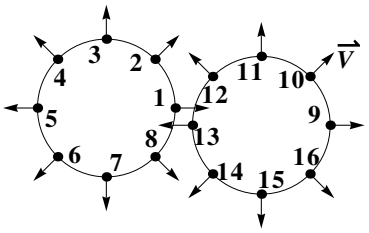
Ensuite, Foucault (2008) incorpore aux résultats de Neyney (2008) des outils numériques permettant de diminuer le temps de calcul, de traiter des paramètres hétérogènes et de traiter des obstacles dans le domaine.

Le principal point manquant après ces travaux était une preuve de convergence pour le schéma numérique. Le premier article de cette thèse porte sur ce sujet. La contribution principale est un schéma numérique pour la résolution de l'équation de propagation d'un feu. Ce schéma, basé sur la méthode de Godunov, converge vers la solution théorique de l'équation. Le public cible est la communauté scientifique intéressée par la simulation de feux de forêt. Il était donc nécessaire d'expliquer les bases mathématiques de la méthode «level-set». L'article contient aussi tous les éléments nécessaires pour comprendre comment utiliser cette méthode efficacement dans le contexte de la simulation de feux de forêt. Pour bien montrer l'applicabilité de la méthode «level-set» au problème, le dernier exemple de cet article montre comment utiliser cette méthode dans un cas réaliste avec des données qui viennent de Prometheus.

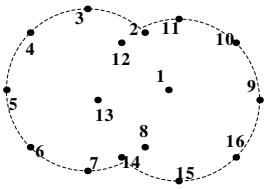
Les prochaines sous-sections de cette introduction expliquent le lien entre le problème de la simulation de la propagation d'un feu de forêt et les équations aux dérivées partielles de type Hamilton-Jacobi. On verra les bases nécessaires pour comprendre la résolution numérique d'équations de ce type. Puis, les deux autres projets de cette thèse seront expliqués avec la théorie nécessaire à leur compréhension.

1.1. Méthode de marqueurs et méthode «level-set»

Cette section explique comment fonctionne la méthode des marqueurs et quels sont ses inconvénients. Cette méthode déplace un front de flammes en plaçant plusieurs marqueurs sur ce front, puis en les bougeant en fonction des vitesses de propagation du feu. Cette technique rencontre des difficultés lorsque, par exemple, deux fronts de flammes entrent en collision (voir figure 1.1). Dans un tel cas, les



(a) Position initiale du front



(b) Position à l'instant suivant

Figure 1.1. Illustration des problèmes reliés à la méthode des marqueurs

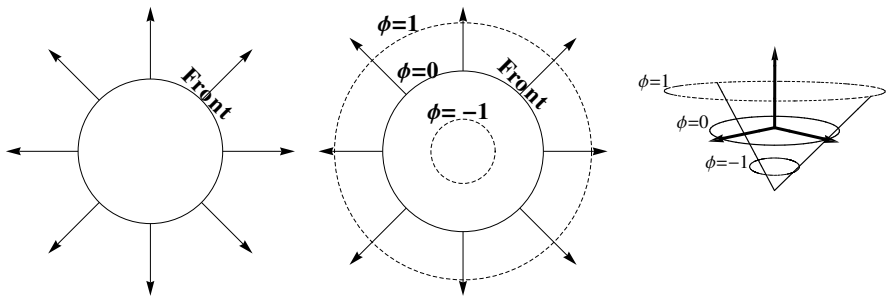
trajectoires des marqueurs peuvent se croiser et il devient difficile de reconstruire le front. Ce genre de problème motive l'utilisation de la méthode «level-set» pour calculer l'évolution des fronts de flamme.

1.2. Introduction à la méthode «level-set»

La méthode «level-set» est une technique très utilisée pour faire le suivi d'interfaces. Elle est due à Osher and Sethian (1988). Elle consiste à représenter une courbe de manière implicite comme la courbe de niveau zéro d'une surface, puis à obtenir le déplacement de cette courbe en déformant la surface à l'aide d'une équation aux dérivées partielles appropriée.

1.3. Représentation implicite d'une courbe

Il existe plusieurs manières de représenter implicitement une courbe. Par exemple, la figure 1.2 montre que le cercle $x^2 + y^2 = 1$ peut être vu comme la courbe de niveau 0 d'un cône.



(a) (b) (c)

FIGURE 1.2. Différentes représentations du front de flamme. Le front de flamme est vu comme une courbe en (a). En (b), il est montré comme la courbe de niveau 0 d'une surface ϕ. En (c), la surface ϕ est tracée.

Pour plusieurs raisons, un choix commun est de représenter implicitement une courbe par la fonction de distance signée (Delfour and Zolesio, 2011) à cette courbe. Ce choix assure une certaine régularité, ce qui est utile pour approximer les dérivées spatiales.

Pour obtenir une équation aux dérivées partielles décrivant le déformation de cette représentation implicite de la courbe, on considère un point sur cette courbe ayant une trajectoire $\vec{x}(t)$. Ce point devra demeurer sur le niveau 0 de la surface :

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

On dérive des deux côtés par rapport à t à l'aide de la dérivée en chaîne pour obtenir

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

Cet argument peut être étendu aux autres courbes de niveau de la surface ϕ pour obtenir l'équation «level-set»

$$\phi_t + \vec{v} \cdot \nabla\phi = 0.$$

Le modèle physique nous permettra de définir la vitesse de propagation $\vec{v}$ sur le front de flammes. La vitesse en tout autre point du domaine est définie en assumant la présence d'un front en ce point.

1.4. ÉQUATION «LEVEL-SET» POUR LE CAS DE LA PROPAGATION ISOTROPE

L'exemple le plus classique est le cas où la vitesse de propagation est constante, égale à F et orientée en direction de la normale au front. Alors, la vitesse $\vec{v}$ est

donnée par

$$\vec{v} = \frac{F\nabla\phi}{\|\nabla\phi\|}.$$

On peut obtenir cette vitesse en supposant que le vecteur normal au front ne change pas d'un instant à l'autre. Puisque la vitesse F ne dépend pas de la normale au front, on dit que la propagation est isotrope. Si F dépend de la normale au front $\frac{\nabla\phi}{\|\nabla\phi\|}$, on dit que la propagation est anisotrope. Ce sera le cas pour l'équation de propagation d'un feu de forêt.

1.5. Solutions numériques d'équations aux dérivées partielles de type Hamilton-Jacobi

1.5.1. Solutions de viscosité

Les équations introduites ci-dessus sont des équations aux dérivées partielles de type Hamilton-Jacobi instationnaires, qu'on peut écrire sous une forme générale :

$$\phi_t + H(\nabla\phi, x) = 0.$$

On considérera aussi plus tard la forme stationnaire :

$$H(\nabla T, x) = 1.$$

En général, les solutions cherchées ne sont pas des solutions au sens classique mais plutôt des «solutions de viscosité». Cette classe de solutions a été définie pour ces équations dans Crandall and Lions (1983) pour tenir compte du fait que les solutions cherchées ne sont pas nécessairement dérivables en tout point du domaine. Par exemple, la solution au problème stationnaire (1.5.1), qui représente le temps d'arrivée T à un point du domaine à partir de l'origine

$$\begin{cases} \|\nabla T\| &= 1 \\ T(0) &= 0 \end{cases} \tag{1.5.1}$$

est un cône centré à l'origine. Donc, la solution n'est pas dérivable à l'origine. C'est pourquoi il faut définir une classe de solutions qui admet ces singularités. Selon Sethian (1999), une fonction u est une solution de viscosité de l'équation $\phi_t + H(\nabla\phi, x) = 0$ si pour toute fonction test lisse v les conditions suivantes sont vérifiées :

(1) Si $u - v$ a un maximum local à un point (x_0, t_0) alors

$$v_t(x_0, t_0) + H(Dv(x_0, t_0), x_0) \leq 0.$$

(2) Si $u - v$ a un minimum local en un point (x_0, t_0) alors

$$v_t(x_0, t_0) + H(Dv(x_0, t_0), x_0) \geq 0.$$

Sethian (1999) résume quelques faits à propos de ces solutions :

(1) Si u est une solution lisse des équations d'Hamilton-Jacobi, alors c'est une solution de viscosité.

(2) Si une solution de viscosité u est différentiable en un point, alors elle satisfait l'équation à ce point.

(3) La solution de viscosité est unique.

(4) Cette solution de viscosité est la même que celle obtenue en considérant la limite quand ϵ tend vers 0 de l'équation $u_t + H(\nabla\phi, x) = \epsilon\Delta u$.

Notons aussi que la solution de viscosité à une équation d'Hamilton-Jacobi correspond à la solution physique du problème considéré. Par exemple, la solution de viscosité au problème (1.5.1) correspond bien au temps d'arrivée.

1.5.2. Conditions pour avoir convergence vers la solution de viscosité

La convergence de solutions numériques vers la solution de viscosité pour des équations aux dérivées partielles de type Hamilton-Jacobi a été étudiée dans les années 80 et 90 par plusieurs personnes (voir par exemple Crandall and Lions (1984) et Barles and Souganidis (1991)). Les conditions pour avoir un schéma convergent sont :

(1) Monotonicité ;

(2) Stabilité ;

(3) Consistance.

Des critères de monotonicité sont donnés dans Oberman (2006). Par exemple, un schéma explicite de la forme

$$u_i^{n+1} = u_i^n + H(u_{i-1}^n, u_i^n, u_{i+1}^n)$$

est monotone si H est une fonction non-décroissante de ses arguments.

1.5.3. Schémas de Godunov

Une manière de définir un schéma numérique monotone pour ces équations est d'utiliser le schéma de Godunov pour l'hamiltonien (Osher and Fedkiw, 2002). Pour le définir, on doit d'abord introduire

(1) ϕ_x^+, ϕ_x^- les différences avant et arrière par rapport à une variable x. On définit ϕ_x^+ ainsi :

$$\phi_x^+ = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}$$

où $\phi_{i,j}$ correspond à la valeur de la fonction ϕ au point (x_i, y_j). ϕ_x^- est définie avec la formule suivante :

$$\phi_x^- = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}$$

(2) l'intervalle I^x définit à chaque point en utilisant les valeurs des approximations des dérivées au point considéré

$$I^x := [min(\phi_x^-, \phi_x^+), max(\phi_x^-, \phi_x^+)]$$

L'approximation numérique de H est alors

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

où $ext_x H(\phi_x)$ est défini ainsi

(1) Si $\phi_x^- < \phi_x^+$, alors $ext_x H$ est obtenu en prenant la valeur de $\phi_x \in I^x$ qui minimise H

(2) Si $\phi_x^- > \phi_x^+$, alors $ext_x H$ est obtenu en prenant la valeur de $\phi_x \in I^x$ qui maximise H

(3) Si $\phi_x^- = \phi_x^+$, alors $ext_x H$ est obtenu en remplaçant dans H la valeur qui approxime ϕ_x

On définit similairement $ext_y H$.
Par exemple, pour l'équation

$$\phi_t + \|\nabla\phi\| = 0$$

la formule (1.5.2) due à Rouy and Tourin (1992) présente de manière concise l'hamiltonien de Godunov correspondant

$$\phi_x^2 = \max\left(\max\left(\phi_x^-, 0\right)^2, \min\left(\phi_x^+, 0\right)^2\right). \tag{1.5.2}$$

Pour le cas anisotrope, il est plus difficile d'écrire l'hamiltonien de Godunov. On aura besoin de formuler le problème d'une autre manière.

1.5.4. Formule de Bellman

Cette section introduit une autre manière d'écrire l'hamiltonien relié à un problème de propagation (Oberman et al., 2009; Kao et al., 2003; Vladimirsky,

2001). Cette nouvelle formulation nous permettra d'écrire un hamiltonien de Godunov pour l'équation considérée dans cette thèse.

On veut considérer des hamiltoniens de forme

$$H\left(\frac{\nabla T(x)}{\|\nabla T(x)\|}, x\right) = F\left(x, \frac{\nabla T(x)}{\|\nabla T(x)\|}\right)\|\nabla T\|.$$

Utiliser de tels hamiltoniens dans l'équation «level-set» signifie qu'on veut calculer la propagation d'une interface se déplaçant à vitesse normale F qui dépend de l'orientation de l'interface. On peut réécrire cet hamiltonien en utilisant la formule de Bellman

$$H(\nabla T, x) = \max_{\|\alpha\|=1} \left((\nabla T \cdot \alpha)\, w(x, \alpha)\right).$$

où $w(x, \alpha)$ représente la vitesse d'une particule sur le front au point x en direction α. On peut obtenir cette formule en considérant la vitesse normale au front d'un point de vue lagrangien. Cette vitesse peut être obtenue en considérant que le front est composé d'une infinité de particules avec une vitesse de grandeur w et que ces particules ont pour but de faire avancer le front en direction normale à lui-même le plus rapidement possible. La vitesse F peut donc s'écrire comme le maximum de la projection de la vitesse des particules sur le vecteur normal au front

$$F\left(x, \frac{\nabla T(x)}{\|\nabla T(x)\|}\right) = \max_{\|\alpha\|=1} \left(\vec{n} \cdot \alpha w(x, \alpha)\right).$$

En multipliant des deux côtés par $\|\nabla T\|$ et en remplaçant le vecteur normal $\vec{n}$ par $\frac{\nabla T}{\|\nabla T\|}$ on obtient la formule suivante pour l'hamiltonien

$$F\left(x, \frac{\nabla T(x)}{\|\nabla T(x)\|}\right)\|\nabla T\| = \max_{\theta\in[0,2\pi]} \left((T_x, T_y) \cdot (\cos\theta, \sin\theta) w(x, \theta)\right).$$

En utilisant cette formulation, il sera plus facile d'écrire l'hamiltonien de Godunov pour discrétiser le problème. Le premier article explique quelle est l'équation «level-set» correspondant à la propagation d'un feu de forêt et comment la résoudre numériquement à l'aide de la formule décrite dans cette section.

1.6. Projet 2

Les sections précédentes expliquent les bases nécessaires à la compréhension du premier projet de cette thèse. Après avoir terminé ce projet, tous les outils nécessaires pour faire la simulation d'un feu de forêt avec la méthode «level-set» étaient développés. Cependant, les développeurs de Prometheus cherchaient les réponses à d'autres questions. Une des tâches que les pompiers forestiers ont à

effectuer est de déterminer la source d'un incendie. Retrouver le point d'ignition peut permettre de blâmer un responsable pour l'incendie et donc de le faire payer pour les dommages. Avoir un outil mathématique permettant de retrouver ce point d'ignition serait très utile. Un autre enjeu majeur est de pouvoir traiter l'incertitude sur les données. En effet, il est difficile d'obtenir des valeurs précises pour les différents paramètres impliqués. Il serait souhaitable de pouvoir calculer une sorte de "carte de probabilité de combustion" qui pourrait associer à chaque point du domaine une probabilité qu'il soit atteint par le feu qui dépendrait de l'incertitude sur les données.

Ces questions pratiques ont motivé le développement d'un outil mathématique présenté dans le deuxième article de cette thèse. Une manière de répondre aux questions soulevées au paragraphe précédent est d'effectuer plusieurs calculs de solution au problème de propagation. Par exemple, on peut obtenir une carte de probabilité en faisant des centaines de simulation en variant les valeurs de paramètres et en comptant le nombre de fois qu'un point est atteint par le feu. Pour retrouver le point d'ignition d'un feu, on peut simuler l'évolution du feu à partir de différents points d'ignition et voir ainsi quel point d'ignition donne un front de flamme qui correspond le plus à ce qui a été observé. Ces approches fonctionnent, mais sont coûteuses en temps de calcul si on utilise la méthode explicite développée dans le premier article de cette thèse. Le schéma semi-implicite présenté dans le deuxième article diminue le temps de calcul nécessaire à l'utilisation de ces idées. Il permet d'utiliser comme condition initiale au problème de propagation une solution à un problème proche. Dans le cas de la recherche du point d'ignition, une condition initiale pertinente serait la solution au problème de propagation obtenue à partir d'un point du domaine voisin. Dans le cas de la carte de probabilité, une bonne condition initiale serait une solution obtenue avec des vitesses de propagation proches.

Cette méthode numérique est aussi rapide en absence de bonne condition initiale et elle s'applique à des cas plus généraux. Dans l'article, on donne le détail pour une équation de type Hamilton-Jacobi quelconque avant de se concentrer sur le cas du modèle de propagation d'un feu.

Le point de départ au deuxième projet est la formulation du problème en terme de temps d'arrivée à laquelle on rajoute une dépendance en un pseudo-temps t :

$$T_t + H(\nabla T) = 1.$$

La solution à l'équilibre est le temps d'arrivée à n'importe quel point du domaine à partir de points d'ignitions prédéterminés. Afin de pouvoir montrer les différences entre la nouvelle approche suggérée ici et les algorithmes existants, les prochaines

sections expliquent le lien entre la formulation «level-set» et la formulation en terme de temps d'arrivée. Elles présentent aussi d'autres algorithmes de résolution du problème du calcul du temps d'arrivée.

1.6.1. Reformulation du problème de propagation d'un front en un problème de temps d'arrivée

On considère dans le premier article un problème de forme

$$\begin{cases} \phi_t + H(\nabla\phi) = 0 \\ \phi(x, y, 0) = \phi_0 \end{cases}$$

qui représente la propagation d'une interface décrite par ϕ. Une autre manière d'écrire le problème est (Sethian, 1999) :

$$\begin{cases} H(\nabla T) = 1 \\ T = 0 \text{ sur } \Gamma \end{cases} \tag{1.6.1}$$

où Γ représente la position initiale du front.

1.6.2. Fast-Marching et ordered upwind

Une des premières méthodes inventées pour résoudre (1.6.1) est la méthode «fast-marching» (Sethian, 1996; Tsitsiklis, 1995). Elle s'applique à des cas de propagation isotrope. L'idée à la base de cette méthode est que l'information doit se propager en partant de points où T est petit vers des points où T est plus grand. Considérons le cas de propagation en deux dimensions à vitesse normale $F = 1$. Supposons qu'on veut calculer le temps d'arrivée à partir d'un point d'ignition fixé x_0. L'algorithme fonctionne de la manière suivante :

(1) Définir K l'ensemble des points où la valeur de T est connue.

(2) Tant qu'il y a des points qui ne sont pas dans K, faire :

   (a) Définir N l'ensemble des points voisins immédiats des points dans K.

   (b) Calculer des valeurs potentielles de T pour les points dans N avec un schéma de Godunov.

   (c) Fixer la valeur du point dans N où la nouvelle valeur de T calculée est la plus petite. Rajouter ce point dans K.

Cette méthode fonctionne pour des cas isotropes seulement. Une manière de voir pourquoi ceci est vrai est de considérer la direction de propagation de l'information dans le problème. Les caractéristiques de l'équation isotrope sont alignées avec le gradient de T. Ceci explique pourquoi il est correct de fixer en premier la valeur de T au point voisin d'un point où la valeur est connue seulement au

point voisin ayant la plus petite valeur de T. Pour pouvoir considérer des cas anisotropes, la méthode «ordered-upwind» a été mise au point. Les différences principales entre la méthode «ordered-upwind» (Sethian and Vladimirsky, 2003) et la méthode de «fast-marching» sont que l'ordre de mise à jour des valeurs du temps d'arrivée reflète la direction des caractéristiques de l'équation et que la formule de mise à jour est différente.

1.6.3. Fast-sweeping

La méthode de fast-sweeping (Zhao, 2004) est une méthode itérative qui combine un schéma de Godunov pour la discrétisation des dérivées spatiales avec une manière particulière d'ordonner les points du domaine pour mettre à jour les valeurs du temps d'arrivée. Supposons qu'on veut résoudre le problème suivant à deux dimensions spatiales :

$$\begin{cases} \|\nabla T\| = 1 \\ T(0,0) = 1. \end{cases}$$

L'algorithme commence par initialiser T en chaque point du domaine avec des valeurs très grandes. L'idée est d'ensuite balayer successivement le domaine selon les 4 orientations possibles : balayages nord-est, nord-ouest, sud-est et sud-ouest. Par exemple, on fait un balayage nord-est en partant du point au coin inférieur gauche du domaine, en remontant le long de la première colonne, puis en passant à la deuxième colonne et ainsi de suite. Lors des balayages, on tente de mettre à jour la valeur en tous les points en fonctions des valeurs aux points voisins en utilisant un schéma de Godunov. Si on réussit à calculer une nouvelle valeur en un point (x, y) qui est plus petite que la valeur déja mise, on met à jour $T(x, y)$.

Il y a une extension de cette méthode au cas anisotrope (Kao et al., 2003). En fait, cet article est la référence principale utilisée pour écrire le premier article de cette thèse. Les auteurs utilisent un schéma de Godunov pour des cas de propagation anisotrope avec la méthode de «fast-sweeping». Le travail principal du premier article de cette thèse est d'adapter ce schéma à l'équation «level-set».

1.6.4. Différences entre ces algorithmes et le schéma semi-implicite

Pour les trois méthodes présentées ci-dessus («fast-marching», «fast-sweeping» et «ordered-upwind»), avoir une bonne estimation de la valeur de la solution ne permet pas d'accélérer les calculs. Dans le cas des méthodes de «fast-marching» et «ordered-upwind», la valeur pour le temps d'arrivée est calculée en propageant la valeur à partir d'un point où la valeur est fixée d'avance. On n'utilise pas de valeur initiale pour les autres points du domaine. Dans le cas de la méthode

de «fast-sweeping», on doit fixer une valeur initiale très grande en tout point du domaine sauf aux points où la valeur initiale est connue. Puis, cette valeur initiale ne fait que descendre. Fixer une valeur initiale qui sous-estime la vraie valeur du temps d'arriver ferait échouer l'algorithme. L'avantage d'utiliser un schéma semi-implicite est qu'un tel schéma nous permettra d'utiliser une solution approximative à un problème donné pour accélérer les calculs.

1.7. Projet 3-Application de la théorie de l'homogénéisation périodique au problème de la propagation de feux de forêt

L'homogénéisation est une théorie mathématique qui permet de traiter des phénomènes ayant un caractère multi-échelle. Un feu de forêt est un bon exemple d'un tel phénomène. Un des paramètres qui a une grande influence sur la propagation des feux de forêt est le vent. Il peut varier à des échelles très différentes. La turbulence et des effets de rétroaction entre le feu et le vent entraînent une variation à petite échelle, tandis que le climat et la météo entraînent des variations à plus grande échelle. Ces variations à chaque échelle influencent la position du feu, mais faire des simulations en représentant bien dans l'ordinateur ce qui se passe à chaque échelle est trop coûteux en temps de calcul. Il est souhaitable de pouvoir calculer des vitesses de propagation effectives en fonction des variations à petite échelle, sans calculer le détail de la solution. Ce genre de moyennage des effets pour obtenir des valeurs effectives pour les coefficients pertinents est étudié pour plusieurs types d'équations. Le but du troisième article de cette thèse est d'appliquer la méthode numérique développée au deuxième article à l'homogénéisation de l'équation de propagation d'un feu. On cherche à répondre à la question : mathématiquement, quel risque courons-nous à ne pas tenir compte des variations à petite échelle dans la vitesse du vent lors de nos simulations ? On verra que négliger ces effets a tendance à diminuer la vitesse de propagation du feu. Les prochaines sous-sections donnent une introduction à l'homogénéisation d'équations de type Hamilton-Jacobi.

1.7.1. Analyse asymptotique d'une équation d'Hamilton-Jacobi

Un outil couramment utilisé en homogénéisation est l'analyse asymptotique (Xin, 1991). Considérons le problème suivant :

$$\begin{cases} H\left(\nabla T^\epsilon(x), \frac{x}{\epsilon}\right) & = 1 \\ T^\epsilon(0) & = 0. \end{cases}$$

La dépendance en $\frac{x}{\epsilon}$ indique que la solution varie à petite échelle. Supposons que la solution peut se décomposer en une partie variant à grande échelle, une partie variant à échelle ϵ, puis une variant à échelle ϵ^2 et ainsi de suite. Supposons donc que la solution peut s'écrire ainsi :

$$T^\epsilon(x) = T^0(x, y) + \epsilon T^1(x, y) + O(\epsilon^2)$$

pour $y = \frac{x}{\epsilon}$. En remplaçant cette forme pour la solution et en regroupant ensemble les termes d'ordre $O(1)$, on trouve le problème cellulaire pour cette équation

$$H(\nabla_x T^0 + \nabla_y T^1, y) = 1.$$

On considère ce problème sur une cellule en la variable y, donc on peut traiter la partie $\nabla_x T^0$ qui varie à grande échelle comme une constante p. Posons $v = \nabla_y T^1$. Si pour tout p possible on trouve v satisfaisant cette équation, on aura trouvé l'hamiltonien effectif $\bar{H}$:

$$\bar{H}(p) = H(p + \nabla v, x).$$

La méthode numérique utilisée dans le troisième article fonctionnera de la manière suivante :

(1) On fixe une valeur de p, ce qui revient à fixer une orientation pour le front.
(2) On fait évoluer ce front à orientation à grande échelle p à travers une boîte périodique correspondant à la variation à petite échelle pour voir à quelle vitesse le front se propage.
(3) On répète la procédure pour d'autres valeurs de p. On pourra ensuite avoir $\bar{H}(p)$ pour n'importe quel p en procédant par interpolation.

Les prochaines sections donnent des conditions nécessaires à l'existence de cet hamiltonien effectif et expliquent comment en calculer.

1.7.2. Existence d'hamiltoniens effectifs et convergence - cas périodique

L'homogénéisation d'équations d'Hamilton-Jacobi périodique a été premièrement étudiée dans Lions et al. (1988). Xin (1991) rappelle leur résultat. Considérons

$$\begin{cases} \phi^\epsilon_t + H(\nabla \phi^\epsilon, \frac{x}{\epsilon}) & = 0 \text{ pour } x \text{ dans } \mathbb{R}^n \times (0, +\infty) \\ \phi^\epsilon(x, 0) & = \phi_0. \end{cases}$$

avec H de période 1 en la deuxième variable. Sous les conditions

(1) H est localement Lipshitz en toutes les variables

(2) $H(p,x) \to \infty$ quand $\|p\| \to \infty$ uniformément pour $x \in \mathbb{R}^n$

(3) ϕ_0 est bornée et uniformément continue

(4) $\nabla\phi_0 \in L^\infty((R))$

La solution ϕ^ϵ converge uniformément vers la solution de viscosité de l'équation

$$\phi_t + \bar{H}(\nabla\phi, x) = 0$$

où l'hamiltonien effectif $\bar{H}$ est obtenu en résolvant un problème cellulaire, tel qu'expliqué à la section précédente

1.7.3. Méthodes de calcul d'hamiltoniens effectifs

Les deux sections précédentes donnent des résultats d'existence de solution pour les problèmes homogénéisés. Cette section introduit des méthodes de calcul d'hamiltoniens effectifs pour le cas des perturbations périodiques. L'article de Luo et al. (2011) rappelle différentes méthodes de calcul. Deux méthodes sont dues à Qian (2003). La «small delta method» et la «large T method». L'idée de base de la «small delta method» est de construire un problème cellulaire approximatif :

$$\delta v^\delta(y) + H(p + \nabla v^\delta(y), y) = 0. \tag{1.7.1}$$

Sous les conditions énoncées dans Qian (2003), δv^δ approxime $-\bar{H(p)}$ si δ est assez petit. On peut résoudre le problème (1.7.1) en fixant δ petit et en discrétisant l'hamiltonien avec n'importe quel schéma adéquat.

La «large T method» est basée sur un problème d'évolution construit à partir du problème cellulaire

$$\begin{cases} u_t + H(p + Du, y) & = 0 \\ u_0 & = g. \end{cases}$$

Définissons un opérateur $G_t(g)(y)$ qui associe à la condition initiale g la solution $u(y,t)$ avec G_t qui dépend de p. Qian (2003) montre que l'hamiltonien effectif est donné par

$$\bar{H}(p) = -\lim_{t\to\infty} \frac{G_t g}{t}.$$

Numériquement, on obtient l'hamiltonien effectif en résolvant ce problème d'évolution avec un schéma approprié jusqu'à un temps suffisamment élevé.

Les deux méthodes de calcul d'hamiltonien effectif précédentes exigent une résolution de problème cellulaire pour toute valeur de p possible. Une autre approche

suggérée par Oberman et al. (2009) nous permet de retrouver l'hamiltonien effectif en solvant un seul problème à très haute résolution, plutôt qu'en résolvant le problème cellulaire pour toute valeur de p. Ils considèrent le problème suivant

$$\begin{cases} H(\nabla T^\epsilon(x), \frac{x}{\epsilon}) & = 1 \\ T^\epsilon(0) = 0 \end{cases}$$

pour des hamiltoniens satisfaisant les conditions suivantes :

(1) H est convexe

(2) $H(tp, x) = tH(p, x)$ pour tout $t > 0$

(3) $c_2\|p\| \leq H(p, x) \leq C_2\|p\|$

(4) H périodique en x.

Leur résultat est que si on écrit l'hamiltonien effectif en utilisant la formule de Bellman

$$\bar{H}(p) = \max_{\|\alpha\|=1} p \cdot \alpha \bar{c}(\alpha),$$

alors la vitesse homogénéisée d'une particule sur le front $\bar{c}(\alpha)$ peut être approximée par la solution du problème à très haute résolution T^ϵ par la formule suivante

$$\frac{1}{\bar{c}(q)} = \frac{T^\epsilon(q)}{\|q\|} + O(\epsilon).$$

Les deuxièmes et troisièmes articles porteront en partie sur une application de la «large-T method» au cas correspondant à la propagation d'un feu. Vu que cette méthode consiste en un problème d'évolution jusqu'à un temps très grand, il est avantageux d'utiliser un grand pas de temps pour résoudre le problème. Le schéma semi-implicite développé dans le deuxième article permet ceci. Notons ici que Achdou et al. (2008) utilise un schéma semi-implicite pour un cas isotrope. La méthode présentée dans cette thèse se veut plus générale.

Chapitre 2

APPLICATION DE LA MÉTHODE LEVEL-SET À LA PROPAGATION ANISOTROPE DE FEUX DE FORÊT

Ce chapitre contient l'article «A level-set algorithm for the simulation of anisotropic firespread». Le but principal de l'article est d'expliquer à la communauté scientifique qui s'intéresse aux feux de forêt comment utiliser rigoureusement la méthode «level-set» pour simuler la propagation d'un feu quand le modèle de propagation utilisé est le modèle de l'ellipse de Richards, un des plus populaires. Les contributions scientifiques principales du travail sont :

(1) Une nouvelle méthode numérique avec preuve de convergence pour résoudre l'équation aux dérivées partielles qui représente la propagation d'un feu de forêt.

(2) Un algorithme de simulation de feu de forêt basé sur la méthode «level-set» permettant de traiter des données réelles. Tous les paramètres importants sont pris en compte. Le feu se propage sur une topographie en présence de vent et avec paramètres spatialement hétérogènes.

A LEVEL-SET ALGORITHM FOR THE SIMULATION OF ANISOTROPIC FIRESPREAD

Alexandre Desfossés Foucault

Prepared for : International journal of wildland fire

ABSTRACT. An algorithm for the solution of anisotropic firespread based on Richards' ellipse model and on the level-set method is presented. An introduction to the level-set method and to the numerical solutions of corresponding Hamilton-Jacobi partial differential equations gives the readers the necessary mathematical background. The firespread problem is then formulated in the level-set context. This gives rise to an equation whose numerical solution requires a new scheme which is introduced in this paper. Numerical simulations show this scheme gives convergent solutions. Examples of simulation with data coming from the Prometheus software shows the applicability of the method to real-life cases.

2.1. Introduction

The goal of this paper is to present a new algorithm for forest fire simulation based on the level-set method and on Richards' ellipse model (Richards, 1994) for fire growth. Firespread simulators are important for government agencies because predictions on the growth of fires are crucial in deciding whether certain regions must be evacuated or not, and they can also help plan fire fighter interventions. For example, the government of Alberta uses the Prometheus software (Barber et al., 2007) to aid in their decisions.

The Prometheus software is based on Richards' ellipse model, which assumes a new fire grows from every point of a flame front in the shape of an ellipse aligned with the wind. The flame front is seen as an infinitely thin curve. Then, the curve expands to simulate fire growth. Prometheus computes the evolution of the curve using a marker method. The idea behind this approach is to choose points on a flame front and to make them move according to the predictions of Richards' model. This leads to some difficulties. The problems arising from this solution to the front-tracking problem are what justify our use of the level-set method.

The level-set method (Osher and Sethian, 1988) is a general interface-tracking mathematical technique with applications ranging from fluid dynamics to image processing. It consists in embedding any interface in a higher-dimension object (for example, a curve can be seen as a level-set curve of a surface) and in evolving the surface by solving numerically a partial differential equation. How to apply

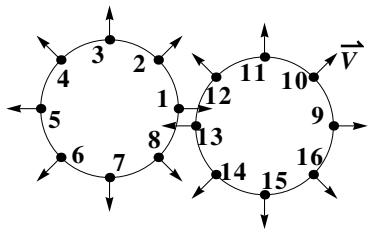
it to problems with isotropic propagation is well-known, but adapting it to anisotropic cases is not straightforward. The main contributions of this paper are a new way to formulate the numerical solution for the level-set partial differential equation in an anisotropic case and an application of this numerical method to the simulation of forest fires with realistic data.

This article is organized in the following way. We begin by discussing the inherent problems of marker methods. An introduction to the level-set method based on the example of isotropic propagation of a fire in homogeneous fuels follows. We then give the partial differential equation that must be solved to simulate firespread when the propagation is anisotropic, in two dimensions and on a topography. The next section explains how to obtain the numerical solution of this equation. We present a few results of our firespread algorithm in idealized test cases where the solution is known and in a case where the spread rates come from the Prometheus software.

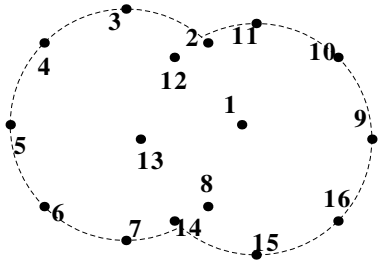
2.2. The level-set method

When considering large-scale forest fire propagation problems, flame fronts can be modelled as infinitely thin curves expanding according to propagation speeds. A fire propagation problem can then be tackled as an interface tracking problem. Prometheus solves this using a marker method (Barber et al., 2007). This approach consists in placing markers on the flame front and in moving them at speeds corresponding to the fire propagation rate. This technique leads to some well known difficulties. Marker trajectories can cross, so updating the position of the flame front can be hard.

Figure 2.1 illustrates problems arising from the collision of two fronts.



(a) Initial position of the front



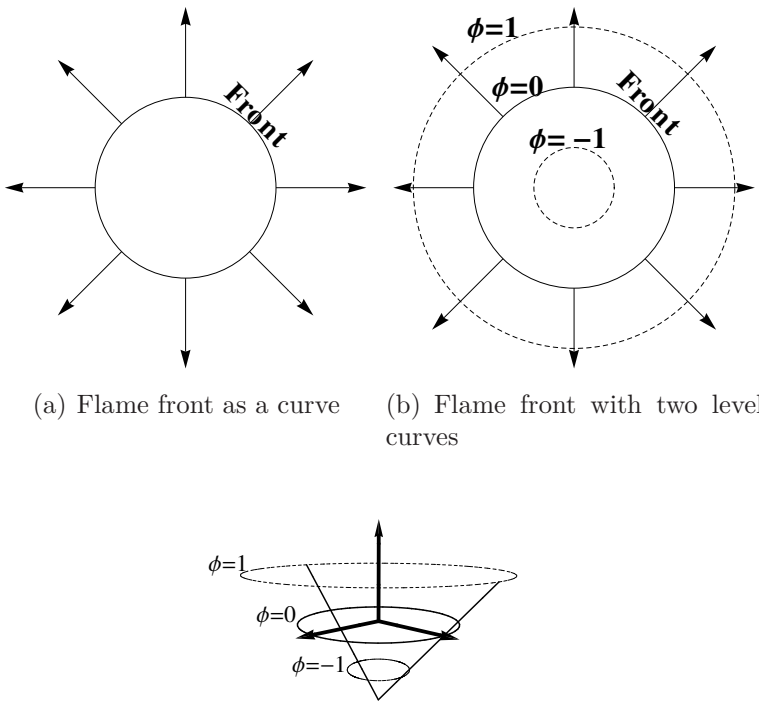
(b) Position after one time step

Figure 2.1. Difficulties with the marker method when two interfaces are merging

It can be difficult to decide where a flame front is just by looking at the marker positions. Figure 2.1(a) shows the initial position of the front and of the associated markers. In figure 2.1(b), we see the position of the markers after one time step. The true position of the front is shown as a dashed line. In this case, markers 1, 8, 12 and 13 become superfluous and must be eliminated while marker pairs 2-11 and 7-14 must be connected. Algorithms to untangle the markers are necessary, but these algorithms are rather arbitrary. The convergence of such a numerical solution cannot be proven.

2.2.1. Derivation of the level-set partial differential equation

An efficient approach that avoids the numerous issues of the marker method is the level-set method (Osher and Sethian, 1988). The interface is represented as a level-set curve of a higher dimension object. For example, consider a circular flame front on a flat plane, satisfying the equation $x^2 + y^2 = r^2$. This curve is the zero level set of any function that is positive outside the circle and negative inside. Figure 2.2 shows the flame front as the zero level curve of a cone of equation $z = \sqrt{x^2 + y^2} - r$. The left-most image shows the flame front. In the middle,



(a) Flame front as a curve

(b) Flame front with two level curves

(c) Implicit representation of the circle as a level curve of a cone

FIGURE 2.2. Different representations of a circular flame front

some level curves of the implicit representation of the flame front are drawn and finally, the picture on the right shows the cone with different level-curves in three dimensions. The next step is to obtain an equation describing the deformation of the surface as time progresses. This deformation of the surface is chosen in such a way that guarantees the zero-level curve moves according to the spread rate

model. The following is the presentation of the derivation of the method given in (Sethian, 1999).

Let ϕ be a function having the flame front as zero level curve. Consider a particle on the front with trajectory $\vec{x}(t)$. We force such a particle to stay on the zero level curve of ϕ at all times :

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

By differentiating this equation with respect to t and by applying the chain rule, we get

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

For t fixed, $\vec{x}'(t)$ stands for the propagation speed at a point of the zero level curve. By repeating the argument for the other level curves, we obtain the following partial differential equation.

$$\phi_t + \vec{v} \cdot \nabla\phi = 0. \tag{2.2.1}$$

Thus, the zero level curve of ϕ will move according to the spread rates predicted by the physical model and the other level curves will evolve according to the speed a fictitious fire at every point on those curves would move.

2.2.2. First example-isotropic propagation

Consider the example of isotropic propagation of an interface on a horizontal plane. This corresponds to a fire propagation case with no wind. The vector field $\vec{v}$ of propagation speeds must be determined. The main idea is to use the Huygens principle for wave propagation. To see how a flame front will propagate, suppose every point of the front is a new fire ignition point and suppose also every new fire grows as a circle, as shown on figure 2.3. The position of the flame front after time Δt is given by the envelope of the new fires.

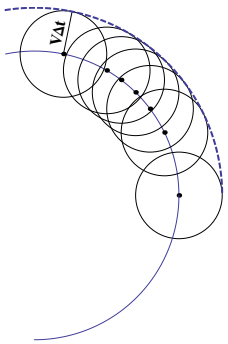


FIGURE 2.3. Huygens principle in absence of wind

The radius $V\Delta t$ of the circle depends on physical parameters defined at this point, such as the fuel type and temperature, and on the duration of the combustion Δt. To find $\vec{v}$, we require the front's normal vector at time $t_0 + \Delta t$ to be the same as the front's normal vector at time t_0. So, we are looking for the point on the circle satisfying this property. The normal vector at time t_0 is given by $\vec{n}=\frac{\nabla\phi}{\|\nabla\phi\|}$. To find a normal vector at time $t_0 + \Delta t$, consider a parameterization of the circle drawn on figure 2.3 :

$$x(s) = V\Delta t\cos(s)$$
$$y(s) = V\Delta t\sin(s).$$

The unit normal vector of this curve is $\vec{w} = (\cos(s), \sin(s))$. We are looking for the value of s making these two normal vectors identical. This gives the new position of the center of the circle of figure 2.3 after a time Δt, and so we can write the propagation speed $\vec{v}$. The value of s can be found by comparing the squares of the quotients of the x and y components of the two expressions for the normal vectors :

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

After some manipulations, we find :

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

The new position of the center of the circle is

$$x = Vdt\frac{\phi_x}{\|\nabla\phi\|}$$
$$y = Vdt\frac{\phi_y}{\|\nabla\phi\|}.$$

To obtain the speed, we divide this by dt. Finally, we get the following equation for the propagation speeds.

$$\vec{v} = \frac{V\nabla\phi}{\|\nabla\phi\|}.$$

This allows us to write the isotropic fire evolution equation as

$$\phi_t + V\|\nabla\phi\| = 0.$$

2.2.2.1. *Discretization of the equation*

The first step for solving numerically this equation is to discretize the fire-spread domain with a uniform grid of $(N \times M)$ points with a given space step

h. We must discretize the time and spatial derivatives. The time discretization will give rise to an iterative process. Using forward Euler discretization, the time derivative at (x_i, y_j) at iteration n is

$$\phi_t = \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t}$$

where $\phi_{i,j}^n$ is the solution at time t^n at point (x_i, y_j) . The numerical approximation of $\phi_{i,j}^{n+1}$ is given by :

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + V\Delta t\sqrt{\phi_x^2|_{i,j}^n + \phi_y^2|_{i,j}^n}. \tag{2.2.2}$$

The main challenge in obtaining the numerical solution of this equation is the approximation of the spatial derivatives. There are many ways of choosing an approximation. For example, the formulas for the forward and backward space derivatives in x (ϕ_x^+ and ϕ_x^-) at point (i, j) at iteration n are

$$\begin{aligned} \phi_x^+|_{i,j}^n &= \frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{h} \\ \phi_x^-|_{i,j}^n &= \frac{\phi_{i,j}^n - \phi_{i-1,j}^n}{h}. \end{aligned}$$

Figure 2.4 shows what happens if only backward space derivatives are used. This

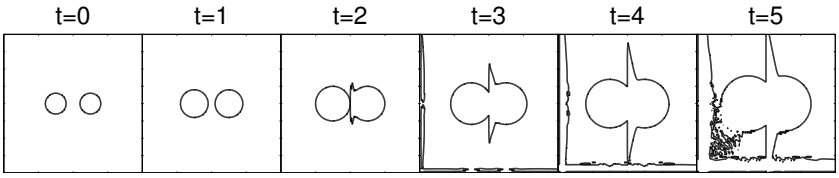


FIGURE 2.4. Zero level-set of the solution to the problem of 2 circles growing outwards obtained by the level-set method with the backward approximation for the spatial derivatives

solution is clearly wrong since it should be two circles expanding and merging. To obtain a good solution, the choice of the numerical derivatives must depend on the direction of flow of information, as shown on figure 2.5.

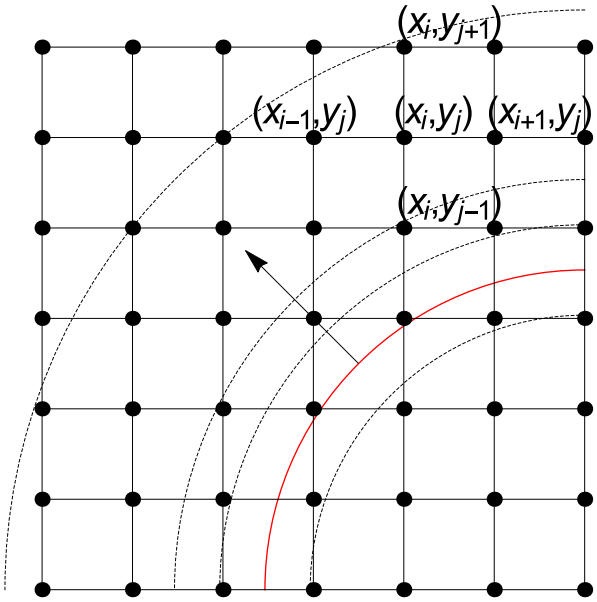


FIGURE 2.5. Discretization of the spatial derivatives

Figure 2.5 shows the flame front in red and other level sets in a dashed black line. The information flows along the gradient of ϕ, as illustrated by the arrow. The choice of approximation of the derivatives at (x_i, y_j) must account for the fact that at this point, information flows from the right to the left and from the bottom to the top. Hence, backward differences must be used in y, but forward differences must be chosen in x.

The previous discussion gives intuitive arguments to justify the choice of approximation of spatial derivatives. A more mathematically rigorous argument is a numerical scheme of equation (2.2.2) will converge to the solution of the equation if it is monotone, consistent and stable (Barles and Souganidis, 1991). Consistency means that as the time and space steps go to zero, the discrete equation converges to the differential equation. Stability means numerical errors remain bounded. It is ensured by choosing sufficiently small time-steps. Δt must be chosen to respect the CFL condition (Osher and Fedkiw, 2002). In this case, the condition is

$$\Delta t < \frac{h}{V}$$

A scheme of the type $\phi_{ij}^{n+1} = H(\phi_{i,j}^n, \phi_{i+1,j}^n, \phi_{i-1,j}^n, \phi_{i,j+1}^n, \phi_{i,j-1}^n)$ is monotone if H is a non-decreasing function of its arguments.

It has been shown that Godunov's scheme (Osher and Fedkiw, 2002) for this problem is monotone. Applying it to equation (2.2.2) gives formula (Rouy and Tourin,

1992) :

$$\phi_x^2 = \max\left(\max\left(\phi_x^-, 0\right)^2, \min\left(\phi_x^+, 0\right)^2\right). \tag{2.2.3}$$

The formula in y is similar. This formula uses the values of the derivatives to determine the direction of propagation and gives the right choice for the approximations. When information at a point comes only from one direction, such as in the previous example, it picks either the backward or forward derivative. When it comes from two directions, it picks the approximation which maximizes the Hamiltonian. So, if information is coming faster from the left than from the right, backward differences are chosen and vice-versa. In all other cases, it chooses 0 for the derivatives.

2.2.2.2. *Initial and boundary conditions*

Consider equation (2.2.1). Since the solution at any iteration comes from the solution at the previous iteration, an initial condition must be specified. This initial condition must be the implicit representation of the initial flame front. There is an infinity of possible choices but for many reasons given in Sethian (1999), the traditional choice is to take ϕ as the signed distance function to the front. The value of this function at a given point is the distance from the flame front to this point, multiplied by -1 if the point is inside the front. For example, consider the circle of radius 1 centered at the origin. The signed distance function to this curve is $\phi(x, y) = \sqrt{x^2 + y^2} - 1$. This is the cone shown on figure 2.2. A more complicated example is the signed distance function to two circles (figure 2.6) . This function is fairly smooth, which is useful when evaluating the numerical

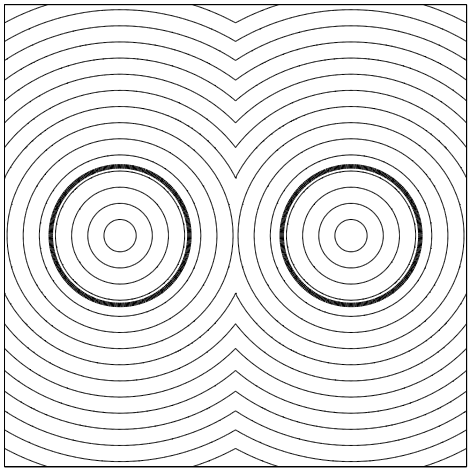


FIGURE 2.6. Some level sets of the signed distance function to 2 circles

derivatives. Note that as we move forward in time, ϕ does not remain a signed distance function. A process called reinitialization (described briefly in a later section) can be used to force the level-set function to be the signed distance function after every iteration.

The last element we need for a numerical level-set algorithm is a set of boundary conditions. The type of partial differential equations we consider requires calculating spatial derivatives on the boundaries of the firespread domain. The points on the boundaries don't have exterior neighbors, so another way of approximating the derivatives at these points must be chosen. Note that the firespread domains studied are very large, so the boundaries should have no effect on simulated fires. To approximate infinitely large domains, a linear extrapolation can be done. Points are added around the firespread domain and a fictitious value of the level-set function is extrapolated at these points. Consider the value of the level-set function at the points on the first column of the domain $\phi_{1,j}$. An extra column of points $\phi_{0,j}$ is added to the left using the following formula for the values of $\phi_{0,j}$:

$$\phi_{0,j} = 2 \cdot \phi_{1,j} - \phi_{2,j}.$$

Suppose the domain has (N×M) points. Rows and columns of points can be added to the top ($\phi_{i,0}$), to the right ($\phi_{N+1,j}$) and to the bottom ($\phi_{i,M+1}$) with the following formulas.

$$\phi_{i,0} = 2 \cdot \phi_{i,1} - \phi_{i,2}$$
$$\phi_{N+1,j} = 2 \cdot \phi_{N,j} - \phi_{N-1,j}$$
$$\phi_{i,M+1} = 2 \cdot \phi_{i,M} - \phi_{i,M-1}$$

2.2.2.3. *Numerical solution for isotropic propagation*

Now that all the elements necessary to use the level-set method are in place, the correct solution to the example of two fronts colliding can be shown (figure 2.7). This solution is clearly better than the one shown in figure 2.4.

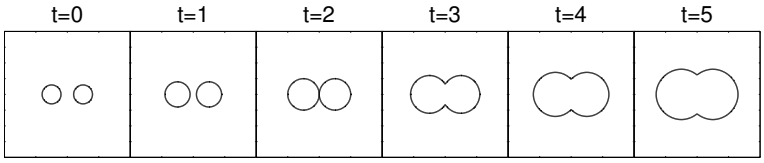


FIGURE 2.7. Numerical solution to the problem of isotropic propagation of two circles

2.2.3. Anisotropic firespread on a horizontal plane

In the case of anisotropic propagation, the propagation speed field is more complicated and it will be harder to discretize the spatial derivatives. Let's begin by obtaining the equation describing the firespread in this case.

A partial differential equation for the evolution of a flame front using Richards' ellipse model (Richards, 1994) must be obtained (for an application of the level-set method in the case of isotropic propagation, see Mallet et al. (2009)). Assume every point of a flame front is a source of a new fire growing in the shape of an ellipse aligned with the wind, as shown in figure 2.8.

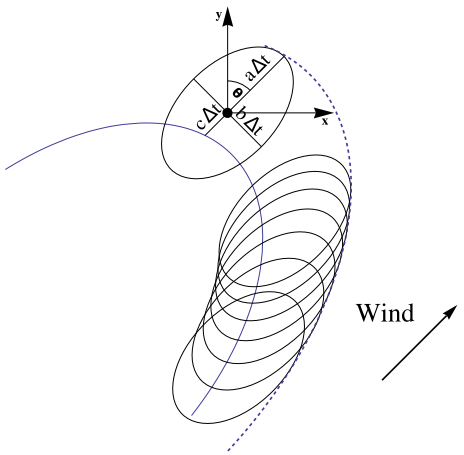


FIGURE 2.8. Illustration of the ellipse model

With this notation, $a + c$ is the rate of spread in the direction of the wind, $a - c$ is the back rate of spread and b is the flank rate of spread. In what follows, these rates of spread will be assumed known at every point of the domain of firespread. Parameter c can be seen as the translation speed of the front caused by the wind. Parameter a is then the propagation speed of the fire in the direction of the wind, minus the translation speed coming from the wind. If $c = 0$ and $a = b$, this reduces to the isotropic case.

In Barber et al. (2007), the authors show how to use this to derive the partial differential equation. The derivation is the same as in the isotropic case (after a suitable change of variables). First, let's define the rotation matrix

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

with θ the wind direction. Define also a matrix A containing the propagation speeds

$$A = \begin{pmatrix} b & 0 \\ 0 & a \end{pmatrix}$$

and a vector $\vec{C}$ giving the advection speed

$$\vec{C} = R_\theta \begin{pmatrix} 0 \\ c \end{pmatrix}$$

The equation describing the fire spread is then

$$\phi_t + \|\tilde{A}\nabla\phi\| + \vec{C} \cdot \nabla\phi = 0. \tag{2.2.4}$$

Note that by setting

$$\begin{aligned} k_1 &= (b^2 \cos^2\theta + a^2 \sin^2\theta) \\ k_2 &= (b^2 \sin^2\theta + a^2 \cos^2\theta) \\ k_3 &= (b^2 \sin\theta\cos\theta - a^2 \sin\theta\cos\theta), \end{aligned}$$

the equation can be written as

$$\phi_t + \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y} + \vec{C} \cdot \nabla\phi = 0. \tag{2.2.5}$$

Writing it this way will help us discretize the spatial derivatives further on.

2.2.4. Fire propagation partial differential equation on a topography

For the case of propagation on a topography, the partial differential equation can still be written as (2.2.5), but the values of the k_1, k_2 and k_3 coefficients are different. Define a topography $z = f(x, y)$. The flame front must remain on this topography so it is effectively moving in two-dimensional (2D) space. It is possible to write a partial differential equation with two spatial dimensions to describe this evolution problem by projecting the flame front on the $x - y$ plane.

In this case, assume $a + c$ is the rate of firespread along the topography in the direction of the wind (called $\hat{e}_y$) and b is the rate of spread along topography but perpendicular to the wind (along direction $\hat{e}_x$). Define p and q as the x and y derivatives of the topography height function. Define also the two vectors described previously $\hat{e}_x$ and $\hat{e}_y$, along with a vector perpendicular to the topography at every point called $\hat{e}_z$ (refer to figure 2.9 for an illustration of these new base

vectors). Finally, define two coefficients n_1 and n_2 :

$$
\begin{aligned}
n_1 &= \sqrt{1+p^2+q^2} \\
n_2 &= \sqrt{1+(p\sin\theta+q\cos\theta)^2} \\
\hat{e}_z &= \frac{1}{n_1}(-p,-q,1) \\
\hat{e}_y &= \frac{1}{n_2}(\sin\theta,\cos\theta,p\sin\theta+q\cos\theta) \\
\hat{e}_x &= \frac{1}{n_1 n_2}\Big((1+q^2)\cos\theta+pq\sin\theta,-(1+p^2)\sin\theta-pq\cos\theta,-q\sin\theta+p\cos\theta\Big)
\end{aligned}
\tag{2.2.6}
$$

With those definitions, the k_1, k_2 and k_3 coefficients become

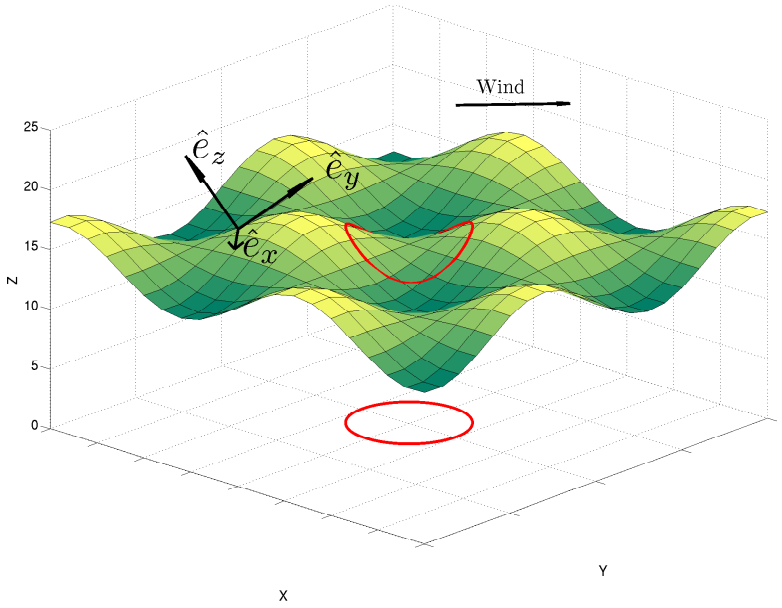


FIGURE 2.9. The 3 new base vectors on a surface

$$
\begin{aligned}
k_1 &= \frac{a^2n_1^2\sin^2\theta+b^2((1+q^2)\cos\theta+pq\sin\theta)^2}{n_1^2n_2^2} \\
k_2 &= \frac{(a^2n_1^2+b^2p^2q^2)\cos^2\theta+b^2(1+p^2)\sin\theta(2pq\cos\theta+(1+p^2)sin\theta)}{n_1^2n_2^2} \\
k_3 &= \frac{b^2pq((1+q^2)\cos^2\theta+(1+p^2)\sin^2\theta)}{n_1^2n_2^2}+ \\
&\quad \frac{\frac{1}{2}(-a^2n_1^2+b^2(1+q^2+p^2(1+2q^2)))\sin 2\theta}{n_1^2n_2^2}
\end{aligned}
$$

The derivation of those values is included in Appendix A.

2.3. Numerical scheme

When considering either the propagation on a horizontal plane or on a topography, we must solve an equation of the form

$$\phi_t + \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y} + \vec{C}\cdot\nabla\phi = 0 \tag{2.3.1}$$

One way to solve a partial differential equations with two terms (besides ϕ_t) like this one is to use a process called splitting. The two terms are treated separately. At every iteration, the equations $\phi_t + \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y} = 0$ and $\phi_t + \vec{C}\cdot\nabla\phi = 0$ are solved one after the other. This gives us the contributions of the two terms needed to calculate ϕ^{n+1}.

So, to obtain the solution at iteration $n+1$ from the solution at iteration n, we must

(1) Choose a time step compatible with the two equations : $\Delta t = 0.5\frac{\Delta x}{max(a,b,c)}$

(2) Solve the equation with the propagation term for one time step. This gives an intermediate solution $\phi^{n+\frac{1}{2}}$:

$$\phi_{i,j}^{n+\frac{1}{2}} = \phi_{i,j}^{n} - \Delta t\sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y}$$

In this equation, ϕ_x and ϕ_y stand for appropriate discretizations of the spatial derivatives of $\phi_{i,j}^n$

(3) Solve the equation with the advection term for one time step using $\phi^{n+\frac{1}{2}}$ as the previous iteration solution.

$$\phi_{i,j}^{n+1} = \phi_{i,j}^{n+\frac{1}{2}} - \Delta t\vec{C}\cdot\nabla\phi^{n+\frac{1}{2}}$$

where $\nabla\phi^{n+\frac{1}{2}}$ is a suitable approximation of the gradient of $\phi^{n+\frac{1}{2}}$ at point (x_i, y_j).

Numerical solutions of equations of type $\phi_t + \vec{C}\cdot\nabla\phi = 0$ have been extensively studied. See for example Osher and Fedkiw (2002). The discretization of the other term is more complicated. The partial differential equation can be written without the advection term as $\phi_t + H(\phi_x, \phi_y) = 0$. It is an unsteady Hamilton-Jacobi equation. The Hamiltonian $H(\phi_x, \phi_y)$ must be approximated carefully to make sure the numerical solution converges to the solution of the equation. As explained previously, a valid approximation of the spatial derivatives for this problem is Godunov's scheme.

In Kao et al. (2003), the authors show how to obtain a Godunov Hamiltonian for a static anisotropic Hamilton-Jacobi equation. In the next section, we will apply this approach to the unsteady case, and introduce a new way to evaluate this numerical Hamiltonian.

2.3.1. Challenges with the Godunov method for the anisotropic case

Obtaining a Godunov's scheme in the anisotropic case is difficult. To understand why this is the case, it is necessary to look at the original definition of Godunov's scheme for the Hamilton-Jacobi partial differential equation. Begin by defining at every domain point an interval I^x based on the two approximation of the spatial derivatives :

$$I^x := [min(\phi_x^-, \phi_x^+), max(\phi_x^-, \phi_x^+)].$$

Godunov's Hamiltonian for the problem is then given by :

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

where $ext_x H(\phi_x)$ is defined thus :

(1) If $\phi_x^- < \phi_x^+$, then $ext_x H$ is obtained by taking the value of $\phi_x \in I^x$ which minimizes H.

(2) If $\phi_x^- > \phi_x^+$, then $ext_x H$ is obtained by taking the value of $\phi_x \in I^x$ which maximizes H.

(3) If $\phi_x^- = \phi_x^+$, then $ext_x H$ is obtained by taking either of the approximations ϕ_x.

The formulas in y are the same. The method is easy to apply in the isotropic case because the optimization can occur dimension by dimension. The Hamiltonian in the isotropic case is $H(\nabla\phi) = \sqrt{\phi_x^2 + \phi_y^2}$. Whatever value is set for ϕ_y has no influence on the choice of value for ϕ_x computed by the ext_x operator, and vice-versa. For the forest fire equation, the Hamiltonian is given by $H(\nabla\phi) = \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y}$. The term $-2k_3\phi_x\phi_y$ means the ext_x and ext_y operators cannot be decoupled. The following subsections give a valid Godunov Hamiltonian for the forest fire problem and explain how to obtain it.

2.3.2. Formula for the numerical Hamiltonian

The equation we are solving can be written as

$$\phi_t + H(\phi_x, \phi_y) = 0 \tag{2.3.2}$$

with

$$H(\phi_x, \phi_y) = \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y}.$$

Define

$$\widetilde{\nabla\phi} = \begin{pmatrix} \tilde{\phi_x} \\ \tilde{\phi_y} \end{pmatrix} = \begin{pmatrix} k_1 & -k_3 \\ -k_3 & k_2 \end{pmatrix} \begin{pmatrix} \phi_x \\ \phi_y \end{pmatrix}$$

With these definitions, the original equation becomes

$$\phi_t + \frac{\widetilde{\nabla\phi} \cdot \nabla\phi}{\sqrt{\widetilde{\nabla\phi} \cdot \nabla\phi}} = 0.$$

This equation describes the evolution of a flame front under the action of a velocity field in the direction of $\widetilde{\nabla\phi}$. As in the case of isotropic propagation, the choice for the approximation of the spatial derivatives has to reflect the direction of flow of information given by $\widetilde{\nabla\phi}$.

To give the formula for the spatial derivatives, we must first introduce notation for all the possible choices for the approximation of $\widetilde{\nabla\phi}$.

$$\widetilde{\phi_x}^{++} = k_1\phi_x^+ - k_3\phi_y^+ \qquad \widetilde{\phi_y}^{++} = k_2\phi_y^+ - k_3\phi_x^+$$
$$\widetilde{\phi_x}^{+-} = k_1\phi_x^+ - k_3\phi_y^- \qquad \widetilde{\phi_y}^{+-} = k_2\phi_y^+ - k_3\phi_x^-$$
$$\widetilde{\phi_x}^{-+} = k_1\phi_x^- - k_3\phi_y^+ \qquad \widetilde{\phi_y}^{-+} = k_2\phi_y^- - k_3\phi_x^+$$
$$\widetilde{\phi_x}^{--} = k_1\phi_x^- - k_3\phi_y^- \qquad \widetilde{\phi_y}^{--} = k_2\phi_y^- - k_3\phi_x^-$$

Also define variables noting the orientation of the approximations of $\widetilde{\nabla\phi}$.

$$D^{++} = \begin{cases} 1 \text{ if } \widetilde{\phi_x}^{++} \le 0 \text{ and } \widetilde{\phi_y}^{++} \le 0 \\ 0 \text{ else} \end{cases} \qquad D^N = \begin{cases} 1 \text{ if } \phi_y^- \ge 0 \\ 0 \text{ else} \end{cases}$$

$$D^{+-} = \begin{cases} 1 \text{ if } \widetilde{\phi_x}^{+-} \le 0 \text{ and } \widetilde{\phi_y}^{+-} \ge 0 \\ 0 \text{ else} \end{cases} \qquad D^W = \begin{cases} 1 \text{ if } \phi_x^+ \le 0 \\ 0 \text{ else} \end{cases}$$

$$D^{-+} = \begin{cases} 1 \text{ if } \widetilde{\phi_x}^{-+} \ge 0 \text{ and } \widetilde{\phi_y}^{-+} \le 0 \\ 0 \text{ else} \end{cases} \qquad D^S = \begin{cases} 1 \text{ if } \phi_y^+ \le 0 \\ 0 \text{ else} \end{cases}$$

$$D^{--} = \begin{cases} 1 \text{ if } \widetilde{\phi_x}^{--} \ge 0 \text{ and } \widetilde{\phi_y}^{--} \ge 0 \\ 0 \text{ else} \end{cases} \qquad D^E = \begin{cases} 1 \text{ if } \phi_x^- \ge 0 \\ 0 \text{ else} \end{cases}$$

Variables D^i indicate whether a certain approximation of the derivative is appropriate, given the direction of the flow of information at a point. For example, D^{--} corresponds to using the backward derivatives in x and y. This choice is correct when the information flows towards the north-east. So, D^{--} is equal to one at points where the vector giving the direction of the flow of information points towards a direction in the north-east quadrant.

Then, the numerical Hamiltonian is of form

$$\hat{H} = \max_i \sqrt{h^i}$$

where the potential h^i are

$$\begin{aligned}
h^{NE} &= (\widetilde{\phi_x}^{--}\phi_x^- + \widetilde{\phi_y}^{--}\phi_y^-)D^{--} \\
h^{NW} &= (\widetilde{\phi_x}^{+-}\phi_x^+ + \widetilde{\phi_y}^{+-}\phi_y^-)D^{+-} \\
h^{SW} &= (\widetilde{\phi_x}^{++}\phi_x^+ + \widetilde{\phi_y}^{++}\phi_y^+)D^{++} \\
h^{SE} &= (\widetilde{\phi_x}^{-+}\phi_x^- + \widetilde{\phi_y}^{-+}\phi_y^+)D^{-+} \\
h^N &= (\phi_y^-)^2\frac{(k_1k_2-k_3^2)}{k_1}D^N \\
h^W &= (\phi_x^+)^2\frac{(k_1k_2-k_3^2)}{k_2}D^W \\
h^S &= (\phi_y^+)^2\frac{(k_1k_2-k_3^2)}{k_1}D^S \\
h^E &= (\phi_x^-)^2\frac{(k_1k_2-k_3^2)}{k_2}D^E.
\end{aligned}$$

These values correspond to the square of possible values of the numerical Hamiltonian. To each choice of the derivatives is associated one candidate. For example, the numerical Hamiltonian corresponding to the backward derivatives in x and y is $\sqrt{h^{NE}}$.

2.3.3. Derivation of the formula

The formulas for h^{NE} and h^N will be obtained in this section. The derivation for the other cases is similar. The starting point is the Bellman formula for the numerical Hamiltonian (Kao et al., 2003) :

$$\begin{aligned}
\hat{H} &= \max_{\alpha\in[0,2\pi]}\left[(\cos\alpha)^+(\phi_x^-) + (\cos\alpha)^-(\phi_x^+) + (\sin\alpha)^+(\phi_y^-) + (\sin\alpha)^-(\phi_y^+)\right]w(\alpha) \\
&= \max_{\alpha\in[0,2\pi]} G(\alpha)
\end{aligned}$$

where $(\cos\alpha)^+ = \max(\cos\alpha, 0)$ and $(\cos\alpha)^- = \min(\cos\alpha, 0)$, with similar definitions for $(\sin\alpha)^\pm$. Here the form $H = \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_Y}$ is useful. With this, $w(\alpha)$ is given by

$$w(\alpha) = \sqrt{\frac{k_1k_2-k_3^2}{k_1\sin^2\alpha + k_2\cos^2\alpha + 2k_3\cos\alpha\sin\alpha}}$$

To find α maximizing G, we divide the interval $[0, 2\pi]$ in four regions, we look for critical points of G inside these regions and we consider the values of G on the boundaries. To find the candidate for the northeastern region, suppose $\alpha \in (0, \frac{\pi}{2})$.

(1) The condition $G'(\alpha) = 0$ gives $\tan\alpha = \dfrac{\widetilde{\phi_y}^{--}}{\widetilde{\phi_x}^{--}}$.

(2) If α is in $(0, \frac{\pi}{2})$ (e.g. if $D^{--} = 1$), then α is a potential maximizer of G. Then, evaluate G at the point $\alpha = \arctan \frac{\widetilde{\phi_y}^{--}}{\widetilde{\phi_x}^{--}}$. After simplification, we find $G(\alpha) = \sqrt{\widetilde{\phi_x}^{--}\phi_x^- + \widetilde{\phi_y}^{--}\phi_y^-}$.

Consider now the case where the maximizer is in $\alpha = \frac{\pi}{2}$. This gives the following value for a potential maximizer :

$$G\left(\frac{\pi}{2}\right) = \phi_y^- \sqrt{\frac{k_1 k_2 - k_3^2}{k_1}}.$$

This value is admissible only if ϕ_y^- is positive (if $D^N = 1$).

2.3.4. Properties of the scheme and comparison with existing methods

This scheme is monotone since it is based on a Godunov Hamiltonian as shown in Kao et al. (2003). It is consistent, since in the limit where $\Delta t \to 0$ and $\Delta x \to 0$, the approximations of the spatial and temporal derivatives converge to the actual derivatives. It is stable, provided Δt is chosen in a manner respecting the CFL condition. So, the scheme is convergent (Barles and Souganidis, 1991). The expected order of convergence for this scheme is $\frac{1}{2}$ (Crandall and Lions, 1984). It will be experimentally verified in an upcoming section.

Other schemes exist for solving this problem, such as the Lax-Friedrichs and the Roe-Fix scheme (Osher and Fedkiw, 2002). These schemes work by adding numerical diffusion to the original equation. This additional smoothing deals with the difficulties coming from caracteristics colliding or moving away from each other at the cost of smoothing the solution. So, a Godunov-type scheme gives a more precise solution.

2.3.5. Applying this scheme to the isotropic case

One way to validate this scheme is to apply it to the isotropic case and to compare the result with formula (1.5.2). This formula maximizes the Hamiltonian by maximizing separately ϕ_x and ϕ_y while the scheme of the previous section maximizes the Hamiltonian by considering both derivatives at once. But, the end result is the same. The following is a list of all the choices for ϕ_x in different cases. In all cases, the choices made by the two schemes are the same.

1 If $\phi_x^- > 0$ and $\phi_x^+ > 0$, the Rouy-Tourin choice for ϕ_x is ϕ_x^-. The new scheme choice is either candidate h^{NE}, h^{SE} or h^E. The scheme rejects from the start candidates h^{NW}, h^{SW} and h^W and clearly, setting $\phi_x = 0$ does not

maximize $H = \sqrt{\phi_x^2 + \phi_y^2}$ so the scheme rejects candidates h^N and h^S and thus chooses $\phi_x = \phi_x^-$.

2 If $\phi_x^- < 0$ and $\phi_x^+ < 0$, the Rouy-Tourin choice for ϕ_x^2 is $\max((\phi_x^-)^2, (\phi_x^+)^2)$. The new scheme choice is either candidate h^{NW}, h^{SW} or h^W using similar arguments as in the previous case. So the choice is $\phi_x = \phi_x^+$.

3 If $\phi_x^- > 0$ and $\phi_x^+ < 0$, the Rouy-Tourin choice for $\phi_x = \max((\phi_x^-)^2, \phi_x^+)^2)$. In the new scheme, all candidates are admissible. ϕ_x will be chosen to maximize $\sqrt{\phi_x^2 + \phi_y^2}$ so $\phi_x^2 = \max((\phi_x^-)^2, \phi_x^+)^2)$.

4 If $\phi_x^- < 0$ and $\phi_x^+ > 0$, the Rouy-Tourin choice is $\phi_x = 0$. The new scheme choice is also $\phi_x = 0$ since then no candidate with a non-zero x derivative is admissible.

In all cases, the choices are the same. The arguments presented here also apply for the y derivative so the two schemes are equivalent in the isotropic case.

2.4. Validation with idealized test cases

2.4.1. Algorithm

For some cases, exact solutions are known. This will allow us to validate the scheme. For example, the exact solutions for the self-similar propagation of circles and ellipses can be found. Convergence will also be shown in a case where the exact solution is not known. Here is the algorithm used to obtain the results.

(1) Calculate the k_1, k_2 and k_3 coefficients everywhere on the domain from the propagation parameters.

(2) Define the initial value of the level-set function. A typical choice is $\phi = \sqrt{x^2 + y^2} - R$ for the propagation of a circle of initial radius R.

(3) For every iteration, use the splitting explained in section 2.3 to consider the propagation term and the advection term separately.

(a) Solve $\phi_t + \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y} = 0$ for one time step at every point :

$$\phi_{i,j}^{n+\frac{1}{2}} = \phi_{i,j}^n - \Delta t\hat{H}$$

with the $\hat{H}$ given in the previous section.

(b) Solve $\phi_t + \vec{C} \cdot \nabla\phi = 0$ for one time-step with an upwind scheme :

$$\phi_{i,j}^{n+1} = \phi_{i,j}^{n+\frac{1}{2}} - \Delta t\frac{(\max(c\sin\theta, 0)\phi_x^-|_{i,j}^{n+\frac{1}{2}} + \min(c\sin\theta, 0)\phi_x^+|_{i,j}^{n+\frac{1}{2}})}{n2}$$

$$-\Delta t \frac{(\max(c\cos\theta, 0)\phi_y^-|_{i,j}^{n+\frac{1}{2}} + \min(c\cos\theta, 0)\phi_y^+|_{i,j}^{n+\frac{1}{2}})}{n2}$$

where n_2 is given in 2.2.6.

2.4.2. Numerical tests

2.4.2.1. *Self-similar propagation of an ellipse on a horizontal plane*

The first example is the case of the self-similar propagation of a rotated ellipse on a horizontal plane. Table 2.1 shows the parameters used for this case.

Parameter	Value
a	2
b	1
c	0
θ	$\frac{\pi}{3}$
ϕ_{init}	$\sqrt{\frac{(x\cos(\theta)-y\sin(\theta))^2}{b^2} + \frac{(x\sin(\theta)+y\cos(\theta))^2}{a^2}} - 1$
t_{final}	5
Exact solution	$\sqrt{\frac{(x\cos(\theta)-y\sin(\theta))^2}{b^2} + \frac{(x\sin(\theta)+y\cos(\theta))^2}{a^2}} - 1 - t$
Δx	0.5
Δt	$\frac{0.5\Delta x}{a}$

TABLE 2.1. Parameters used for the self-similar propagation of a rotated ellipse

Figure 2.10 shows the numerical solution in red and the exact solution in blue at different times.

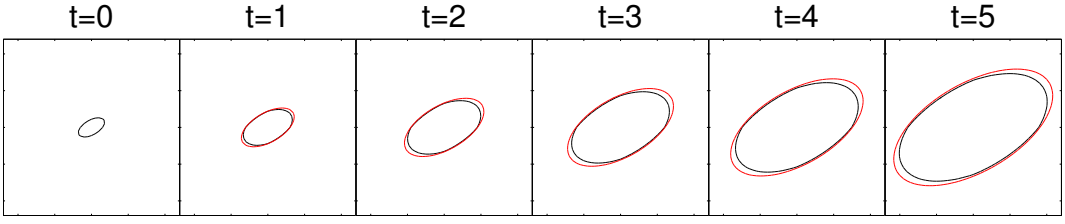


FIGURE 2.10. Numerical solution in red and exact solution in blue for the case of the self-similar propagation of a rotated ellipse

The exact solution considered is an implicit representation for an ellipse which grows in a self-similar fashion. Table 2.2 gives the errors measured for different time and space steps, for a final time $t = 1$. The orders of convergence are

1.04, 1.03 and 0.99 for the L_1, L_2 and L_∞ norms. The expected order of convergence for a general Hamilton-Jacobi equation solved with a monotone scheme is $\frac{1}{2}$(Crandall and Lions, 1984) (Zhao, 2004). The order observed here is close to 1, so it is higher than the value predicted by the theory. The difference between this value and the theoretical value can be explained by the facts we are using first order operators for the derivatives and that we are not considering the whole domain. The error was only measured at points satisfying $|\phi| < 0.5$ since we only want to compare the position of the zero level-set of the numerical solution to the zero level-set of the implicit representation of the exact solution. The consideration of the whole domain (with a singularity in the derivatives of the solution) would lower the experimental order of convergence.

Δx	Δt	L_2	L_1	L_∞
0.125	0.03125	0.16386	0.63101	0.084863
0.0625	0.015625	0.07902	0.29793	0.04281
0.03125	0.0078125	0.038837	0.14536	0.021554
0.015625	0.0039062	0.019255	0.071832	0.010812

TABLE 2.2. Absolute errors in different norms for different time and space steps for the case of self-similar propagation of a rotated ellipse on a horizontal plane

2.4.2.2. *Self-similar propagation of an ellipse on an inclined plate with advection*

Consider the self-similar propagation of an elliptical flame front on the plane

$$z = x + 2y$$

The exact solution is

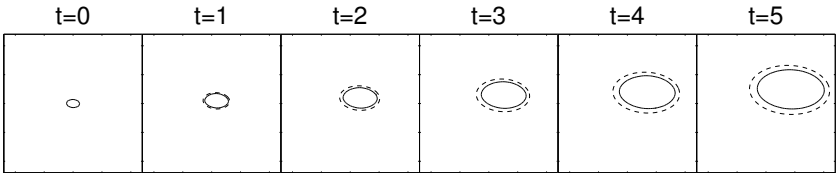
$$\phi = \sqrt{\frac{\hat{x}^2}{b^2} + \frac{(\hat{y} - ct)^2}{a^2}} - 1 - t.$$

where $\hat{x}$ and $\hat{y}$ are coordinates aligned with the inclined plate. The parameters used are given in table 2.3 :

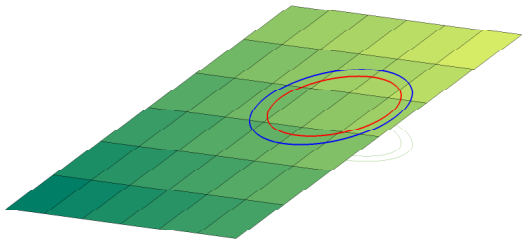
Parameters	Value
a	4
b	1
c	1
θ	$\frac{\pi}{3}$
ϕ_{init}	$\phi = \sqrt{\frac{\hat{x}^2}{b^2} + \frac{\hat{y}^2}{a^2}} - 1$
t_{final}	5
Δx	0.2
Δt	$\frac{0.5\Delta x}{a}$

TABLE 2.3. Parameters used in the case of the self-similar propagation of an ellipse with advection of an inclined plane

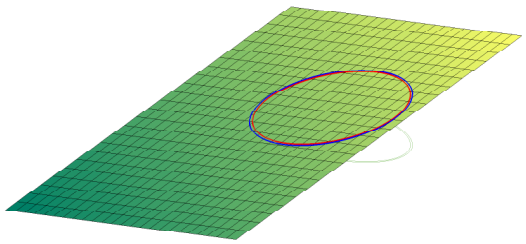
Figure 2.11 shows graphs of the numerical (red) and exact solutions (blue) for different values of Δx. The numerical solution is shown to be a good approximation of the exact solution. As Δx diminishes, the approximation improves.



(a) 2D projection of the result for $\Delta x = 0.4$



(b) Three-dimensional view of the solution at the final time for $\Delta x = 0.4$



(c) Three-dimensional view of the solution at the final time for $\Delta x = 0.1$

FIGURE 2.11. Numerical solution in red and exact solution in blue for the case of the self-similar propagation of an ellipse on an inclined plane with advection

Table 2.4 gives the errors measured for different time and space steps. The orders of convergence are 1.07, 1.04 and 1.00 for the L_1,L_2 and L_∞ norms respectively. The methodology used here is the same as in the previous case.

Δx	Δt	L_2	L_1	L_∞
0.125	0.015625	0.23604	0.89345	0.12675
0.0625	0.0078125	0.11172	0.40766	0.063968
0.03125	0.0039062	0.054441	0.19566	0.031933
0.015625	0.0019531	0.026886	0.09609	0.016014

TABLE 2.4. Absolute errors in different norms for different time and space steps for the case of self-similar propagation of a rotated ellipse on an inclined plane

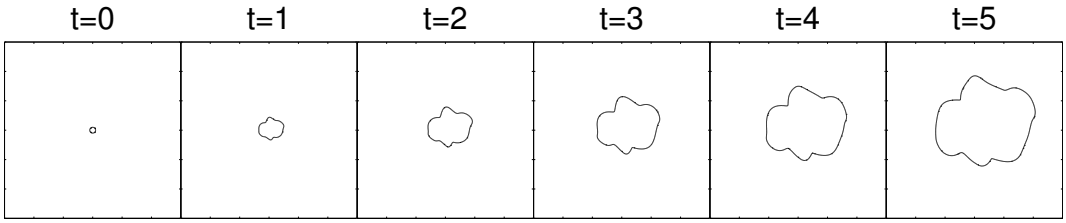
2.4.2.3. *A case where the exact solution is not known*

In this section, the numerical solution for the propagation of a flame front on the topography $z = \sin(x/4)\cos(x/4)$ will be shown. The parameters used are given in table 2.5. The choice for the topography and for the a, b and c parameters is arbitrary.

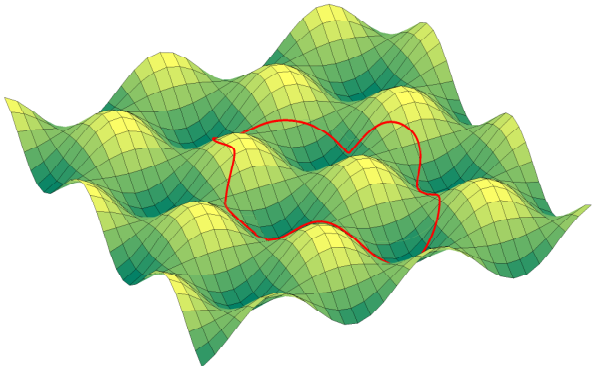
Parameters	Value
a	$2\lvert\cos\arctan\frac{y}{x+\epsilon}\rvert + 2$
b	$3\lvert\sin 3\arctan\frac{y}{x+\epsilon}\rvert + 0.5$
c	1
θ	$\frac{\pi}{3}$
ϕ_{init}	$\phi = \sqrt{\frac{\hat{x}^2}{b^2} + \frac{\hat{y}^2}{a^2}} - 1$
t_{final}	5
Δx	0.2
Δt	$\frac{0.5\Delta x}{\max(a,b,c)}$

TABLE 2.5. Parameters used in the case of propagation on $z = \sin(x/4)\cos(x/4)$

The numerical solution for the case $\Delta x = 0.2$ is shown in figure 2.12.



(a) Result example for $\Delta x = 0.2$



(b) Three-dimensional view of the solution at the final time

FIGURE 2.12. Numerical solution for the propagation of a flame front on $z = \sin(x/4)\cos(x/4)$

In this case, the exact solution is not known but the convergence of the scheme can still be verified by comparing solutions obtained with different space steps to a reference solution obtained with a smaller space step. For the different space steps used, the error on the numerical solution $\phi_{\Delta x}$ can be calculated with formula $err = \|\phi_{ref} - \phi_{\Delta x}\|$ where we take either the L_2, L_1 or L_∞ norms. Table 2.6 shows how the error diminishes along with the space and time steps, for solutions of the previous problem at final time 2. The orders of convergence are 1.36, 1.27 and 1.06 for the L_1, L_2 and L_∞ norms when the reference solution is computed with $\Delta x = 0.05$. So, this numerical experiment shows the numerical solution converges in a case where the exact solution is not known.

Δx	L_2	L_1	L_∞
0.125	1.0286	9.946	0.36291
0.0625	0.48817	4.424	0.20143
0.03125	0.17596	1.5069	0.083135

TABLE 2.6. Measured error for different time and space steps for the case of propagation on a curvy surface

2.5. A test-case with data from Prometheus

In this last section, we show how to force the level-set function to remain sufficiently smooth as time goes and a way to reduce the calculation time of the level-set method is introduced. Finally, all the techniques described in this paper are applied to a realistic example.

2.5.1. Reinitialization

When the fire spread rates are highly heterogeneous, the level-set function can become irregular. This can be a problem since irregularity in this function can render the approximation of the derivatives erroneous. This is why a reinitialization process must be used.

Rouy and Tourin (1992) and Sussman et al. (1994) have shown that for a given surface $\phi_0(x, y)$, there exists an equation whose solution at the stationary state is the signed distance function to the 0 level curve of $\phi_0(x, y)$.

$$\frac{\partial \phi}{\partial \tau} + sign(\phi_0)(\parallel \nabla \phi \parallel -1) = 0 \tag{2.5.1}$$

where $sign(\phi_0)$ is 1 where ϕ_0 is positive, -1 where it is negative and 0 où where $\phi_0 = 0$.

Solving numerically this equation after every iteration of fire propagation ensures the numerical solution remains sufficiently smooth at all times, even if the propagation parameters are very heterogeneous. Note that this reinitialization process does not move the zero level curve.

2.5.2. Narrow-band

One of the disadvantages of the level-set method with respect to the marker method is the calculation time. For a marker method, this cost is proportional to the length of the flame front. For level-set methods, the calculation time grows with the domain size, since it requires a calculation to be performed at every point of the discretization grid. For the forest fire application, domain sizes can be very large and the level-set method can be costly. Fortunately, there exists a way to reduce the number of points where calculations must be made and it is possible to implement a level-set method where the calculation time grows with the size of the flame front.

Since only the position of the zero level curve is of interest, it is possible to only update ϕ at points close to this curve. The reinitialization gives us a criteria for finding those points. If it is used, ϕ is always close to a signed distance function. This is the narrow-band method (Chopp, 1993).

At every iteration, do :

(1) Choose the points where the equation is solved. Take points (x, y) that are less than k grid points away from the front :

$$\{(x, y) : |\phi(x, y)| \leq k\Delta x\} \tag{2.5.2}$$

where k is an integer.

(2) Solve the forest fire propagation equation at the chosen point

(3) Choose a band width for reinitialization. This band must include the computation band. One possible choice is $\{(x, y) : |\phi(x, y)| \leq 2kh\}$

(4) Set $\phi(x, y) = (\pm 1.1)2kh$ at points outside the reinitialization band (take $+$ if (x, y) is unburnt and $-$ if (x, y) is burnt).

(5) Add to the reinitialization band all neighbors of points in the reinitialization band. The ensures the band grows at every iteration.

(6) Reinitialize the level-set function at points in the reinitialization band.

For example, consider the flower-shaped front of figure 2.13.

The points inside the band where the propagation equation is solved are shown in blue and the points where the reinitialization equation is solved are shown in green.

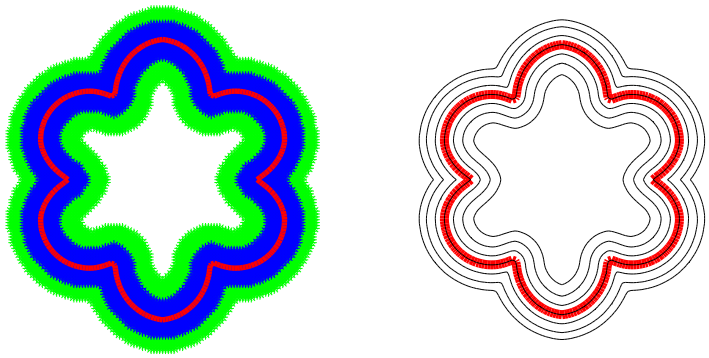


FIGURE 2.13. Points inside the band for the propagation equation in blue and points in the reinitialization band in green

2.5.3. Algorithm

Given the data for the elevation and for the other propagation parameters, the algorithm for a simulation with realistic data is the same algorithm used for idealized data, plus two post-processing steps and with restriction of the calculation domain to a narrow band :

(1) Reinitialize the level-set function to a signed distance function at every iteration with the process described above.

(2) In the case of realistic data, some grid points can be marked as points where the fire cannot cross. We make sure the flame front does not go over obstacles by setting the propagation speeds at 0 at those points. Also, we force ϕ to be positive there by setting $\phi = \max(\phi, \Delta x)$ on those points.

2.5.4. Description of the data used

In this section, results of the application of the algorithm to realistic data are shown. The propagation parameters are taken from Prometheus' dogrib2 example and they are updated hourly. Figure 2.14 shows the values of the a parameter at the initial time, in meters per minute. This figure also shows in black points where the fire can't burn and vectors showing the direction of the field governing the advection $(c \sin\theta, c \cos\theta)$. The values of the b and c parameters vary from 0 to 6.9 and 0 to 43.7, respectively. Figure 2.14 gives an idea of the heterogeneity of the data and of the scale of the spread rates.

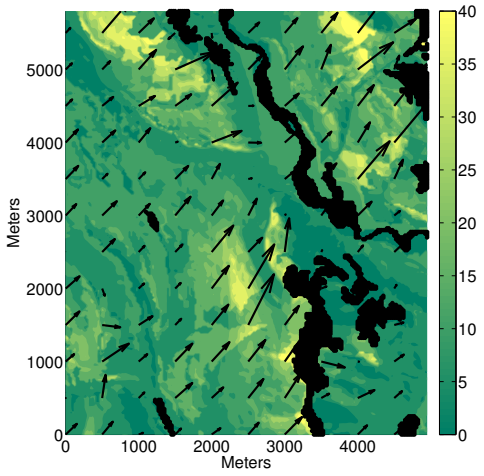


FIGURE 2.14. Map of the values of the a parameter

2.5.5. Simulation Result

Results for a simulation done with the data set described previously are shown in figure 2.15. The simulated length of the simulation is four hours and the flame fronts obtained at every 20 minutes are drawn.

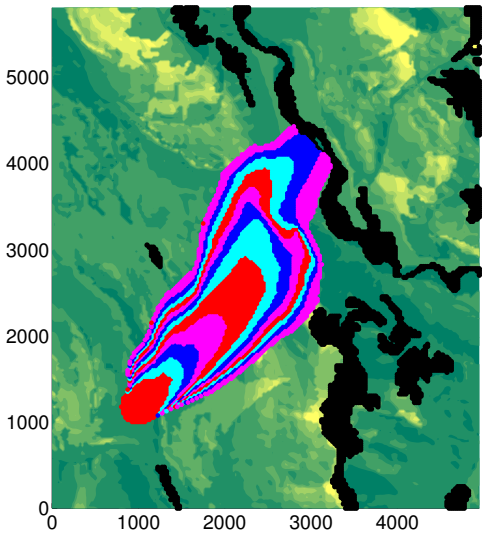
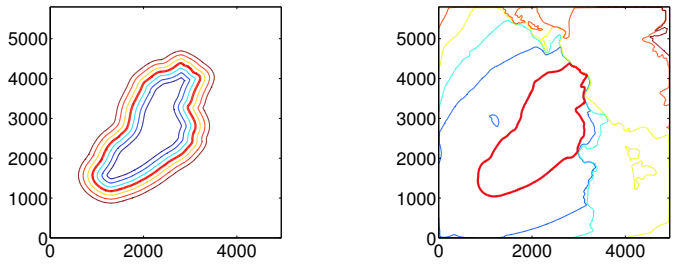


FIGURE 2.15. Result obtained by the level-set algorithm

Figure 2.16 shows the effect of using reinitialization by comparing the solution ϕ at the final time computed using reinitialization to the solution computed without reinitialization. The solution computed without reinitialization is far less smooth, which can lead to errors in the numerical solution. Also, reinitialization allowed us to only compute ϕ in a narrow-band around the front, which divided by two the computation time.

2.5.6. Treatment of unburnt pockets

The main advantage of the level-set method over the Prometheus algorithm is the treatment of flame front collisions. This subsection shows in detail an example where two fronts merge. Consider the case where a fire burns around an obstacle, as shown in figure 2.17. On the other side of this obstacle, two flame fronts will collide. The level-set method deals with this automatically.



(a) Contours when reinitialization is used

(b) Contours when reinitialization is not used

FIGURE 2.16. Illustration of the effect of using reinitialization on the contours of the solution in a realistic test case

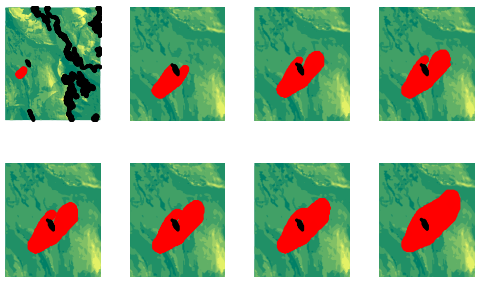


FIGURE 2.17. Firespread around an obstacle with 2 flame fronts merging

2.6. CONCLUSION

An algorithm for firespread simulation based on Richards ellipse model and on the level-set method was presented. The main advantage of the level-set method over other existing techniques such as the marker method is the automatic treatment of front collisions. Numerical examples in this paper showed how the propagation of a flame front around an obstacle, which results in two fronts merging, is dealt with automatically with no need to reconnect markers using complicated geometric procedures.

The algorithm is based on the numerical solution of a Hamilton-Jacobi partial differential equation. The computation of this solution required a new numerical scheme which was described in detail. This numerical scheme is based on rigorous mathematical principles, so the numerical solution converges to the exact solution of the problem as the resolution gets finer. The scheme was given for the case of anisotropic propagation when the anisotropy is described by an ellipse, but it could be extended to more general cases.

Test-cases where the exact solution is known showed the convergence of the numerical solution to the expected solution. Test-cases based on real data coming from Prometheus showed the applicability of the method to real-life situations. Standard tools related to the level-set method, such as reinitialization and computing the solution in a narrow-band, were used in this case. The use of the narrow-band method resulted in an algorithm with competitive computational complexity. The computation time depends linearly on the length of the flame front.

In conclusion, the combination of :

(1) the level-set method ;

(2) reinitialization ;

(3) the reduction of the computation domain to a narrow-band around the flame front ;

(4) adequate treatment of obstacles ;

(5) a rigorous numerical scheme for the solution of the level-set equation arising from Richards' ellipse model ;

resulted in an algorithm which could be used in real-life situations for the computation of firespread.

2.7. Acknowledgments

I would like to thank my PHD advisor, professor Anne Bourlioux, for many fruitful discussions about the contents of this paper and for her financial support.

Chapitre 3

SCHÉMA SEMI-IMPLICITE POUR LA RÉSOLUTION NUMÉRIQUE D'ÉQUATIONS DE TYPE HAMILTON-JACOBI ANISOTROPES

Ce chapitre est consitué de l'article «A semi-implicit scheme for steady anisotropic Hamilton-Jacobi partial differential equations with applications to the simulation of forest fires». Le but principal de l'article est de présenter un nouveau schéma (avec une preuve de convergence) pour la résolution numérique d'équations aux dérivées partielles de type Hamilton-Jacobi convexe et anisotrope. L'avantage principal de ce nouveau schéma est qu'il permet d'utiliser des solutions à des problèmes proches pour accélérer le temps de calcul. Les contributions principales de l'article sont :

(1) un nouveau schéma semi-implicite avec une preuve de convergence ;

(2) des exemples d'application à la simulation de feux de forêt ;

(3) l'illustration du fait qu'on peut réutiliser des solutions à des problèmes proches comme condition initiale à la méthode pour diminuer le temps de calcul ;

(4) un exemple d'application à l'homogénéisation périodique.

A SEMI-IMPLICIT SCHEME FOR STEADY ANISOTROPIC HAMILTON-JACOBI PARTIAL DIFFERENTIAL EQUATIONS WITH APPLICATIONS TO THE SIMULATION OF FOREST FIRES

Alexandre Desfossés Foucault

Prepared for : Computers and mathematics with applications

ABSTRACT. A semi-implicit numerical method for the efficient computation of the steady-state solution of a Hamilton-Jacobi partial differential equation is presented. The motivation comes from the modelling of forest fires as propagating fronts. The main advantage of the method presented here is that it allows the use of previously computed solutions for nearby problems as initial guesses which can lead to significant speed up. Some applications are presented. The first one is to find the ignition point of a fire given its front position at some later time. The second one is to study the response of the fire to random perturbations in the model data. The third one is to do an exhaustive study of the influence of a parameter on solutions. The last one is periodic homogenization.

3.1. Introduction

Hamilton-Jacobi partial differential equations arise in many different fields such as seismic imaging, image processing and front tracking. Many numerical methods for solving such equations already exist. The first one introduced in the literature was the level-set method (Osher and Sethian, 1988). The main idea of the method is to use an implicit representation $\phi(x,t)$ of an interface $\Gamma(x,t)$ and to compute the movement of this interface by solving equation (3.1.1).

$$\begin{cases} \phi_t + H(x, \nabla\phi) & = 0. \\ \phi(x,0) & = 0 \text{ on } \Gamma. \end{cases} \tag{3.1.1}$$

In Osher and Sethian (1988), the authors discuss valid numerical schemes for this type of partial differential equation. The convergence of numerical solutions to this problem was further studied in the 80's and 90's. In Crandall and Lions (1983), the authors define the class of solutions of (3.1.1) called viscosity solutions. This was necessary because solutions to (3.1.1) may not be differentiable everywhere. Then, a proof for the convergence to viscosity solutions of monotone, consistent and stable schemes was given (Crandall and Lions, 1984; Barles and Souganidis, 1991). On the numerical side, improvements were made to the level-set method

such as introducing the concept of reinitialization of ϕ to a signed distance function (Sussman et al., 1994) and restricting the computation domain to a narrow-band around the interface (Chopp, 1993).

Another type of partial differential equation useful for people interested in front tracking is the time independent Hamilton-Jacobi problem (3.1.2) :

$$\begin{cases} H(x, \nabla T) & = 1, \\ T(x) & = 0 \text{ on } \Gamma. \end{cases} \tag{3.1.2}$$

The solution $T(x)$ to this problem is the arrival time at a point x. There exists many different algorithms for solving (3.1.2). One of the first ones is the fast-marching method (Sethian, 1996) which works for isotropic cases. Later on, ordered upwind methods (Sethian and Vladimirsky, 2003) and the fast-sweeping method (Zhao, 2004; Kao et al., 2003) were introduced for more general cases.

In this paper, we are interested in solving (3.1.2) in the convex anisotropic case using a semi-implicit scheme by adding a pseudo-time dependence as shown in equation (3.1.3) :

$$\begin{cases} T_t + H(x, \nabla T) & = 1, \\ T(x) & = 0 \text{ on } \Gamma. \end{cases} \tag{3.1.3}$$

Implicit methods have not often been considered before since in the level-set context, the reduced number of iterations made necessary by bigger time steps is offset by an error depending on Δt and in the time-independent context, there exist non-iterative methods. Recently, Vladimirsky and Zheng (2014) showed in some cases using implicit schemes can be advantageous, in the sense that their computational cost per accuracy can be competitive.

The main goal of our paper is to show another advantage of a semi-implicit method : the fact it allows us to use previously computed solutions to nearby problems as initial guesses to speed up the numerical computations. This would not be possible with an explicit scheme since this class of scheme must obey the CFL condition. This means the solution is computed from a boundary point, and the information propagates from this boundary point at a limited speed, with no gain given by previously existing guesses on the solution. The semi-implicit scheme presented here is not bound by this condition, hence the numerical domain of dependency of the solution at a given point is much larger than in the case of the explicit scheme. This allows quicker convergence to steady state. Other numerical methods for this class of problem were not designed to use existing guesses on the solution. For example, the fast-sweeping method must use as an

initial guess a solution which overestimates the arrival time at every point, otherwise it will not work. The fast-marching and ordered-upwind methods only take as initial input a boundary point from which information will propagate. The implicit method method presented in Vladimirsky and Zheng (2014) would also overwrite an initial guess.

Our semi-implicit scheme for (3.1.2) consists in an iterative process in which we must solve a linear system at each iteration. Using a good guess for solution T will reduce the number of iterations necessary to converge to steady state. To the best of our knowledge, no other numerical method for solving Hamilton-Jacobi partial differential equations can use information on the solution to speed up the calculation time. Also, the number of iterations required for convergence can be quite small, even in the absence of a good initial guess.

The paper is organized in the following way. Section 3.2 presents the semi-implicit scheme for a general convex anisotropic Hamiltonian. Section 3.3 proves the convergence of the scheme to the viscosity solution of the partial differential equation. Section 3.4 introduces an anisotropic application example : the simulation of forest fires. Section 3.5 shows the numerical convergence of the scheme and the fact using a good initial guess reduces the number of iterations required. Section 3.6 explains how one can obtain burn probability maps by repeatedly solving the equation while adding perturbations to the propagation speeds and using previously computed solutions as a good guess. Section 3.7 explains a way to find the ignition point of a fire by computing solutions to a sequence of propagation problems. Section 3.8 discusses the computational efficiency of the scheme. Finally, section 3.9 shows how to apply the semi-implicit scheme to the problem of homogenization.

3.2. Semi-implicit scheme

Consider the following equation :

$$T_t + H(\nabla T) = 1.$$

The semi-implicit scheme introduced in this paper will be of the form

$$T^{n+1} = T^n - \Delta t H(\nabla T^n, \nabla T^{n+1}) + \Delta t.$$

where T^n and T^{n+1} stand for the solutions at iterations n and $n+1$. The scheme will be obtained by using the discretization of the spatial derivatives dictated by T^n but by evaluating parts of the Hamiltonian at iteration $n+1$.

Begin by using the Bellman formula for the Hamiltonian

$$H(\nabla T) = \max_{\alpha\in[0,2\pi]} (T_x \cos\alpha + T_y \sin\alpha) f(\alpha),$$

where $f(\alpha)$ is the speed profile of a particle on the front.

Kao et al. (2003) show the numerical version of this formula given by (3.2.1) is a Godunov Hamiltonian.

$$\hat{H} = \max_{\alpha\in[0,2\pi]} \left[\cos\alpha^+(T_x^-) + \cos\alpha^-(T_x^+) + \sin\alpha^+(T_y^-) + \sin\alpha^-(T_y^+)\right] f(\alpha) \quad (3.2.1)$$

We will explain later how to compute the maximizer α_{max} at every point of the domain to obtain the Godunov Hamiltonian at iteration n :

$$\hat{H} = (T_x^n \cos\alpha_{max}^n + T_y^n \sin\alpha_{max}^n) f(\alpha_{max}^n). \quad (3.2.2)$$

The next step is to evaluate parts of the Hamiltonian at different iterations as given by

$$H(\nabla T^n, \nabla T^{n+1}) = (T_x^{n+1} \cos\alpha_{max}^n + T_y^{n+1} \sin\alpha_{max}^n) f(\alpha_{max}^n).$$

This equation is linear in the unknowns evaluated at iteration $n+1$. This gives a linear system of equations to be solved at each time step.

To obtain the Godunov Hamiltonian, we follow the steps given in Foucault (2015a). We start by considering the numerical version of the Bellman formula (3.2.1) and by dividing the interval $[0, 2\pi]$ in four quadrants. We look for potential maximizers in all the quadrants and along the axes. For example, consider the first quadrant. The Bellman formula restricted to this quadrant is given by

$$\hat{H} = \max_{\alpha\in(0,\frac{\pi}{2})} \left[\cos\alpha(T_x^-) + \sin\alpha(T_y^-)\right] f(\alpha).$$

Note that everything is evaluated at iteration n. The maximizer angle α_{max}^{NE} must be found. This angle is the angle of the characteristics of the equation. For the isotropic case, where $f(\alpha) = 1$, $\alpha_{max} = \arctan \frac{\phi_y^-}{\phi_x^-}$. In the first quadrant, this angle will depend on ϕ_x^- and on ϕ_y^-.

The candidate for this quadrant, given by (3.2.3), can now be written down.

$$H^{NE} = (T_x^- \cos\alpha_{max}^{NE} + T_y^- \sin\alpha_{max}^{NE}) f(\alpha_{max}^{NE}) A^{NE}, \quad (3.2.3)$$

where A^{NE} is equal to 1 if the angle α_{max}^{NE} is in the right quadrant and is equal to 0 if it is not.

The next possible candidate is $\alpha = \frac{\pi}{2}$. The corresponding value for the Hamiltonian is given by (3.2.4).

$$H^N = T_y^- f\left(\frac{\pi}{2}\right) A^N \quad (3.2.4)$$

where A^N equals 1 if $T_y^- > 0$, and is 0 otherwise. Proceeding in the same way for the 6 other candidates and taking the candidate maximizing the Hamiltonian gives the correct numerical Hamiltonian.

The semi-implicit scheme based on this choice of derivatives can now be written. With the maximizer angle known for every point in the domain, the time-split equation becomes

$$\frac{T^{n+1} - T^n}{\Delta t} + (T_x^{n+1} \cos \alpha_{max}^n + T_y^{n+1} \sin \alpha_{max}^n) f(\alpha_{max}^n) = 1.$$

Define for every point $U_x^-, U_x^+, U_y^-, U_y^+$, variables equal to 1 if the corresponding derivative approximation must be used and equal to 0 if not. With this notation and after discretizing the spatial derivatives at time $n+1$, the equation is given by

$$\begin{aligned}\frac{T^{n+1} - T^n}{\Delta t} + (&T_x^{n+1,-} U_x^- \cos \alpha_{max}^n + T_x^{n+1,+} U_x^+ \cos \alpha_{max}^n + \\ &T_y^{n+1,-} U_y^- \sin \alpha_{max}^n + T_y^{n+1,+} U_y^+ \sin \alpha_{max}^n) f(\alpha_{max}^n) = 1.\end{aligned}$$

Let us call g_1 the coefficient in front of the terms containing the x derivative and g_2 the one in front of the y derivative terms.

$$\begin{aligned} g_1 &= \cos \alpha_{max}^n f(\alpha_{max}^n), \\ g_2 &= \sin \alpha_{max}^n f(\alpha_{max}^n). \end{aligned}$$

After re-arranging the terms, the following equation is obtained :

$$\begin{aligned}T^{n+1} + \Delta t(&T_x^{n+1,-} U_x^- g_1 + T_x^{n+1,+} U_x^+ g_1 + \\ &T_y^{n+1,-} U_y^- g_2 + T_y^{n+1,+} U_y^+ g_2) = \Delta t + T^n.\end{aligned}$$

By putting the unknowns T^{n+1} in a vector, the scheme can be given in matrix form as shown by

$$G^n T^{n+1} = \Delta t + T^n. \tag{3.2.5}$$

Line i of matrix G^n, corresponding to the ith unknown is given by equation

$$\begin{aligned}G_i^n = I_i + \Delta t((g_1)_i (U_x^-)_i (D_x^-)_i + (g_1)_i (U_x^+)_i (D_x^+)_i) + \\ \Delta t((g_2)_i (U_y^-)_i (D_y^-)_i + (g_2)_i (U_y^+)_i (D_y^+)_i),\end{aligned}$$

where I_i is line i of the identity matrix. Note that the line j corresponding to the ignition point has zeroes everywhere except for a 1 at element j. This choice, coupled with putting the right hand side to 0 at this point, allows us to deal correctly with the ignition point.

For boundary conditions, set the propagation speed to 0 on the boundary and give large values to the solution at those points.

3.3. Convergence of the scheme

To prove the convergence of the scheme, the results of Barles and Souganidis (Barles and Souganidis, 1991) will be used. Monotone, consistent and stable approximation schemes for a degenerate elliptic partial differential equation converge to the unique viscosity solution.

Definition 3.3.1. *A scheme is* ***consistent*** *if in the limit* $\Delta x \to 0$ *and* $\Delta t \to 0$*, the scheme converges to the equation.*

Definition 3.3.2. *A scheme* $T_{ij}^{n+1} = F(T_{ij}, T_{i\pm k_1, j\pm k_2})$ *with* k_1 *and* k_2 *integers chosen to cover all considered grid points is* ***monotone*** *if* F *is a non-decreasing function of every argument (Oberman, 2006).*

Definition 3.3.3. *A scheme is* ***stable*** *if the numerical solution* $T^n(\Delta x, \Delta t)$ *has a bound independent of* Δx *(Barles and Souganidis, 1991).*

Proposition 3.3.1. *The scheme is consistent.*

Proof. The scheme was obtained by discretizing the spatial and temporal derivatives and by evaluating parts of the Hamiltonian at different iterations. This spatial discretization was shown to be consistent in Kao et al. (2003). Also, as Δt goes to 0, the approximation of the time derivative converges and $H(\nabla T^n, \nabla T^{n+1})$ converges to $H(\nabla T)$. □

Lemma 3.3.1. *Elements on the main diagonal of matrix* G^n *are always positive. Elements elsewhere are always negative. This remains true at every iteration.*

Proof. Line i of the matrix is

$$\begin{aligned} G_i^n = I_i &+ \Delta t (g_1)_i (U_x^-)_i (D_x^-)_i \\ &+ \Delta t (g_1)_i (U_x^+)_i (D_x^+)_i \\ &+ \Delta t (g_2)_i (U_y^-)_i (D_y^-)_i \\ &+ \Delta t (g_2)_i (U_y^+)_i (D_y^+)_i \end{aligned}$$

Every line i of the matrix will have non-zero elements at a maximum of 3 positions. One of the non-zero elements will be the coefficient G_{ii}^n. Another non-zero element will come from the x derivative. It will be either the coefficient G_{ii-N}^n or the coefficient G_{ii+N}^n with N the number of points along the y dimension. Only one of those coefficients can be non-zero since the scheme doesn't allow the use of both backward and forward approximations at any one point. Off-diagonal

coefficients coming from the x derivative will always be negative since if the backward approximation must be used, the off-diagonal coefficient coming from D_x^- is negative and coefficient g_1 is positive. If the forward derivative must be used, the off-diagonal coefficient coming from D_x^+ is positive but then g_1 is negative. Finally, the last non-zero coefficient comes from the y derivative. It will be either G^n_{ii+1} or G^n_{ii-1} and it is also always negative. Table 3.1 lists all possible cases. It can be seen that no matter what the choice of the derivatives is, all the coefficients on the main diagonal are always positive and those elsewhere are always negative. □

Lemma 3.3.2. *The matrix G^n is strictly diagonally dominant.*

Proof. As shown by Table 3.1, $|G^n_{ii}| > |G^n_{ii-N}| + |G^n_{ii+N}| + |G^n_{ii-1}| + |G^n_{ii+1}|$ for every i, so the matrix is always strictly diagonally dominant. □

Lemma 3.3.3. *The matrix G^n is non-singular*

Proof. Since G^n is strictly diagonally dominant, it is non-singular. □

Proposition 3.3.2. *The scheme is monotone.*

Proof. The scheme can be written as $T^{n+1} = F^n(T^n)$ where $F^n(T^n) = (G^n)^{-1}(T^n + \Delta t)$. Since that eigenvalues of G^n are all positive (applying Gershgorin's circle theorem shows the eigenvalues are all bigger than 1) and that G^n has positive elements on its main diagonal and negative elements elsewhere, G^n is an M-matrix. The inverse of an M-matrix has only positive coefficients (Fiedler and Ptak, 1962), so F^n is a non-decreasing function of all its arguments and thus F^n is monotone. So, the scheme is monotone. □

Proposition 3.3.3. *The scheme is unconditionally stable.*

Proof. Consider F^n the mapping describing the scheme. This mapping commutes with the addition of a constant vector c. This can be shown using the fact $c = G^n c$ since matrix G^n is the identity matrix, to which we add terms depending only on the derivatives.

$$\begin{aligned} F^n(T^n + c) &= (G^n)^{-1}(T^n + c + \Delta t) \\ &= (G^n)^{-1}(T^n + G^n c + \Delta t) \\ &= (G^n)^{-1}(T^n + \Delta t) + c \\ &= F^n(T^n) + c \end{aligned}$$

This implies the mapping is non-expansive in the l_∞ norm (Crandall and Tartar, 1979; Crandall and Lions, 1984), so the scheme is stable (Crandall and Lions,

Choice of the derivatives	Sign of g_1	Sign of $(D_x)_{ii}$	G^n_{ii-N}	G^n_{ii+N}	Sign of g_2	Sign of $(D_y)_{ii}$	G^n_{ii-1}	G^n_{ii+1}	G^n_{ii}
T_x^- and T_y^-	+	+	$-\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$	0	+	+	$-\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$	0	$1+\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}+\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$
T_x^+ and T_y^-	−	−	0	$-\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$	+	+	$-\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$	0	$1+\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}+\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$
T_x^- and T_y^+	+	+	$-\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$	0	-	-	0	$-\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$	$1+\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}+\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$
T_x^+ and T_y^+	−	−	0	$-\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$	-	-	0	$-\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$	$1+\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}+\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$
T_x^- and 0	+	+	$-\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$	0	0	0	0	0	$1+\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$
T_x^+ and 0	−	−	0	$-\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$	0	0	0	0	$1+\Delta t\frac{\lvert g_{1,i}\rvert}{\Delta x}$
0 and T_y^+	0	0	0	0	-	-	0	$-\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$	$1+\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$
0 and T_y^-	0	0	0	0	+	+	$-\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$	0	$1+\Delta t\frac{\lvert g_{2,i}\rvert}{\Delta y}$

TABLE 3.1. Coefficients of line i of matrix G^n

1984; Oberman, 2006). Since no restriction was introduced on Δt, the scheme is unconditionally stable. □

Theorem 3.3.1. *The scheme converges to the viscosity solution of the partial differential equation.*

Proof. Since the scheme is monotone, stable and consistent (by Propositions 3.3.1, 3.3.2 and 3.3.3), it converges to the viscosity solution of the equation. □

3.4. Application of the scheme to flame front propagation

The physical model for firespread used here will be Richards' ellipse model (Richards, 1994). It is based on Huygens' principle. Consider a given flame front at time Δt, the flame front at the next instant $t+\Delta t$ can be obtained by assuming every point on the flame front is an ignition point of a new fire and that these fires grow like ellipses aligned with the wind, as shown on Figure 3.1.

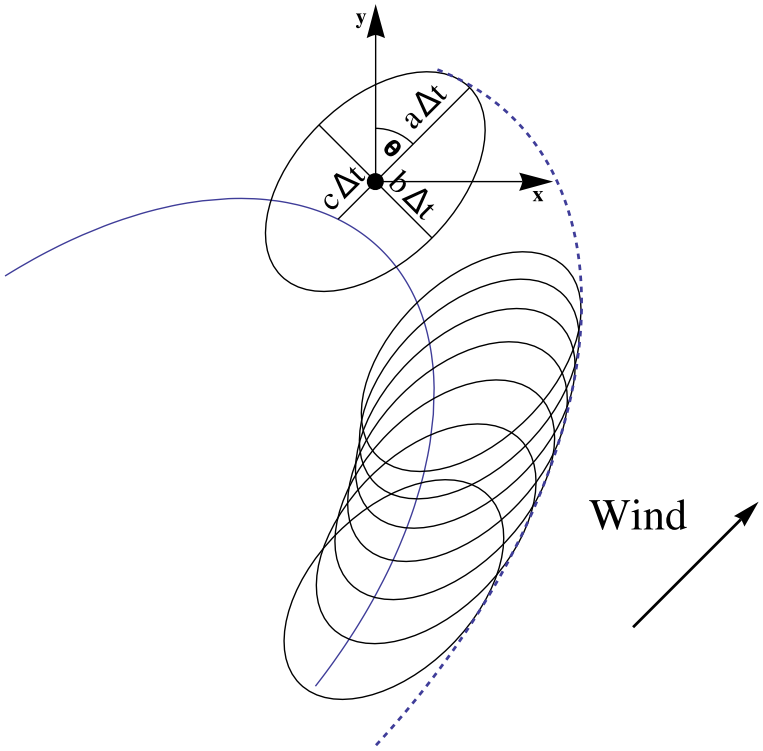


Figure 3.1. Illustration of Richards' model

The ellipses are also transported in the direction of the wind. Parameters a and b give the axes of the ellipse. Parameter c gives the magnitude of the advection in direction θ. As explained in Foucault (2015a), the forest fire propagation using this model can be simulated with a level-set partial differential equation. The corresponding arrival time equation is

$$T_t + \sqrt{k_1 T_x^2 + k_2 T_y^2 - 2k_3 T_x T_y} + (c\sin\theta, c\cos\theta)\cdot\nabla T = 1 \tag{3.4.1}$$

k_1, k_2 and k_3 are given as follows :

$$k_1 = (b^2\cos^2\theta + a^2\sin^2\theta)$$
$$k_2 = (b^2\sin^2\theta + a^2\cos^2\theta)$$
$$k_3 = (b^2\sin\theta\cos\theta - a^2\sin\theta\cos\theta)$$

The angle θ is the angle measured clockwise from the y-axis, using the notation from Prometheus (Tymstra et al., 2010). The model assumes that fires grow outwards, so $a > c$. With this condition, the Hamiltonian can be written as $H(\nabla\phi) = F(\nabla\phi)\|\nabla\phi\|$, so the Bellman formula is valid. The corresponding speed profile $f(\alpha)$ is the equation in polar coordinates of a rotated and translated ellipse :

$$f(\alpha) = \frac{\sqrt{2}\sqrt{a^2b^2\left((a^2-b^2-c^2)\cos(2(\theta+\alpha)) + a^2+b^2-c^2\right)} + 2b^2c\sin(\theta+\alpha)}{a^2 + (a-b)(a+b)\cos(2(\theta+\alpha)) + b^2} \tag{3.4.2}$$

This is valid for the propagation of a flame front in two dimensions. The case of firespread on a topography can also be formulated as a 2D propagation problem with elliptical speed profiles (Foucault, 2015a), but the ellipse coefficients are more complicated in that case. For simplicity, only the 2D case is studied in this paper.

The semi-implicit scheme for the particular case of the firespread problem can now be written down. The first thing to do is to give the Godunov Hamiltonian for this case. To obtain the candidate that maximizes the Hamiltonian in the first quadrant (refer to (3.2.3)), the direction of the characteristics of the equation is needed. It is given by :

$$\alpha_{max}^{NE} = \arctan \frac{\frac{\tilde{T}_y^{--}}{\sqrt{\tilde{\nabla T}^{--}\cdot\nabla T^{--}}} + c\cos\theta}{\frac{\tilde{T}_x^{--}}{\sqrt{\tilde{\nabla T}^{--}\cdot\nabla T^{--}}} + c\sin\theta}$$

where $\begin{pmatrix}\tilde{T}_x^{--}\\ \tilde{T}_y^{--}\end{pmatrix} = M\begin{pmatrix}T_x^-\\ T_y^-\end{pmatrix}$ and where $M = \begin{pmatrix}k_1 & -k_3\\ -k_3 & k_2\end{pmatrix}$. Plugging this α_{max}^{NE} in (3.2.3) gives the first maximizer candidate for the Hamiltonian. After simplification :

$$H^{NE} = (\sqrt{T_x^-\tilde{T}_x^{--} + T_y^-\tilde{T}_y^{--}} + (c\sin\theta, c\cos\theta)\cdot(T_x^-, T_y^-))A^{NE}.$$

The next candidate, the one where $\alpha = \frac{\pi}{2}$, is given by (3.2.4) where f is the elliptical speed profile. For the 6 other candidates, the procedure is the same.

The coefficients g_1 and g_2 can now also be given explicitly. At points where backward derivatives must be used in both x and y, the coefficients become :

$$g_1 = \frac{\tilde{T}_x^{--}}{\sqrt{T_x^-\tilde{T}_x^{--} + T_y^-\tilde{T}_y^{--}}} + c\sin\theta,$$

$$g_2 = \frac{\tilde{T}_y^{--}}{\sqrt{T_x^-\tilde{T}_x^{--} + T_y^-\tilde{T}_y^{--}}} + c\cos\theta.$$

3.5. Using the scheme to solve arrival time problems

3.5.1. A rotated ellipse

The goal of this section is to show that the scheme can solve equation (3.4.1). This test case corresponds to the one where the values are given in Table 3.2 for the ellipse parameters a, b, c and θ in the particle speed formula and the origin for the ignition point. The exact solution for the arrival time is known for this case. It is the function whose level-sets are the corresponding ellipses. Define

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \frac{x\cos\theta - y\sin\theta}{b} \\ \frac{x\sin\theta + y\cos\theta}{a} \end{pmatrix}.$$

Define also

$$\hat{c} = \frac{c}{a}.$$

Then, the steady state solution to the problem is given by equation :

$$T(\hat{x},\hat{y}) = \frac{\hat{c}\hat{y} - \sqrt{-\hat{c}^2\hat{x}^2 + \hat{x}^2 + \hat{y}^2}}{\hat{c}^2 - 1}.$$

Table 3.2 gives the parameters used for this test and Figure 3.2 shows the solution.

Domain	$[-2,2]\times[-2,2]$
Δx	$\frac{1}{256}$
a	2
b	1
c	0
Ellipse angle θ	$\frac{\pi}{4}$
Δt	100
Stopping criterion tolerance	$10^{-1}\Delta x^2$
Initial guess	$\sqrt{x^2+y^2}$

Table 3.2. Parameters used for the example of section 3.5.1

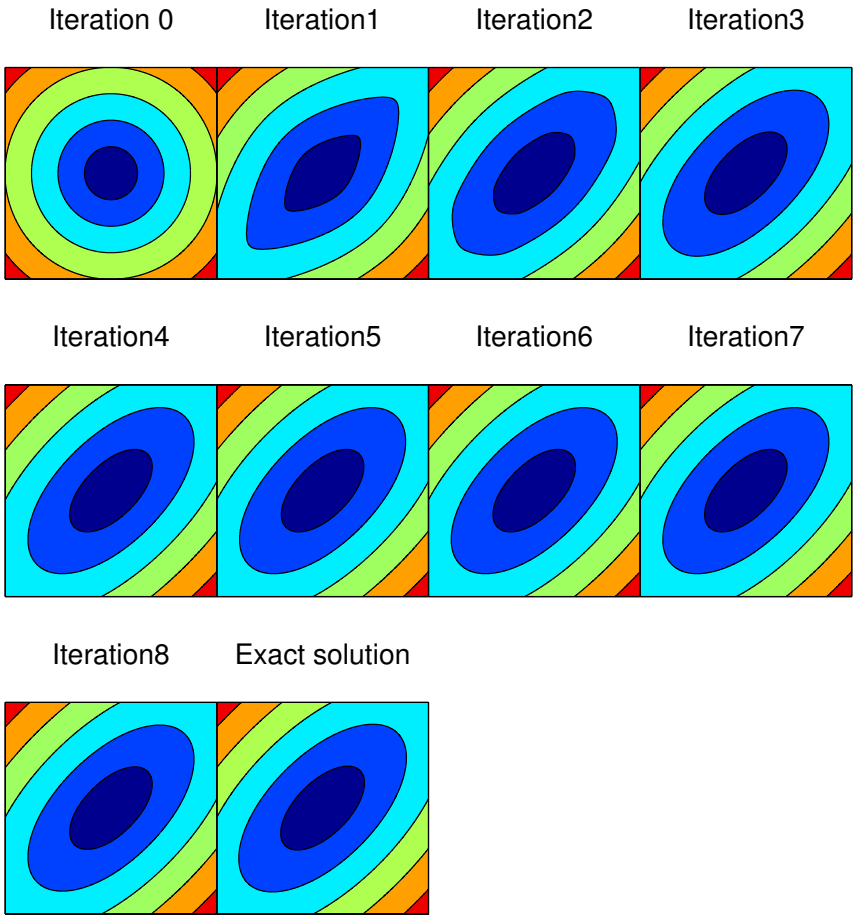


FIGURE 3.2. Solution to the test case of section 3.5.1 with advection

The solution converges in 8 iterations. Convergence Table 3.3 is obtained by varying the space step.

Δx	Δt	L_1	L_1 order	L_2	L_2 order	L_∞	L_∞ order
0.0625	100	0.5323	-	0.18929	-	0.11786	-
0.03125	100	0.32855	0.69612	0.11682	0.69629	0.07218	0.70736
0.015625	100	0.19671	0.74006	0.070143	0.73594	0.043056	0.74539
0.0078125	100	0.11517	0.7723	0.041208	0.76738	0.025136	0.77647
0.0039062	100	0.066248	0.79781	0.023778	0.79331	0.014414	0.80223

TABLE 3.3. Error obtained for different space steps for the example without advection

The order of convergence at line n is computed by taking the ratio of error values from lines n and $n-1$. The expected order of convergence is $\frac{1}{2}$ (Crandall and Lions, 1984). The values measured here indicate an order of convergence between 0.5 and 1.

3.5.2. Adding the advection term

For this test, we use the same parameters as before, except for $a = 3$, $c = 1.5$ and the initial condition is now $T(x, y) = x^4 + y^4 + x^2y^2$. Figure 3.3 shows the solution after 9 iterations.

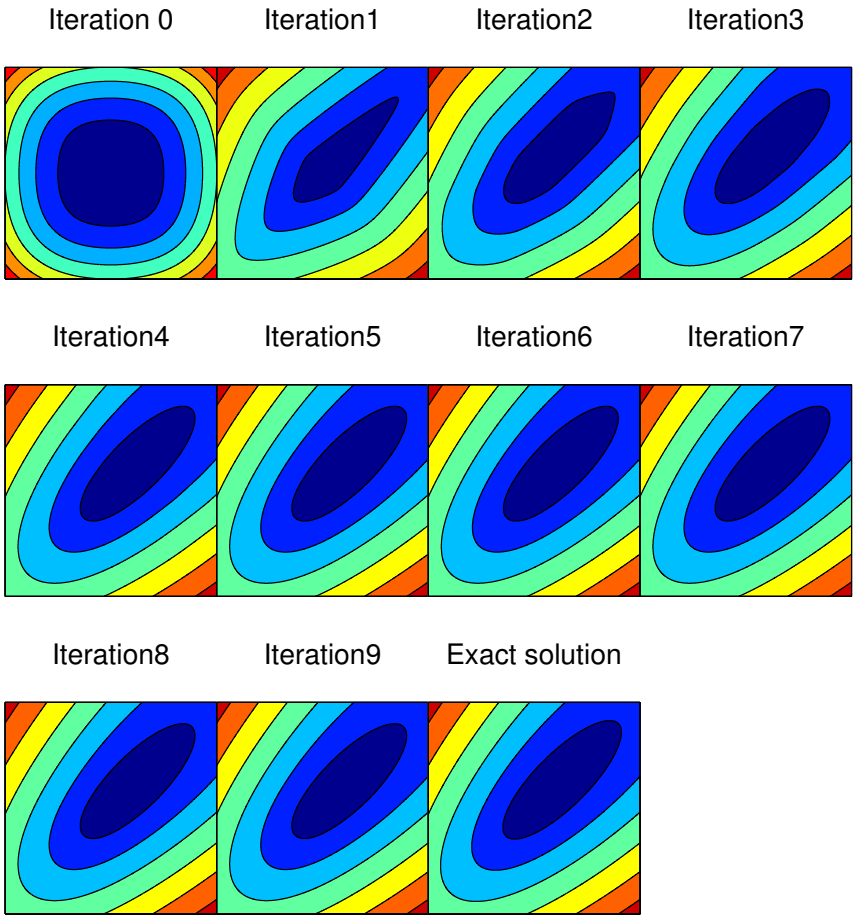


FIGURE 3.3. Solution to the test case of Section 3.5.1

Table 3.4 shows a convergence test.

Δx	Δt	error L_1	L_1 order	L_2 error	L_2 order	L_∞ order	L_∞ order
0.0625	100	0.60553	-	0.23625	-	0.15741	-
0.03125	100	0.37428	0.69409	0.14666	0.68784	0.097555	0.69027
0.015625	100	0.22475	0.73577	0.088636	0.72651	0.058845	0.72929
0.0078125	100	0.1321	0.76669	0.052421	0.75774	0.034702	0.76191
0.0039062	100	0.076315	0.79162	0.030443	0.78402	0.020077	0.78946

TABLE 3.4. Error obtained for different space steps for the example with advection

Once again, the measured order of convergence is between 0.5 and 1, as expected. So, the numerical solution converges to the exact solution as the space step diminishes.

3.5.3. Choice of the time step

To justify the choice $\Delta t = 100$, the first thing to notice is that the solution is independent of the time step. This can be seen by considering what happens at steady state in (3.2.5). If $T^{n+1} = T^n$, Δt cancels out of the equation. Also, we have counted the number of iterations required for convergence for two different space steps and for the parameter choice of the previous section. Figures 3.4 and 3.5 show the results of this test.

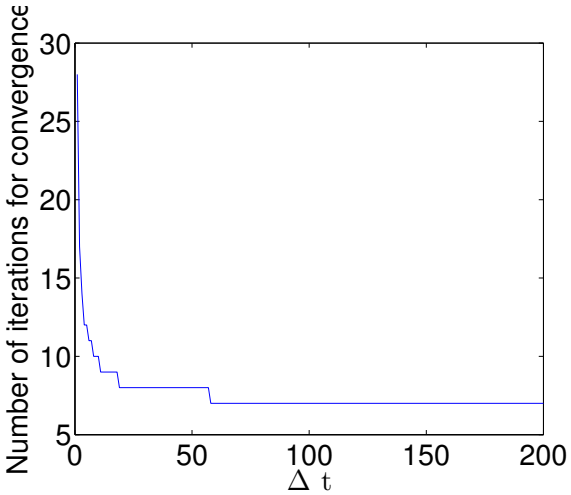


FIGURE 3.4. Numbers of iterations required for convergence as a function of the time step when $\Delta x = 0.125$

For both cases, taking Δt larger than 100 does not speed up the convergence.

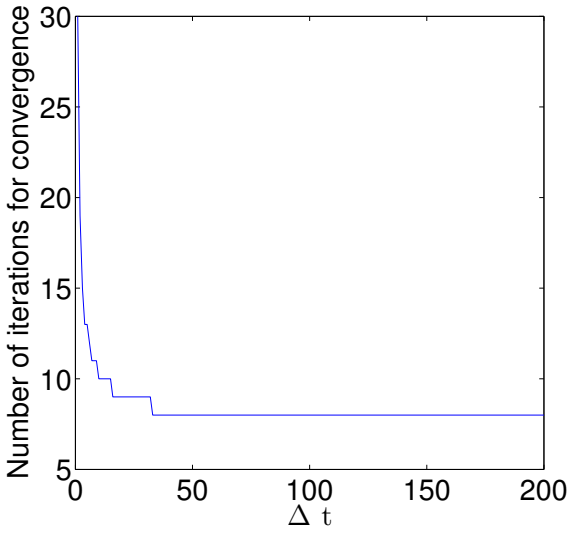


FIGURE 3.5. Numbers of iterations required for convergence as a function of the time step when $\Delta x = 0.0625$

3.5.4. Choice of the initial guess

The point of this section is to show that starting from a better guess speeds up the convergence. Consider the test of the previous section. Using the cone $T(x, y) = \sqrt{x^2 + y^2}$ as an initial condition ("bad guess"), the solution converged in 7 iterations. The initial guess will now be the solution to the propagation problem with a slightly different angle ($\theta = \frac{\pi}{4} + 0.2$) ("good guess"). The solutions obtained with the two different initial guesses will be compared at every iteration by drawing error maps. The reference solution for these error maps is the converged solution to the problem with tolerance 10^{-10}. Figures 3.6 and 3.7 show maps of the log of the error at each iteration.

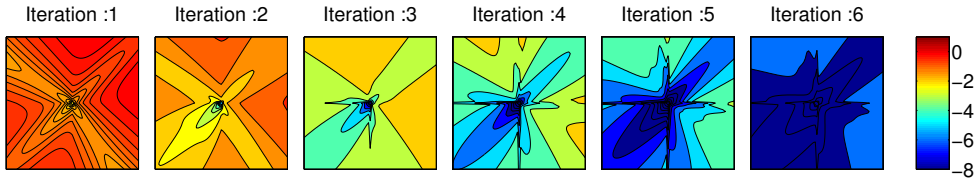


FIGURE 3.6. Log of the error at each iteration over the domain when the initial guess is bad

The number of iterations required for convergence is four when a good guess is used while it is six with a worse initial guess.

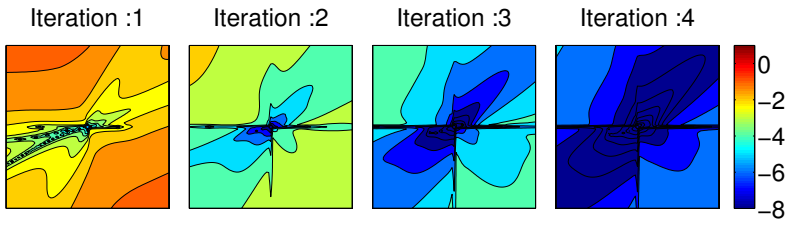


FIGURE 3.7. Log of the error at each iteration over the domain when the initial guess is good

Figure 3.8 shows the error at every iteration for the two solutions. This graph shows the convergence criteria is satisfied faster when a better initial guess is used.

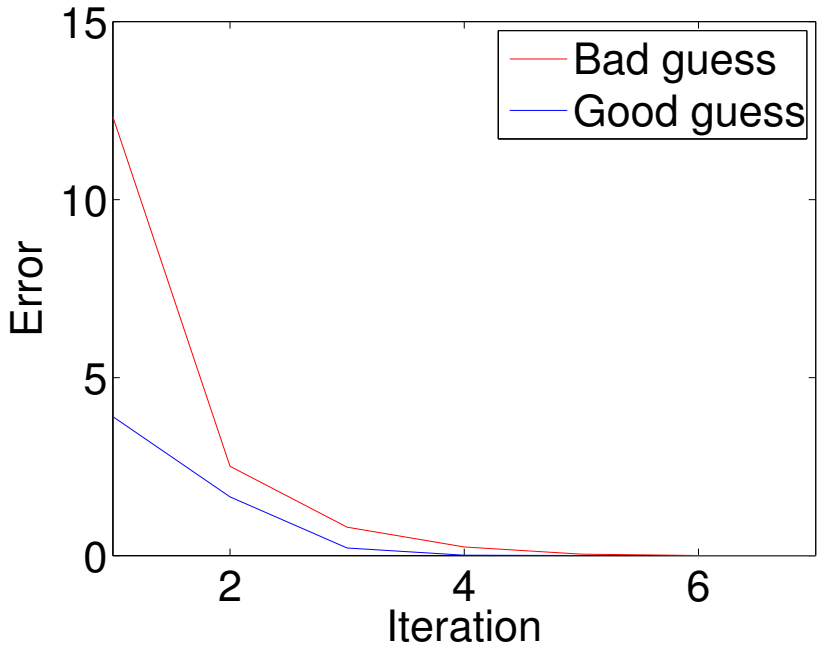


FIGURE 3.8. Error as a function of the iteration for the two different cases

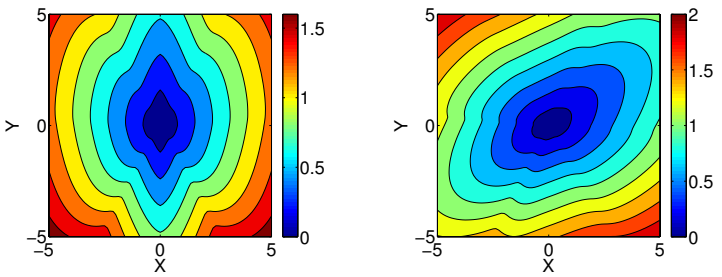
3.5.5. Computing a sequence of solutions

Another example is computing the solution to a sequence of problems while varying the wind angle from one simulation to the next. Consider the parameters given by Table 3.5.

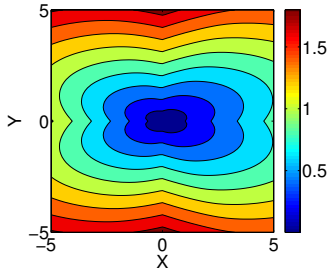
Domain	$[-5,5] \times [-5,5]$
Δx	0.125
a	$3\lvert \sin(3 \arctan \frac{y}{x}) \rvert + 4$
b	$2\lvert \cos(\arctan \frac{y}{x}) \rvert + 2$
c	1
Ellipse angle θ	Between 0 and $\frac{\pi}{2}$
Δt	100
Stopping criterion tolerance	$10^{-1}\Delta x^2$

TABLE 3.5. Parameters used when computing the sequence of solutions

The solution for 100 angles between 0 and $\frac{\pi}{2}$ is computed using the previously computed solutions as initial guesses. Figure 3.9 shows the solutions for different wind angles.



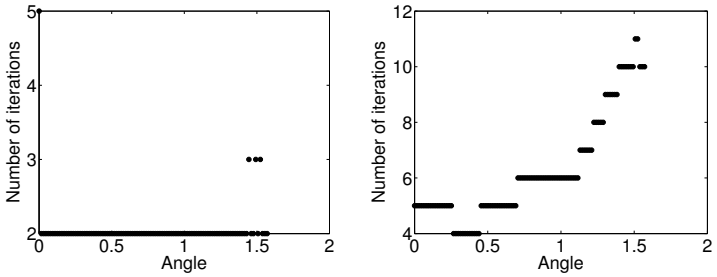
(a) $\theta = 0$ (b) $\theta = \frac{\pi}{4}$



(c) $\theta = \frac{\pi}{2}$

FIGURE 3.9. Solutions obtained by computing a sequence of solutions while varying the wind direction

By using good guesses, most solutions are computed in 2 iterations (figure 3.10). As a comparison, if the initial condition is a simple cone, around 6 iterations are needed to compute most solutions. Using a better initial guess saves calculation time. The next 2 sections show applications of this fact.



(a) Using a good guess (b) Using a bad guess

FIGURE 3.10. Number of iterations necessary for convergence if you use a good or a bad guess

3.6. Burn probability maps

A possible application of the semi-implicit scheme is to study the effect of noise on the data. One way of doing this is to compute solutions to many perturbed problems. If the perturbations $\epsilon(x, y)$ are small enough, the corresponding solutions should be close to the solution obtained with no noise. This means the noiseless solution should be a good initial guess for the next simulations. Using this approach and the parameters in Table 3.6 can reduce computation time.

Parameter	Value
Δx	0.125
Domain	$[-15, 15] \times [-15, 15]$
a	$4 + \epsilon_a(x, y)$
b	$2 + \epsilon_b(x, y)$
c	1
θ	$\frac{\pi}{4}$
Tolerance	$10^{-1}\Delta x^2$

Table 3.6. Parameters used for the burn probability map example

The burn probability map is generated using the following algorithm :

(1) Pick some values for the original ellipse parameters.

(2) For every simulation, add noise $\epsilon_a(x, y)$ to parameter a with the following procedure :
   — Define a coarse grid on the domain using a Δx 8 times as big the one used to define the domain.
   — At every point of this coarse grid, generate a random value on $[0, 1]$ using a uniform distribution.
   — Use interpolation to get the noise values at every point of the domain.
   — Add noise to the a parameter by mapping the random value between $[0, 1]$ at each point to a value between $[-0.7\bar{a}, 0.7.\bar{a}]$ where $\bar{a} = 4$.
   — Add noise to b and c in the same way, using the same random seed for all three parameters.

(3) Compute the solutions to the noisy problems using the solution obtained without noise as an initial guess.

(4) Write down which points are reached by the fire at a given time t.

(5) The proportion of points reached at time t gives us a burn probability map.

The burn probability map example obtained is shown in Figure 3.11
The front at $t = 3.5$ for the noiseless solution is shown in white.

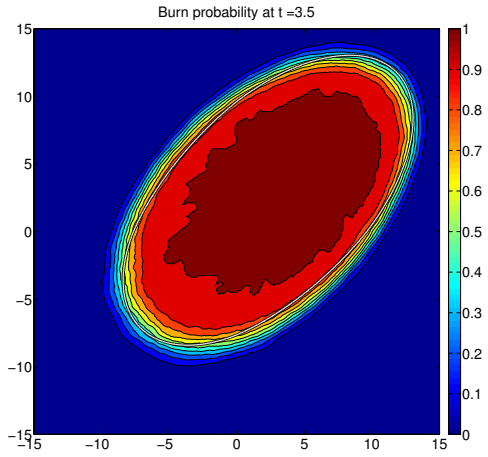


FIGURE 3.11. Burn probability map at time $t = 3.5$

Figure 3.12 shows examples of solutions to the perturbed problem.

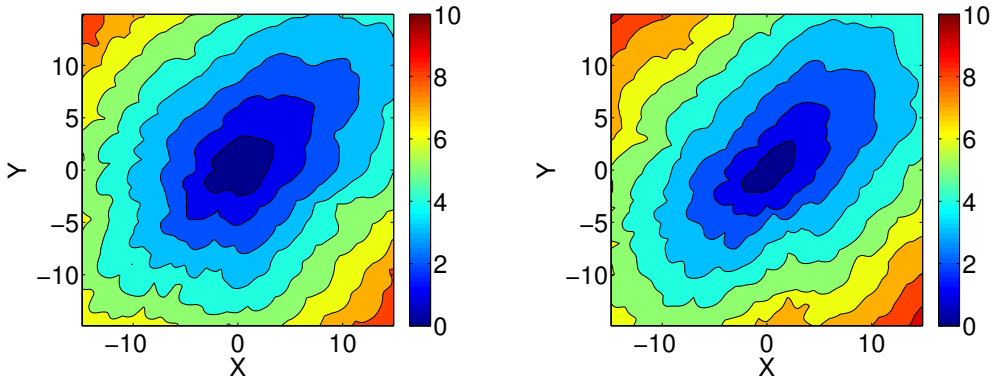


FIGURE 3.12. Two examples of perturbed solutions

Figure 3.13 shows the $T = t$ contour for the solution obtained without noise in black, for the mean of the arrival time of all problems in red and for different perturbed solutions in blue.

Figure 3.14 shows the number of iterations required for convergence for all simulations. The average value using a good guess is 22.5 and 29.3 using a bad guess.

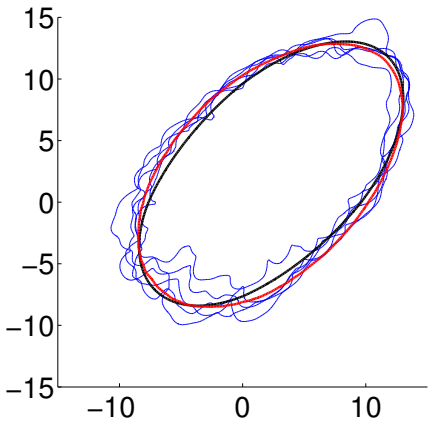
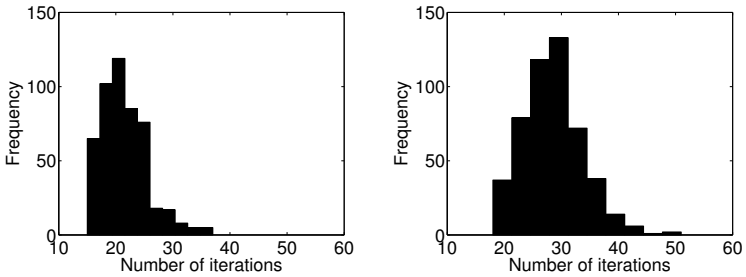


FIGURE 3.13. The $T = 3.5$ contour for different solutions



(a) (b)

FIGURE 3.14. Number of iterations needed for convergence if we use a good guess (a) and if we use a bad guess (b)

To generate a more realistic example, a simulation using the fuel map shown on Figure 3.15 was run. This map was loosely derived from data given with the Prometheus fire propagation software. For a wind speed of 20 km/h, the parameters (a, b, c) vary from $(1.6, 0.6, 1.4)$ at points shown in dark green to $(7.5, 16.7, 19.4)$ at yellow points.

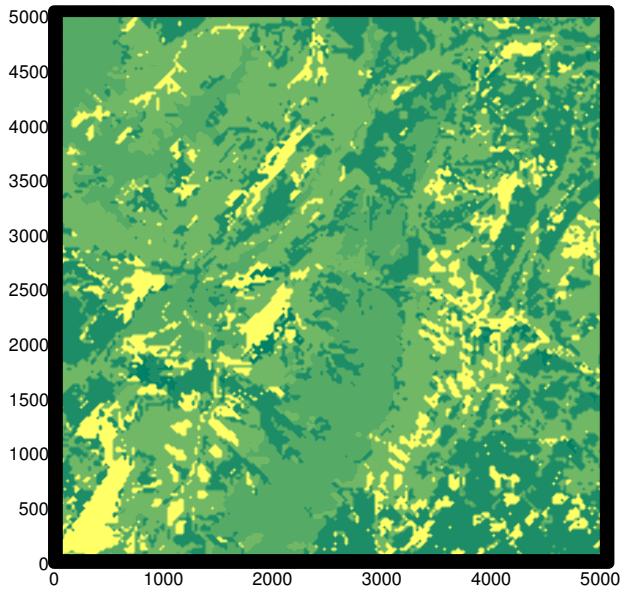


FIGURE 3.15. Fuel map for a realistic burn probability map example

One hundred simulations were run while varying the spatially homogeneous wind direction between $(\frac{\pi}{4}-0.1, \frac{\pi}{4}+0.1)$ and the wind speed between 20 km/h and 30 km/h. This variation in the wind speed is heterogenous in space. At every point of the grid, the windspeed is chosen according to a uniform distribution on [20,30]. Using these values, the corresponding a, b, c can be computed using the FBP (fire behavior prediction) model (Group, 1992). This variation in the windspeed gives values for a varying between 19.4 and 32. These parameters produce the burn probability map given by Figure 3.16. The white contour corresponds to the solution in the absence of noise (with a uniform wind speed of 25 km/h).

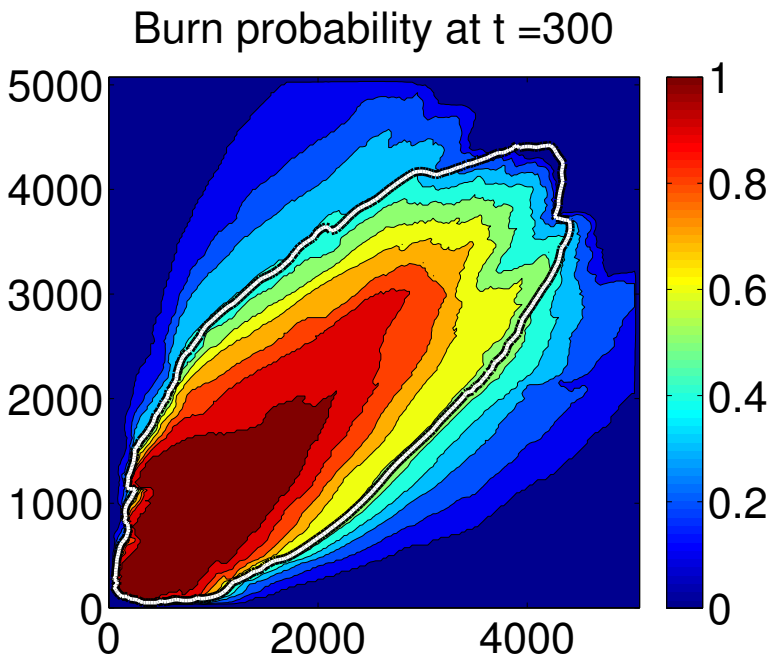
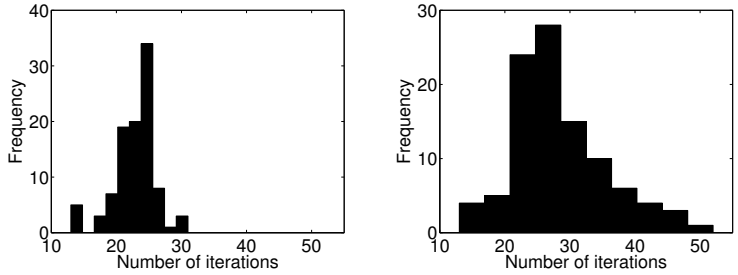


FIGURE 3.16. Burn probability map for the realistic example with 100 simulations

For this test, the initial guess is the solution computed with a wind speed of 25 km/h. Figure 3.17 shows the required number of iterations is reduced by using a good guess.



(a) (b)

FIGURE 3.17. Number of iterations needed for convergence if we use a good guess (a) and if we use a bad guess (b)

The examples of this section show the semi-implicit scheme can speed up the computation of burn probability maps by using good initial guesses.

3.7. Finding the ignition point

Another application of the semi-implicit scheme is finding the ignition point of a fire. One way of finding this ignition point would be to guess a general area where the ignition point could be, to compute the solution to the propagation problems for many ignition points in this general area and then to compare all the obtained solutions to the observed flame front. By using as initial guess solutions computed for neighboring ignition points, it is possible to save time. For example, suppose we are searching for the ignition point of the flame front shown in red on Figure 3.18 (this front was obtained by computing the arrival time from the ignition point $(x, y) = (-0.5, -1)$ with the parameters given in Table 3.7). On the same figure, the green square shows the general area where the ignition point should be.

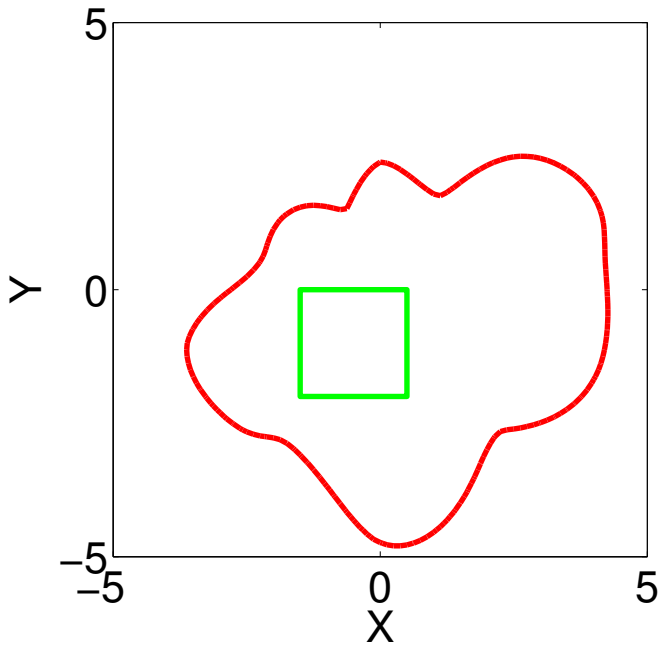


Figure 3.18. Observed flame front with a guess for the whereabouts of the ignition point

By computing the solution to the propagation from every point on the grid in the green square, we can give a score to every point p with formula :

$$Score(p) = 100 * (1 - \frac{\sigma(T_p(S))}{\max \sigma(T_p(S))}) \tag{3.7.1}$$

where $T_p(S)$ is the arrival time from point p evaluated on the observed front S and σ is the standard deviation. $\sigma(T_p(S) = 0$ if solution T_p has a contour which fits perfectly with the observed contour. This means $Score(p) = 100$ if the solution

to the propagation problem from point p fits perfectly with the observed front and $Score(p) = 0$ for the solution which gives the worst fit. We use the following algorithm. For a given flame front :

(1) choose a square area that is likely to contain its point of origin ;

(2) for every grid point in this square area, do

(a) If a solution to the propagation problem from a neighboring point is available(if we are not at the first point in the square), modify this neighboring solution by triangulation to obtain an initial guess. Suppose the arrival time from point A is known and we want to obtain a good guess for the arrival time from point B. Form a triangle between point A, point B and any other point C in the domain. The side lengths correspond to the arrival times. We know the side lengths from A to C and from A to B. Using basic trigonometry, we can obtain a side length for the side from B to C.

(b) Solve the equation to obtain the arrival time using the previously described initial guess.

(c) Assign a score to the point with formula 3.7.1.

Parameter	Value
Δx	0.0625
Domain	$[-5, 5] \times [-5, 5]$
b	$3\lvert \sin(3 \arctan(\frac{y}{x})\rvert + 0.5$
a	$2\lvert \cos \arctan(\frac{y}{x}\rvert + 2$
c	$0.8\lvert \cos \arctan(\frac{y}{x}\rvert + 0.5$
windDirection	$\pi/3 + 0.4 \cos(x)$

TABLE 3.7. Parameters used in the first ignition point search example

Figure 3.19 shows the score for every point in the green square. The point with the highest score is as expected the real ignition point of the observed flame front.

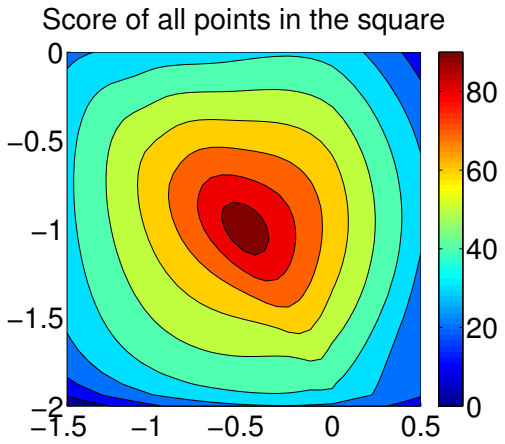
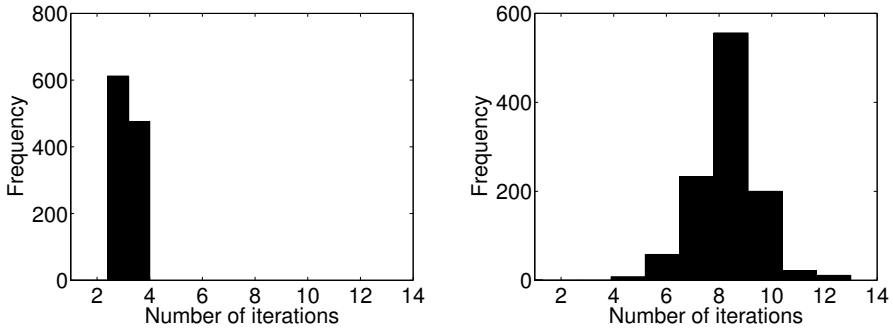


FIGURE 3.19. Score given to every point in the square

Figure 3.20 shows the gain in iterations obtained by using a good guess.



(a) Number of iterations required when using a good guess

(b) Number of iterations required when using a bad guess

FIGURE 3.20. Number of iterations required for convergence when computing the propagation from different ignition points when using different initial conditions

The next example show the application of the ignition point search algorithm to realistic data. The fuel map used is obtained in a similar fashion to the one of the previous section. The flame front of interest is shown in red in Figure 3.21. It started from point $(1000, 1000)$.

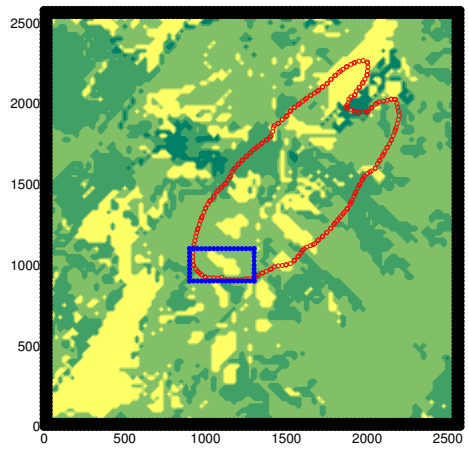


FIGURE 3.21. Fuel map and flame front of interest

By computing the solutions to the propagation problem for all ignition points in the square, the ignition point can be found, as shown in Figure 3.22. Figure 3.23 shows the reduction in iterations obtained by using good initial conditions.

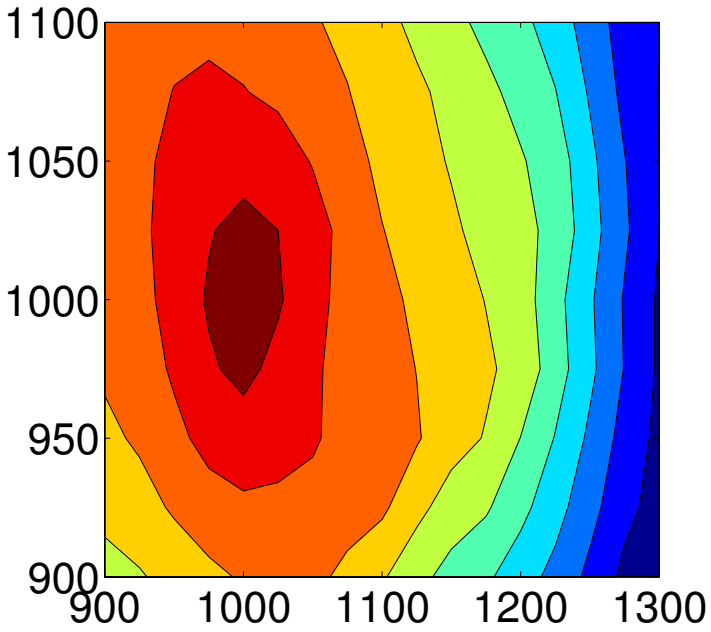
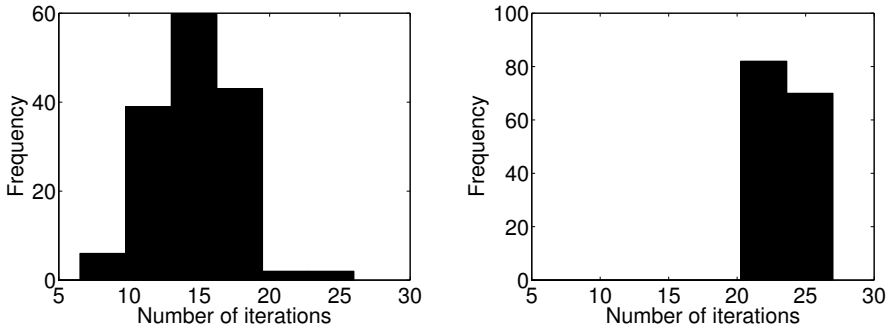


FIGURE 3.22. Score given to all points in the rectangle



(a) Number of iterations required when using a good guess

(b) Number of iterations required when using a bad guess

FIGURE 3.23. Number of iterations required for convergence when computing the propagation from different ignition points when using different initial conditions

3.8. Computational efficiency of the scheme

To give an idea of the computational cost of the algorithm, the Matlab profiler (Mathworks, 2015-03-14) was run on a test case for three different domain sizes. The computer used has core I7 960 processor (3.2 GHz) with 6 GBs of RAM. The parameters used for this test are given in table 3.8.

Domain	$[-2,2]\times[-2,2]$
Δx	$\frac{1}{64}, \frac{1}{128}, \frac{1}{256}$
a	3
b	1
c	1.5
Ellipse angle θ	$\frac{\pi}{4}$
Δt	100
Stopping criterion tolerance	Δx^2

TABLE 3.8. Parameters used in the first computational cost test

For all cases, the initial guess is the cone $T = \sqrt{x^2+y^2}$. The computation times for the different parts of the algorithm are given in Table 3.9 (in seconds). The times showed are obtained by running the code 10 times and taking the average.

The total times for the main parts of the algorithm for the different resolutions are 0.67, 3.1 and 15.13. The computation time growth as the number of points grows is superlinear, mainly driven by the fact the growth of the time required for solving the linear system is superlinear. As the number of points is multiplied by four, the time required by all parts of the algorithm except the solution of the

Task \Domain size	257×257	513×513	1025×1025
Derivation matrices	0.096	0.364	1.713
Compute Hamiltonian	0.166	0.743	3.066
Create scheme matrix	0.238	1.020	3.933
Solve Linear system	0.171	0.973	6.413

TABLE 3.9. Computation time for the test

linear system is approximately multiplied by 4. For the linear system resolution, this multiplication factor is closer to 6. In an ideal case for applying the algorithm (when using a good initial guess for the solution), the number of iterations required for convergence would be much smaller, around 2 or 3, which would considerably diminish the time necessary for convergence.

An improvement which could be made to the algorithm to decrease computation time is to replace the direct method for the resolution of the linear system by an iterative method.

3.9. Periodic homogenization

3.9.1. Presentation of the problem and algorithm

Another application of the semi-implicit scheme is the problem of periodic homogenization. Consider a Hamilton-Jacobi partial differential equation with coefficients varying periodically :

$$\phi_t + H\left(\frac{x}{\epsilon}, \nabla\phi\right) = 0, \tag{3.9.1}$$

with the dependance on $\frac{x}{\epsilon}$ standing for small-scale periodic perturbations. Under certain conditions (Xin, 1991), the solution to this problem converges to the solution of

$$\phi_t + \bar{H}(\nabla\phi) = 0 \tag{3.9.2}$$

where $\bar{H}$ is an effective Hamiltonian no longer varying on the small scale.

In Achdou et al. (2008), the authors use a semi-implicit scheme to compute $\bar{H}$ when the problem is isotropic. The scheme introduced here can generalize their approach to anisotropic cases. Define variable $y = \frac{x}{\epsilon}$. Since the period of the perturbations is ϵ, the period of variable y is 1. The effective Hamiltonian is computed by taking the limit of $w(t, y)$ as $t \to \infty$ where $w(t, y)$ is the solution of

$$w_t + H(y, \nabla w + p) = 0. \tag{3.9.3}$$

over a domain covering one period of y. Solving this problem numerically over a box of size $[0, 1]\times[0, 1]$ until a large value of t and taking the mean of function $\frac{w}{t}$ (or taking the mean of $H(y, \nabla w + p)$) over the domain gives $\bar{H}(p)$ for the given value of p. $\bar{H}(p)$ is obtained by solving the problem repeatedly for different values of p. Note that since the Hamiltonians of interest are such that $H(\lambda\nabla\phi) = |\lambda|H(\nabla\phi)$, only the values of p on a circle must be computed.

Semi-implicit schemes are a good choice for this problem, since taking big time steps allows us to reach a large final time much faster. The scheme presented earlier can be used to solve the problem. The only differences are periodic boundary conditions must be used and the argument to the Hamiltonian is $\nabla w+p$ instead of just being ∇w. Also, the right-hand side of the linear system will be different. To obtain the right-hand side, consider (3.9.3). Discretizing the time and evaluating parts of the Hamiltonian at different times gives :

$$\begin{aligned} &w^{n+1} + \Delta t\left(w_x^{n+1}\cos\theta^n f(\theta^n) + w_y^{n+1}\sin\theta^n f(\theta^n)\right) = \\ &w^n - \Delta t\left(p_x\cos\theta^n f(\theta^n) + p_y\sin\theta^n f(\theta^n)\right) \end{aligned}$$

which results in the right-hand side. One final difference between applying the scheme to homogenization and applying it to arrival-time problems is we can use as an initial guess a function w which is a solution to the problem for a different value of p.

3.9.2. Validation

To show the effectiveness of the proposed method, this section will compare solutions of (3.9.1) to a solution of (3.9.2). The example studied will be the propagation of a circle of radius 2 centered at the origin using the level-set method. The small-scale periodic variation will be given by parameters

$$\begin{aligned} a &= 4 + 3\cos\frac{2\pi x}{\epsilon}\cos\frac{2\pi y}{\epsilon}, \\ b &= 3 + 2\cos\frac{2\pi x}{\epsilon}\sin\frac{2\pi y}{\epsilon}, \\ c &= 1 + 0.75\cos\frac{2\pi x}{\epsilon}\cos\frac{2\pi y}{\epsilon}, \\ \theta &= \frac{\pi}{3}. \end{aligned}$$

The period of the perturbations is ϵ. To obtain the effective Hamiltonian, equation (3.9.3) is solved for different values of p with a resolution $\Delta y = \frac{1}{64}$. After computing $\bar{H}(p)$ for 24 values of p on the unit circle inside each quadrant, we can compute the solution to (3.9.2) using the level-set method with the following steps :

(1) compute the Legendre transform of $\bar{H}$ to obtain the homogenized speed profile $\bar{f}$;

$$\bar{f}(\alpha) = \min_{-\frac{\pi}{2} \le \nu - \alpha \le \frac{\pi}{2}} \frac{\bar{H}(\cos\nu, \sin\nu)}{\cos(\nu - \alpha)} \tag{3.9.4}$$

(2) use formula (3.2.1) to obtain a valid discretization of the spatial derivatives (solve the maximization problem numerically) ;

(3) choose Δt satisfying the CFL condition by looking at the maximal speed inside a periodic box. This maximal speed is $a + c$.

The next step is to compute the solution to (3.9.1). We will choose $\epsilon = 3$ and $\epsilon = 1.5$ with $\Delta x = \frac{\epsilon}{64}$. Figure 3.24 shows a comparison of the two high-resolution solutions with the homogenized solution and the solution without perturbations. The figure shows that as ϵ diminishes, the fit between the homogenized solution and the high resolution solution improves. Computing the homogenized solutions is much cheaper than computing the high resolution solution.

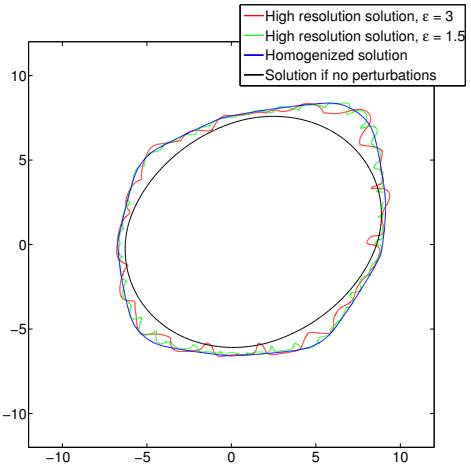


FIGURE 3.24. A comparison of two solutions to the propagation problem at time $t = 0.7$, one where the small-scale perturbations were well resolved and one where we homogenized the perturbations

3.10. CONCLUSION

A semi-implicit scheme for the computation of numerical solutions to anisotropic Hamilton-Jacobi partial differential equations was presented. The main advantage of this scheme is that it allows the use of good guesses for the expected solutions as initial conditions, which speeds up the computation time.

The scheme was shown to be monotone, consistent, stable and thus convergent. The main arguments behind the proof relied on the fact the scheme can be written as a linear system of equations. The study of the properties of the matrix underlying this linear system allowed the deduction of the properties of the scheme. No restrictions were introduced on the initial condition and on the time step, so the scheme is unconditionally stable. This unconditional stability means large time steps can be taken, which speeds up convergence and is useful in the context of homogenization.

The scheme was presented for a general convex Hamilton-Jacobi partial differential equation. It is applicable for all cases where the Bellman formula is valid. It is of particular interest for front propagation problems where the propagation speed is anisotropic. Numerical examples were shown in one such case, where the anisotropy arises from elliptical speed profiles for a particle on the flame front.

Numerical experiments showed the convergence to the exact solution with the expected order of convergence. Other numerical tests showed practical applications of the scheme. Good guesses can be used by this scheme to reduce computation time in two applications in the context of forest fire simulation. One

is a way to generate burn probability maps, and the other is a method for finding the ignition point corresponding to a given flame front.

Another application is periodic homogenization. The fact large time steps can be used is particularly useful for the computation of effective Hamiltonians in the presence of small-scale periodic perturbations. A numerical experiment showed the semi-implicit scheme can be used in computing homogenized solutions. These homogenized solutions closely fit high-resolution solutions, but are cheaper to compute.

Future work could include investigating ways of reducing the computation time of the scheme. For example, an iterative method could be used instead of a direct method for the solution of the linear system arising from the scheme.

3.11. Acknowledgments

I would like to give thanks to my PHD advisor, professor Anne Bourlioux, for discussing with me the ideas presented in this paper. This work was also financed with her research funds.

Chapitre 4

APPLICATION DE LA THÉORIE DE L'HOMOGÉNÉISATION À L'ÉTUDE DE L'EFFET DES PERTURBATIONS À PETITES ÉCHELLES DANS LA VITESSE DU VENT SUR LA VITESSE DE PROPAGATION DE FEUX DE FORÊT

Ce chapitre contient l'article «On the influence of small-scale variation in the wind speed on the propagation of forest fires». Le but principal de l'article est d'utiliser les outils des deux premiers articles pour traiter l'aspect multi-échelles du problème de propagation des feux de forêt. L'auditoire visé est la communauté scientifique intéressée par la simulation de feux de forêt. Les contributions principales de l'article sont :

(1) On montre comment utiliser l'homogénéisation dans le contexte de la simulation de feux de forêt et on documente les résultats de l'homogénéisation d'une équation aux dérivées partielles de type Hamilton-Jacobi pour un cas de propagation anisotrope.

(2) On observe les effets des perturbations à petites échelles dans l'équation de propagation d'un feu. On conclut que ces perturbations tendent à accélérer la vitesse de propagation et donc que les simulations actuelles qui ignorent ces fluctuations risquent de présenter des solutions qui sont en retard sur la réalité.

(3) On fournit une piste de réponse à une question qui a longtemps troublé la communauté scientifique qui s'intéresse aux feux de forêt. Le modèle le plus utilisé pour la propagation d'un feu de forêt est le modèle de l'ellipse, qui assume que chaque point d'un front de flammes est la source d'un

nouveau feu qui grandit comme une ellipse alignée avec le vent. Mais, des tests en laboratoire et sur le terrain tendent à montrer que cette forme n'est pas la forme appropriée dans tous les cas, en particulier si les paramètres influençant la propagation sont hétérogènes. On montre dans cet article que coupler des hétérogénéités dans les paramètres avec le modèle de l'ellipse peut donner des formes de feu qui correspondent à ce qui est observé dans la réalité.

ON THE INFLUENCE OF SMALL-SCALE VARIATION IN THE WIND SPEED ON THE PROPAGATION OF FOREST FIRES

Alexandre Desfossés Foucault

Prepared for : International Journal of Wildland Fire

ABSTRACT. The goal of this paper is to show how to apply the results of homogenization theory to deal with the multi-scale nature of the simulation of forest fires. The anisotropic firespread is computed using a Hamilton-Jacobi partial differential equation (PDE) which comes from Richards' ellipse model. The effect of small-scale perturbations on the equation coefficients can be calculated using homogenization. The study of idealized periodic perturbations shows zero-mean perturbations tend to increase the fire propagation speed and that this increase is a linear function of the amplitude of the perturbations. Another observation is that small-scale heterogeneities alter the global shape of the fire which means using the ellipse model can result in non-elliptical fire shapes such as a teardrop. The observations made on idealized perturbations result in a procedure that can be applied in practical cases.

4.1. Introduction

Wildfires cause millions of dollars worth of damage every year. One of the tools used by governments to help in managing these disasters is the numerical simulation of forest fires. Computer programs can predict where the flame fronts will be at a given time, which can help firefighting agencies decide, for example, if certain communities must be evacuated or what is the best possible action for firefighters.

One issue in simulating forest fires is dealing with the fact that this phenomenon is multi-scale. Some factors influencing the firespread such as the wind and vegetation can vary on a wide range of scales.

The goal of this paper is to study mathematically the effect of small-scale variations in the wind speed on the propagation of forest fires, when some structure on the small-scale variation is assumed. Sources of small-scale variability for the wind include boundary layer turbulence and fire-wind feedback effects. Sun et al. (2009) and Clark et al. (1996a) have shown the importance of including the small-scale effects to get more accurate results.

The test cases will consist of periodic zero-mean small-scale perturbations. The advantage of using this idealized form for the perturbations is that the theory of homogenization will allow us to quickly compute the resulting effective speeds.

Homogenization is a method used to deal with differential equations containing coefficients varying at different scales. As a first example, consider the case of front propagation in one dimension in a heterogeneous medium (Oberman et al., 2009). Suppose a propagation domain of length 1 is divided in $2N$ small intervals of length $\epsilon = \frac{1}{2N}$ where the propagation speed is alternating between 1 and 5. The time required to go through this domain is $N\epsilon + N\epsilon\frac{1}{5} = 3/5$. This means

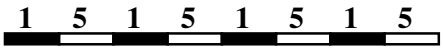


FIGURE 4.1. $1D$ propagation domain with small-scale periodic variation in the propagation speed

the effective large-scale propagation speed in this domain is $\frac{5}{3}$ and not the mean of the two speeds. Obtaining this effective speed from the coefficients varying at small scales is an example of homogenization.

Homogenization is helpful when computing the small-scale details of a system is too costly. Sometimes, these details are not essential but they cannot be neglected because they affect the macroscopic behavior. This is the case in forest fire simulation. What is needed is a global estimate of the flame front position, the small-scale shape of the front is not crucial. Homogenization allows us to include the effect of small-scale perturbations, without computing the detail of the system at that level. This greatly diminishes the calculation time.

Homogenization can be applied in many different fields. Another area of application close to the forest fire problem is combustion. Bourlioux (2002) shows how homogenization can be applied to an advection-reaction-diffusion equation used in combustion to describe the evolution of a reactive scalar T which can be seen as a mass fraction of a reacting product :

$$\frac{\partial T}{\partial t} + \left(\bar{v}(x,t) + v'\left(\frac{x}{\epsilon^\alpha}, \frac{t}{\epsilon^\alpha}\right)\right) \cdot \nabla T = \epsilon \triangle T + \frac{1}{\epsilon} f(T).$$

As ϵ goes to 0, this equation describes the position of a flame front which follows :

$$G_t = F(\nabla G),$$

where G is a function representing the flame front and F the effective normal speed of the front. F no longer varies on the small scale.

In this paper, results of periodic homogenization will be applied to an equation describing firespread according to Richards' ellipse model. Section 4.2 presents the forest fire equation. Section 4.3 explains how to homogenize this equation. Section 4.4 shows result examples when the small-scale variation of the equation coefficients is described by a checkerboard pattern. Section 4.5 presents results

of homogenization when the variations on the equations coefficients are induced by perturbations on the wind speed. Section 4.6 suggests practical ways of using homogenization results. Section 4.7 considers more realistic test cases.

4.2. Simulating firespread with Hamilton-Jacobi partial differential equations

4.2.1. Partial differential equation model and link with Richards' model

The physical model used for the firespread in this paper is Richards' model for forest fire propagation. The main idea of the model is to assume the fire spreads according to a modified Huygens' principle. Suppose each point on a flame front is a new ignition point and that all these new fires grow like translated ellipses aligned with the wind as shown by Figure 4.2. Parameters a, b, c and θ give respectively the major axis of the ellipse, the minor axis, the advection and the ellipse alignment. The position of the flame front after a time Δt is the envelope

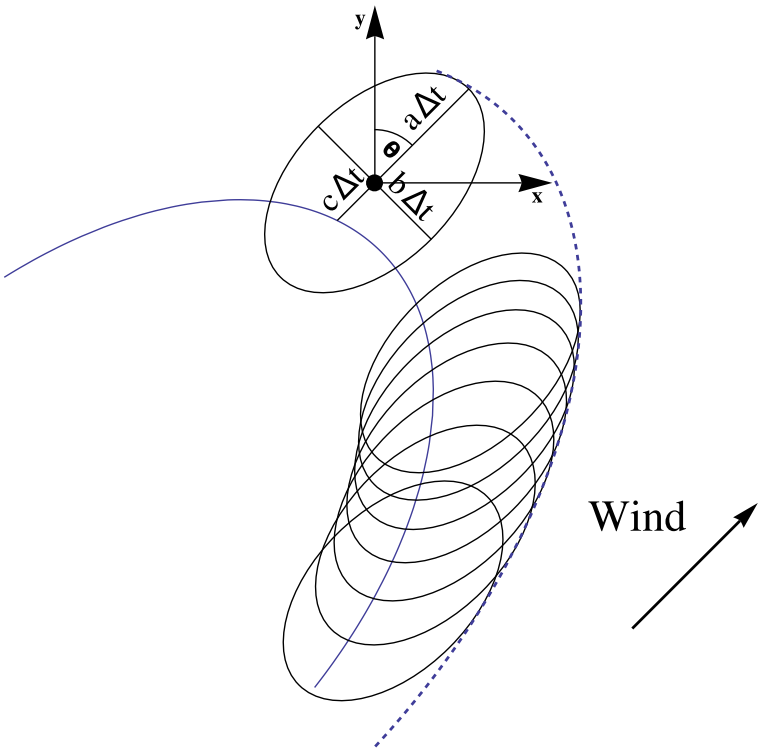


Figure 4.2. Illustration of Richards' model

of all the new fires. The rate of spread (ROS) of the fire is given by $a + c$, the back rate of spread (BROS) is $a - c$ and the flank rate of spread (FROS) is b. Foucault (2015b) shows how fire propagation using Richards' ellipse model can be computed by solving the following Hamilton-Jacobi partial differential equation :

$$\phi_t + \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y} + (c\sin\theta, c\cos\theta)\cdot\nabla\phi = 0,$$

where the Hamiltonian is

$$H(\nabla\phi) = \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y} + (c\sin\theta, c\cos\theta)\cdot\nabla\phi.$$

The k_1, k_2 and k_3 parameters in the case of propagation on a horizontal plane are given by

$$\begin{aligned} k_1 &= (b^2\cos^2\theta + a^2\sin^2\theta), \\ k_2 &= (b^2\sin^2\theta + a^2\cos^2\theta), \\ k_3 &= (b^2\sin\theta\cos\theta - a^2\sin\theta\cos\theta). \end{aligned}$$

4.3. Periodic homogenization of Hamilton-Jacobi partial differential equations and Wulff shapes

The Hamilton-Jacobi partial differential equation describing the firespread can be given in a general form as

$$\phi_t + H(\nabla\phi, \frac{x}{\epsilon}) = 0, \tag{4.3.1}$$

where the dependence on $\frac{x}{\epsilon}$ indicates a periodic dependence on the small-scale ϵ. It is known that under certain assumptions on the function $H(\nabla\phi, \frac{x}{\epsilon})$ (Xin, 1991), the solution to (4.3.1) converges to the solution of :

$$\phi_t + \bar{H}(\nabla\phi) = 0, \tag{4.3.2}$$

when ϵ goes to zero with $\bar{H}$ not varying on small-scales. This $\bar{H}$ is the effective Hamiltonian. It takes into account small-scale effects, without having small-scale variations. Foucault (2015b) shows how to compute it. Using this $\bar{H}$, solutions to propagation problems involving small scales can quickly be computed and compared to solutions to problems with no small-scale variations to see what is the global effect of the variations.

Another diagnostic that can be used to see the global effect of perturbations on the wind speed is to look at the Wulff shape of the effective Hamiltonian. The theory of Hamilton-Jacobi partial differential equations tells us that a curve moving in space under the action of a normal speed will eventually reach an asymptotic shape, if the equation coefficients are homogeneous (Osher and Merriman, 1997). For example, if a closed curve is the initial condition for the equation obtained from Richards' ellipse model, the asymptotic shape will be an ellipse aligned with the wind. This asymptotic shape is self-similar to a translated ellipse of parameters a, b, c and θ which is the Wulff shape. One way of looking at the homogenization result is to see what is the Wulff shape of the homogenized Hamiltonian. Comparing this shape to the shape associated with the equation with

no perturbations gives us a good idea of the effect of the small scale perturbations on the effective propagation speed.

The polar coordinates equation of the Wulff shape corresponding to a given Hamiltonian is given by (Osher and Merriman, 1997; Kao et al., 2003) :

$$W(\theta) = \min_{-\frac{\pi}{2} \leq \nu - \theta \leq \frac{\pi}{2}} \frac{H(\cos\nu, \sin\nu)}{\cos(\nu - \theta)}. \tag{4.3.3}$$

It is also possible to obtain the front normal speed (and thus, the Hamiltonian) from the Wulff shape (Vladimirsky, 2001). This can be used to replace the ellipse shape in Richards' ellipse model by another shape (for another approach to this problem, see Glasa and Halada (2011)). The formula giving the front normal speed at a point $\vec{x}$ from the Wulff shape $W(\vec{x}, \theta)$ is

$$F(\vec{x}) = \max_{\theta \in [0, 2\pi]} \left(\vec{n} \cdot (\cos\theta, \sin\theta) W(\vec{x}, \theta) \right),$$

where $\vec{n}$ is the normal to the front and depends on x. This formula can also be used to obtain the velocity of a particle on the front at position $\vec{x}$. Its direction is given by the maximizer angle θ_{max} in the previous formula. Its amplitude is given by the corresponding value $W(\vec{x}, \theta_{max})$. Figure 4.3 illustrates the link between the normal speed of the front and the corresponding speed profile for a particle on the front.

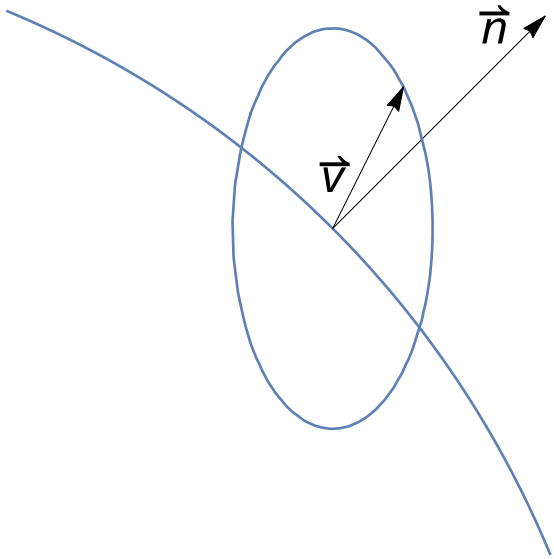


FIGURE 4.3. Normal vector $\vec{n}$ and vector $\vec{v} = (W(\vec{x}, \theta_{max}) \cos\theta_{max}, W(\vec{x}, \theta_{max}) \sin\theta_{max})$

The Wulff shape can also be used to quantify the burnt area as a function of time. Suppose a curve described by a Wulff shape $W(\nu)$ is propagating in a

self-similar fashion. The self-similar fire shape as a function of time is given by $W(\nu, t) = W(\nu)(1 + t)$. The formula giving the variation of burnt area between the current time t_1 and another time t_2 is

$$\Delta A = \int_0^{2\pi} \int_{W(\nu,t_1)}^{W(\nu,t_2)} r dr d\nu$$

Developing the right-hand side :

$$\begin{aligned}
\Delta A &= \frac{1}{2} \int_0^{2\pi} W(\nu, t_2)^2 - W(\nu, t_1)^2 dr d\nu \\
&= \frac{1}{2} \int_0^{2\pi} (1 + t_2)^2 W(\nu)^2 - (1 + t_1)^2 W(\nu)^2 dr d\nu \\
&= ((1 + t_2)^2 - (1 + t_1)^2) \int_0^{2\pi} \frac{W(\nu)^2}{2} d\nu \\
&= (1 + 2t_2 + t_2^2 - 1 - 2t_1 - t_1^2) \int_0^{2\pi} \frac{W(\nu)^2}{2} d\nu \\
&= (2(t_2 - t_1) + (t_2 - t_1)(t_1 + t_2)) \int_0^{2\pi} \frac{W(\nu)^2}{2} d\nu \\
&= (2\Delta t + \Delta t(t_1 + t_2)) \int_0^{2\pi} \frac{W(\nu)^2}{2} d\nu \\
&= \Delta t \left(1 + \frac{t_1 + t_2}{2}\right) \int_0^{2\pi} W(\nu)^2 d\nu
\end{aligned}$$

Dividing this by Δt and taking the limit as t goes to zero gives the following differential equation.

$$\frac{dA}{dt} = (1 + t) \int_0^{2\pi} W(\nu)^2 d\nu$$

which has solution

$$A(t) = (t + \frac{t^2}{2}) \int_0^{2\pi} W(\nu)^2 d\nu + A(t_0)$$

The coefficient

$$\int_0^{2\pi} W(\nu)^2 d\nu \tag{4.3.4}$$

characterizes the growth in burnt area. It will be referred to subsequently as the burnt area growth factor. For example, consider the case of an ellipse with $c = 0$. Then, $W(\nu)$ can be computed by setting $x = W \cos\nu$ and $y = W \sin\nu$ in equation $\frac{x^2}{b^2} + \frac{y^2}{a^2} = 1$ and solving for W. This gives $W(\nu)^2 = \frac{1}{\frac{\sin^2(\nu)}{a^2} + \frac{\cos^2(\nu)}{b^2}}$ and the burnt area is

$$A(t) = \left(t + \frac{t^2}{2}\right) 2\pi ab + \pi ab$$

This is the expected result since the area of an ellipse with big axis a and small axis b is πab. If it is growing self-similarly, the area as a function of time becomes $A(t) = \pi a(1+t)b(1+t)$, the same result as before.

Thus, plotting the response in $\int_0^{2\pi} W(\nu)^2 d\nu$ due to changes in the small-scale parameters is a good way to measure the effect of small-scale perturbations on the burnt area.

4.4. The checkerboard problem

As a first example of periodic homogenization, let us consider a checkerboard pattern for the periodic small-scale variations. This is an idealized representation of small-scall heterogeneity of the burning fuels. The propagation domain is shown on Figure 4.4.

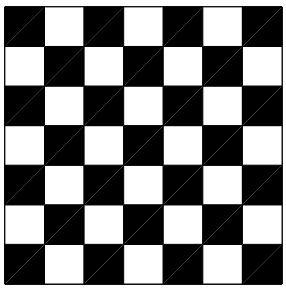


Figure 4.4. Checkerboard pattern for the small-scale perturbations

The ellipse parameters are then defined in the following way.

(1) $a =$ 1-λ or 1+ λ, depending on the position in the periodic cell

(2) $b = \gamma a$

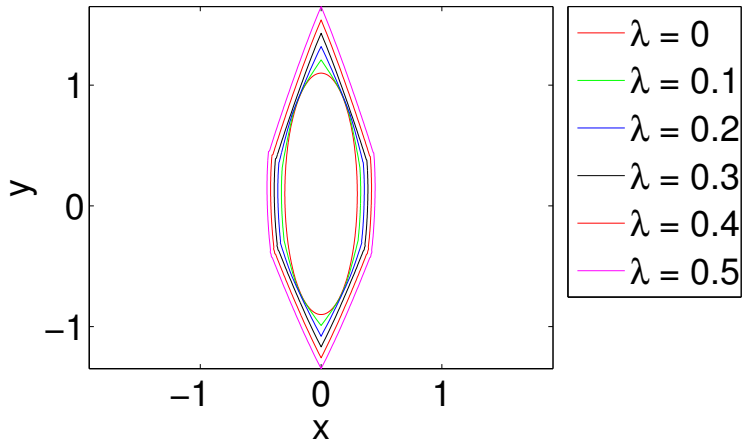
(3) $c = \alpha a$

(4) $\theta = \bar{\theta}$

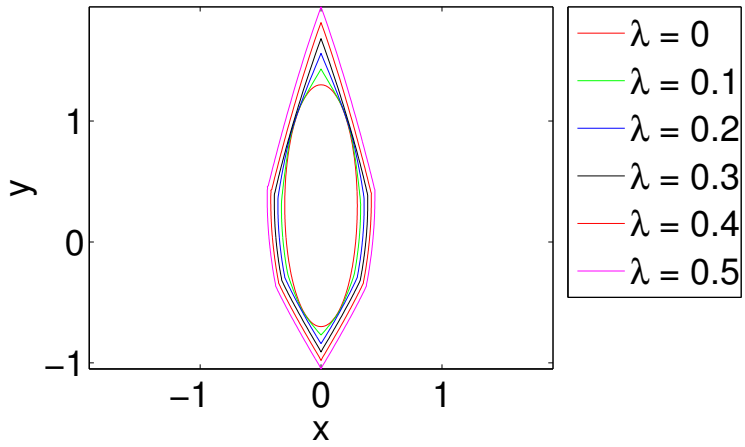
The influence of the perturbation amplitude λ, the anisotropy factor γ and the coefficient giving the advection strength α can then be observed by varying their corresponding values. λ will be chosen to vary from 0 to 0.5, γ from 0.3 to 1 and α from 0 to 0.5. The Wulff shapes are shown for these different parameter values. Figure 4.5 shows the results for $\gamma = 0.3$. Figure 4.6 shows the results for $\gamma = 0.7$. Figure 4.7 shows the results for $\gamma = 0.9$.

These tests show that zero-mean perturbations can increase the firespread rate since larger perturbations lead to larger Wulff shapes. They also show a change in fire shape resulting from small-scale perturbations. Richards' model supposes that forest fires propagate by assuming every point on a flame front is a new ignition point of a fire growing as an ellipse aligned with the wind. But this elliptical shape is maybe not the best shape. It has been observed (Tymstra et al., 2010; Green, 1983; Glasa and Halada, 2011) that other shapes could work better in certain conditions, such as in the presence of heterogeneities in the fuels. The shapes observed here are teardrop-like, as seen in Glasa and Halada (2011). This suggests the ellipse model could be valid even though other shapes are observed on a large scale since coupling elliptical firegrowth with heterogeneities results in other shapes.

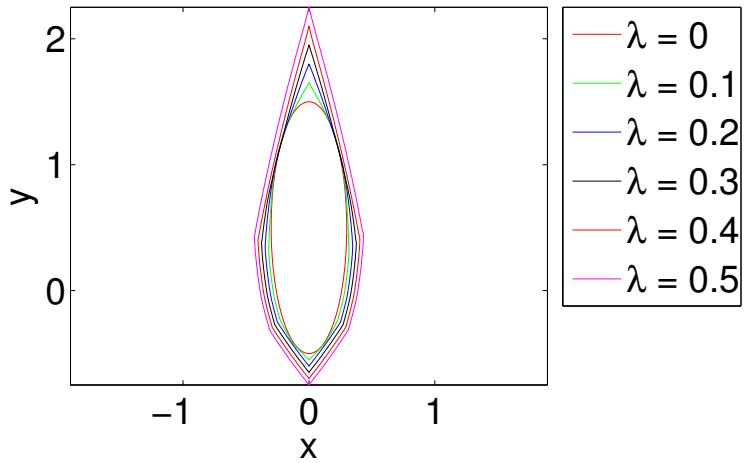
A characteristic of the resulting Wulff shapes is the presence of sharp angles. This is not due to a lack of numerical precision, this type of curve often arises in the study of homogenization of Hamilton-Jacobi partial differential equations : see for example Oberman et al. (2009).



(a) $\alpha = 0.1$,$\gamma = 0.3$, $\theta = 0$

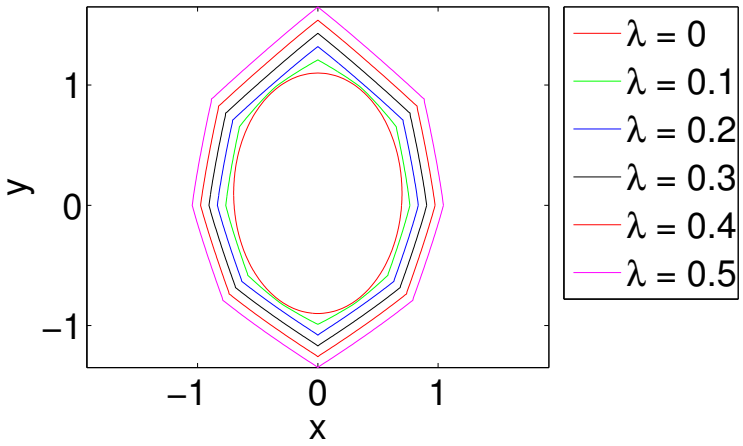


(b) $\alpha = 0.3$,$\gamma = 0.3$, $\theta = 0$

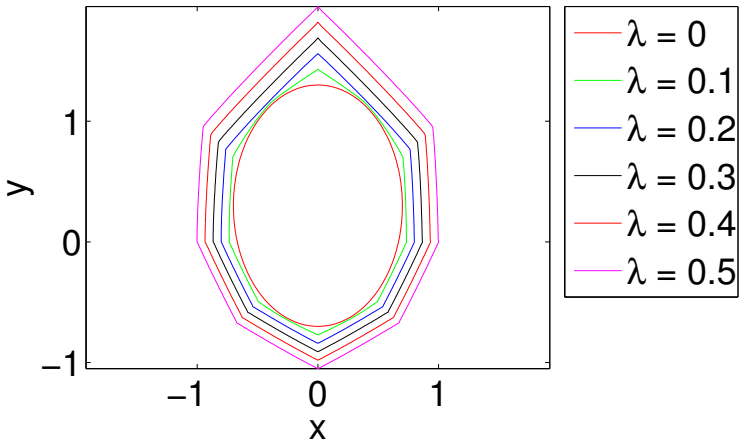


(c) $\alpha = 0.5$,$\gamma = 0.3$, $\theta = 0$

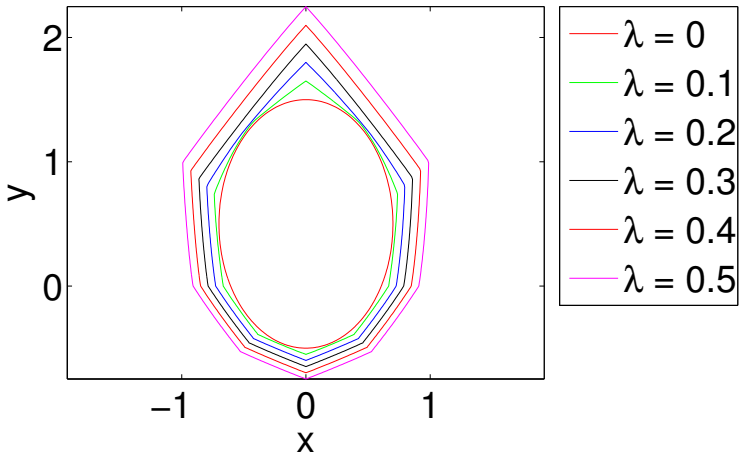
FIGURE 4.5. Wulff shapes resulting from the homogenization of the small-scale perturbations in the case $\gamma = 0.3$ for different perturbation amplitudes



(a) $\alpha = 0.1$,$\gamma = 0.7$, $\theta = 0$

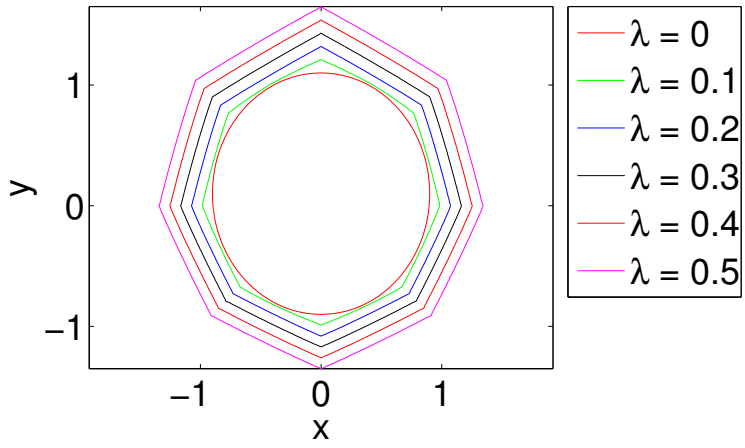


(b) $\alpha = 0.3$,$\gamma = 0.7$, $\theta = 0$

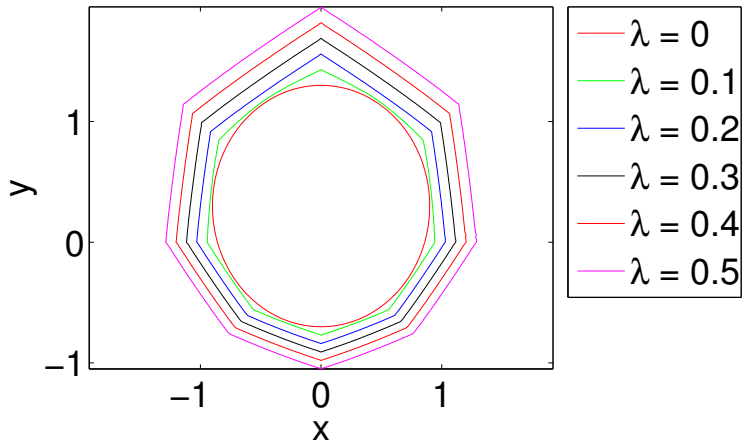


(c) $\alpha = 0.5$,$\gamma = 0.7$, $\theta = 0$

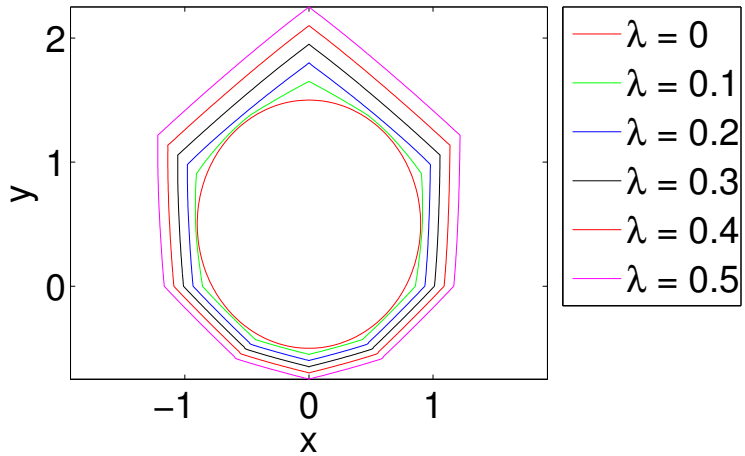
FIGURE 4.6. Wulff shapes resulting from the homogenization of the small-scale perturbations in the case $\gamma = 0.7$ for different perturbation amplitudes



(a) $\alpha = 0.1$,$\gamma = 0.9$, $\theta = 0$



(b) $\alpha = 0.3$,$\gamma = 0.9$, $\theta = 0$



(c) $\alpha = 0.5$,$\gamma = 0.9$, $\theta = 0$

FIGURE 4.7. Wulff shapes resulting from the homogenization of the small-scale perturbations in the case $\gamma = 0.9$ for different perturbation amplitudes

4.5. Periodic perturbations on the wind speed

4.5.1. Linearized model for the a, b and c coefficients

The next step is to consider more realistic perturbations. The parameters a, b, c, θ depend on many physical quantities such as the vegetation, humidity, temperature and wind. The main goal of this paper is to see the effect on fire propagation of small-scale wind variations. Periodic perturbations corresponding to three different flows will be considered. These three different flows are a shear, vortices and a mix of vortices with shears. This choice was made for many different reasons. The actual detail of the small-scale wind flow in real cases is unknown. Modeling what goes on at this level would be too complicated. But, it has been observed that fires generate vortices in the wind flow (Clark et al., 1996b). Hence, a wind flow composed of an average wind to which are added small-scale vortices and shears is an idealized representation of the actual flow.

The ellipse parameters must be computed from the wind speed at every point. To remove the necessity of using a complicated procedure (Group, 1992) to obtain them, a linearized model for the a, b, c as a function of the wind will be used. This choice was made since for the values of wind considered (between 15 km/h and 60 km/h), the variation in firespread rate is almost a linear function of the windspeed. Typical curves representing the value of these parameters as a function of wind, for fixed values of vegetation, humidity and all other pertinent parameters are shown in Figure 4.8. The equation linking the propagation speeds to these parameters was taken in Group (1992). These curves will be replaced by the following straight

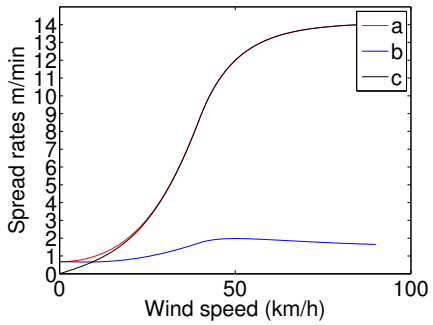


Figure 4.8. Examples of typical curves of parameters a, b, c as a function of the wind speed

lines, chosen arbitrarily in such a way as to give isotropic propagation in the absence of wind and that gives similar propagation speed when the wind is around

100

60 km/h.

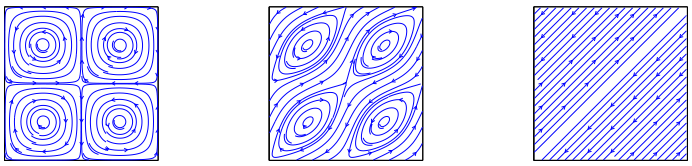
$$a_{linear} = \frac{13v}{60} + 1,$$
$$b_{linear} = \frac{v}{60} + 1,$$
$$c_{linear} = \frac{13.9v}{60}.$$

4.5.2. Presentation of the tests

The a, b, c values will be computed from the perturbed wind speed. Define $\bar{v_x}$ and $\bar{v_y}$ the average wind speeds in x and y and v_x^ϵ and v_y^ϵ the perturbations on these speeds.

$$\begin{aligned}
\bar{v} &= \sqrt{\bar{v_x}^2 + \bar{v_y}^2}, \\
v_x^\epsilon &= \bar{v_x} - \lambda\bar{v}(\cos(2\pi Y)\sin(2\pi X) - \delta\cos(2\pi X)\sin(2\pi Y)), \\
v_y^\epsilon &= \bar{v_y} + \bar{v}\lambda(-\delta\cos(2\pi Y)\sin(\pi X) + \cos(2\pi X)\sin(\pi Y)).
\end{aligned}$$

This is the Childess-Soward flow (Khouider, 2002). Modifying λ changes the amplitude of the perturbations. Changing the δ parameter changes the wind flow type. For example, $\delta = 0$ gives vortices, $\delta = 1$ a shear and $\delta = 0.5$ a mix of vortices and shears as shown on Figure 4.9. The next section shows the effect of



(a) $\delta = 0$ (b) $\delta = 0.5$ (c) $\delta = 1$

FIGURE 4.9. Streamlines of the perturbations generated by the Childess-Soward flow for different values of δ

changing the perturbation amplitudes for these different flow types. Conclusions about the effect of periodic perturbations in the forest fire propagation model will be drawn by looking at the following graphs (for different values of λ, δ, and for different orientations of the average wind $\bar{\theta}$) :

(1) a figure showing the streamlines of the perturbation field for $\lambda = 0.5$ in a periodic box ;

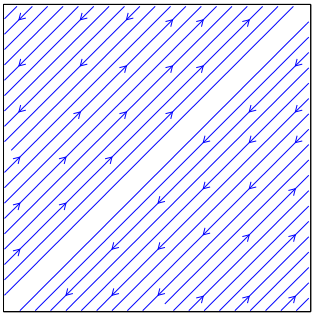
(2) a figure showing the complete wind field (average wind plus small-scale perturbations) in a periodic box ;

(3) a figure showing the Wulff shapes (equation (4.3.3)) for $\lambda = 0, 0.25, 0.5$ where the units of the axes correspond to effective propagation speeds ;

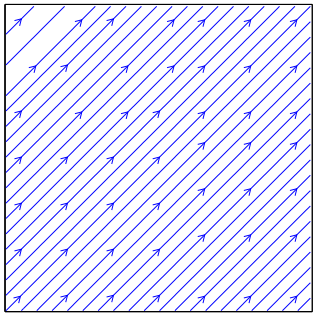
(4) a figure showing the normalized maximal speed increase, where the normalization factor is the same for all the cases. This factor is the maximal value of $a + c$ in the case of the shear with $\lambda = 0.5$, minus the value of $a + c$ when there are no perturbations. This corresponds to the maximal possible increase.

4.5.3. Presentation of the results

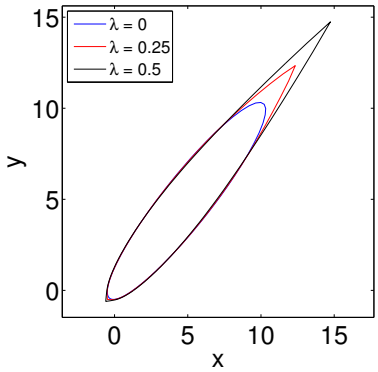
Figure 4.10 shows the results for perturbation amplitudes $\lambda = 0, 0.25, 0.5$ for the case $\delta = 1$, $\bar{v} = 30$ and $\bar{\theta} = \frac{\pi}{4}$ This figure shows the perturbation field over



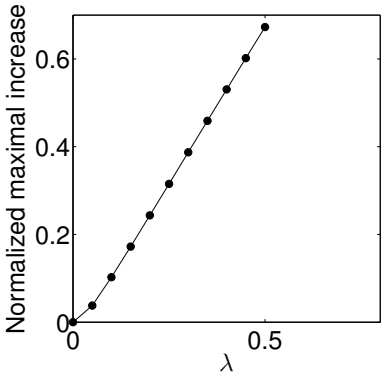
(a) Perturbation field



(b) Resulting wind field



(c) Wulff shapes for different λ's

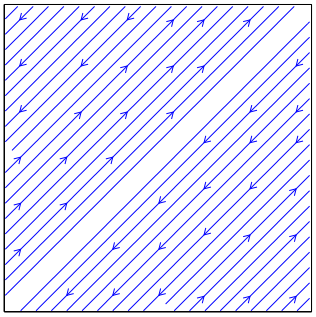


(d) Normalized maximal increase in speed as a function of λ

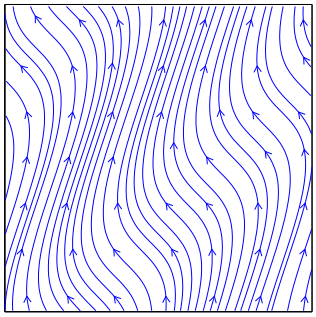
FIGURE 4.10. Perturbations, wind flow, Wulff shapes and normalized maximal speed increase for the case $\delta = 1$, $\bar{v} = 30$ in direction $\theta = \frac{\pi}{4}$

one period is a shear aligned with the average wind. It can be seen that as the perturbation amplitude increases, the Wulff shapes get bigger. This means the fire spreads faster in the presence of perturbations, even though their average is zero. The fourth graph shows this speed increase is a roughly linear function of λ.

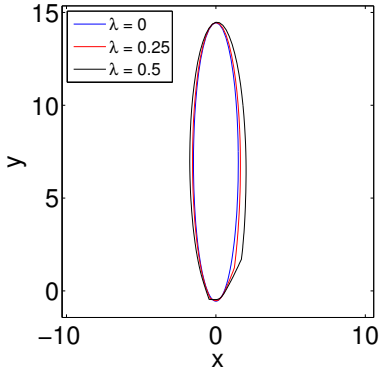
Figure 4.11 shows the results for perturbation amplitudes $\lambda = 0, 0.25, 0.5$ for the case $\delta = 1$, $\bar{v} = 30$ and $\bar{\theta} = 0$.



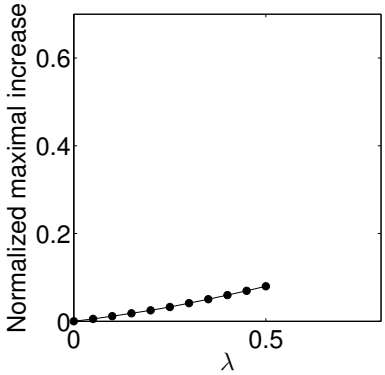
(a) Perturbation field



(b) Resulting wind field



(c) Wulff shapes for different λ's

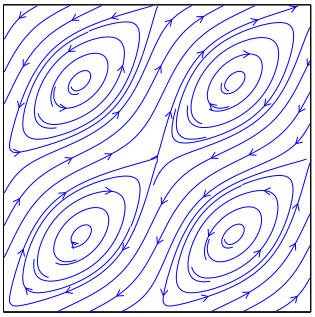


(d) Normalized maximal increase in speed as a function of λ

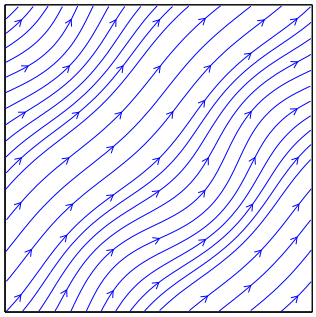
FIGURE 4.11. Perturbations, wind flow, Wulff shapes and normalized maximal speed increase for the case $\delta = 1$, $\bar{v} = 30$ in direction $\theta = 0$

The difference between figure 4.11 and figure 4.10 is the angle of the average wind speed. In figure 4.11, the shear is no longer aligned perfectly with the average wind. The resulting speed increase is much smaller.

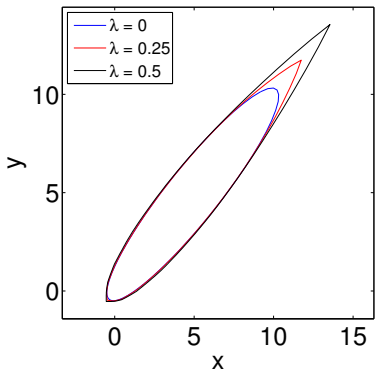
Figure 4.12 shows the results for the case $\delta = 0.5$, $\bar{v} = 30$ and $\bar{\theta} = \frac{\pi}{4}$



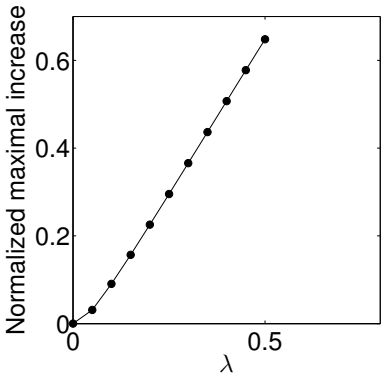
(a) Perturbation field



(b) Resulting wind field



(c) Wulff shapes for different λ's

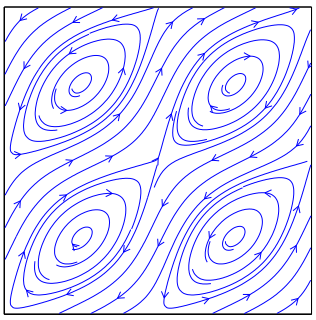


(d) Normalized maximal increase in speed as a function of λ

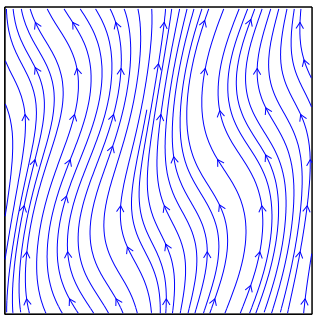
FIGURE 4.12. Perturbations, wind flow, Wulff shapes and normalized maximal speed increase for the case $\delta = 0.5$, $\bar{v} = 30$ in direction $\theta = \frac{\pi}{4}$

This case shows what happens when the perturbations are a mix of vortices with shear flows. The perturbations are aligned with the average wind. The speed increase in this case is not as big as in the case of the shear-only perturbations.

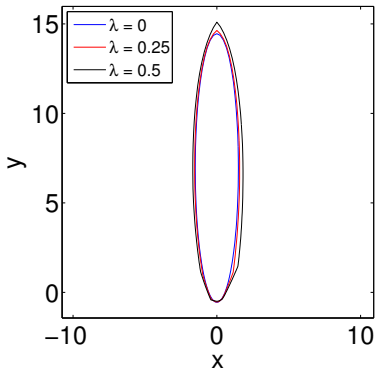
Figure 4.13 shows the results for the case $\delta = 0.5$, $\bar{v} = 30$ and $\bar{\theta} = 0$.



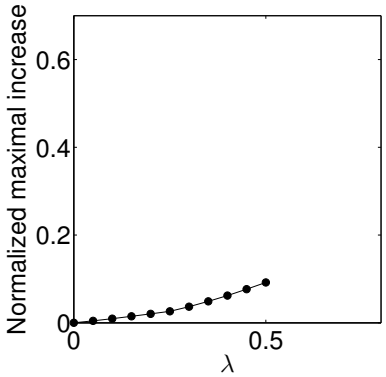
(a) Perturbation field



(b) Resulting wind field



(c) Wulff shapes for different λ's

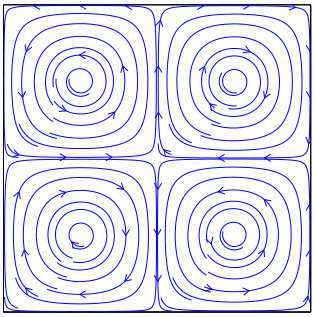


(d) Normalized maximal increase in speed as a function of λ

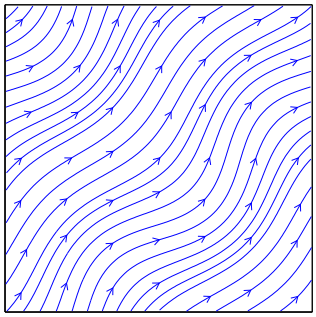
FIGURE 4.13. Perturbations, wind flow, Wulff shapes and normalized maximal speed increase for the case $\delta = 0.5$, $\bar{v} = 30$ in direction $\theta = 0$

This case has perturbations composed of vortices and shears, but now they are no longer aligned with the average wind. This considerably reduces the speed increase.

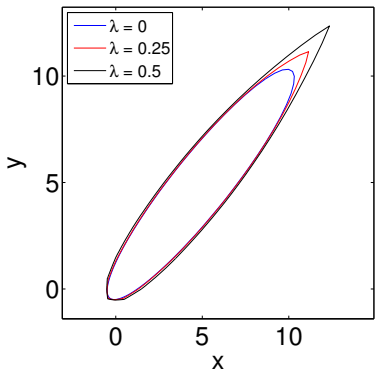
Figure 4.14 shows the results for the case $\delta = 0$, $\bar{v} = 30$ and $\bar{\theta} = \frac{\pi}{4}$.



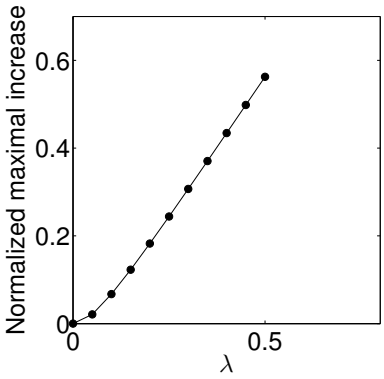
(a) Perturbation field



(b) Resulting wind field



(c) Wulff shapes for different λ's

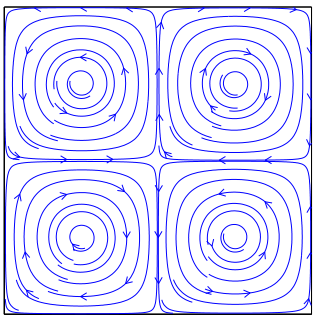


(d) Normalized maximal increase in speed as a function of λ

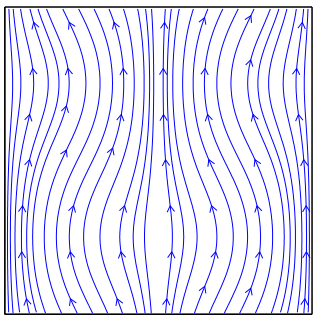
FIGURE 4.14. Perturbations, wind flow, Wulff shapes and normalized maximal speed increase for the case $\delta = 0$, $\bar{v} = 30$ in direction $\theta = \frac{\pi}{4}$

The speed increase is smaller than in the cases $\delta = 0.5$ and $\delta = 1$.

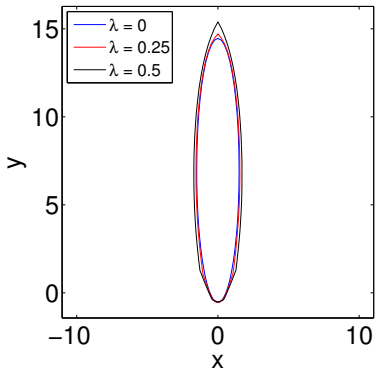
Figure 4.15 shows the results for the case $\delta = 0$, $\bar{v} = 30$ and $\bar{\theta} = 0$.



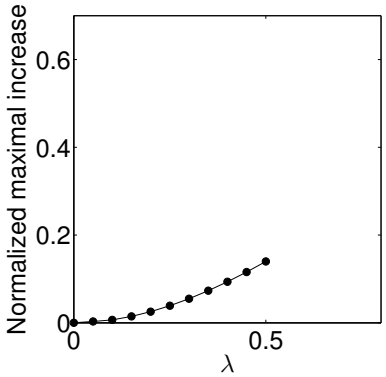
(a) Perturbation field



(b) Resulting wind field



(c) Wulff shapes for different λ's



(d) Normalized maximal increase in speed as a function of λ

FIGURE 4.15. Perturbations, wind flow and Wulff shapes, and normalized maximal increase for the case $\delta = 0$, $\bar{v} = 30$ in direction $\theta = 0$

For these last two tests, the small scale flow is composed of vortices. This case shows the least amount of speed increase.

4.5.4. Summary of the results

A compact way of looking at the effect of the various parameters on the effective speed is to plot the normalized maximal speed increase as a function of λ, for many different parameters. This is shown on Figure 4.16

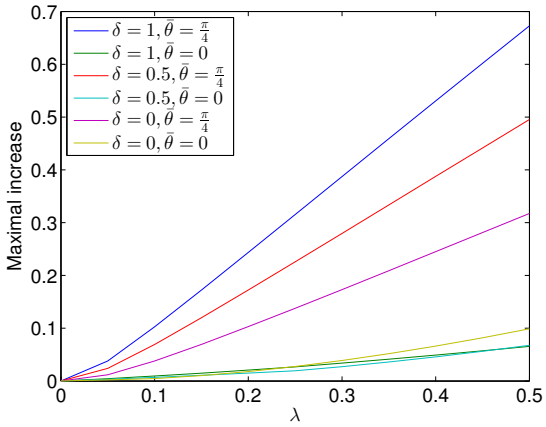


FIGURE 4.16. Normalized speed increase as a function of λ for different values of δ and $\bar{\theta}$

This graph shows the increase is biggest when the perturbations are well-aligned with the average wind. Also, the case with the biggest increase is the case of the shear. The graphs are approximately linear so the effective speed seems to increase linearly with the amplitude of the perturbations.

4.5.5. Effect on the burnt area growth factor

This section considers the case $\delta = 1$. A graph showing the value of the burnt area growth factor (equation (4.3.4)) for different values of λ is presented in Figure 4.17.

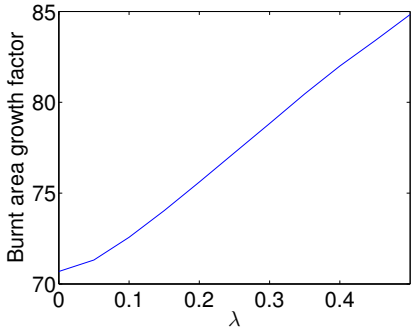


FIGURE 4.17. Burnt area growth factor as a function of λ

The coefficient varies approximately linearly with the amplitude of the small-scale perturbations. This is expected since Wulff shapes corresponding to larger perturbation amplitudes are stretched along one dimension.

4.5.6. Comparing different types of solutions

This sections shows a comparison of solutions computed with well-resolved small-scale perturbations, solutions computed by homogenizing the perturbations and solutions computed when there is no small-scale variation. Figure 4.18 shows a high resolution solution with the wind field for $\delta = 1$. The propagation of a circle of radius 2 centered at the origin is computed until time 0.7.

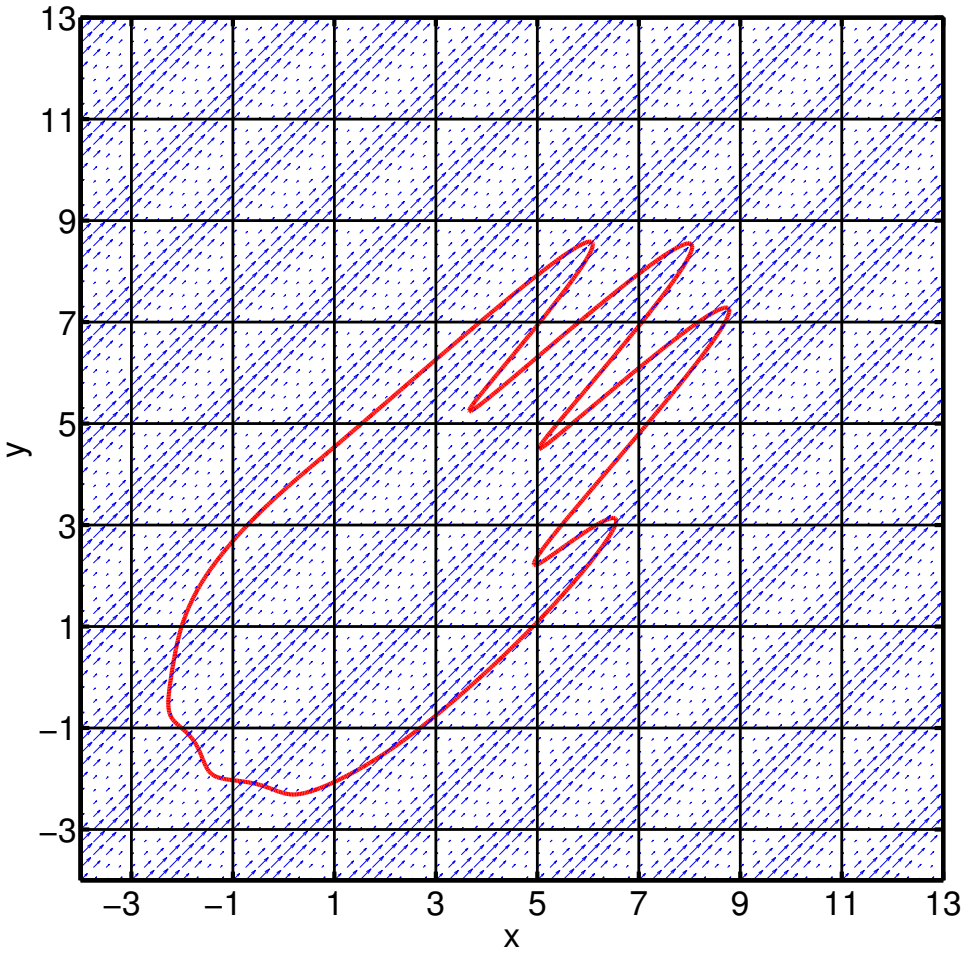


FIGURE 4.18. High resolution solution with the wind field for $\delta = 1$

Figure 4.19 shows the 3 types of solutions.

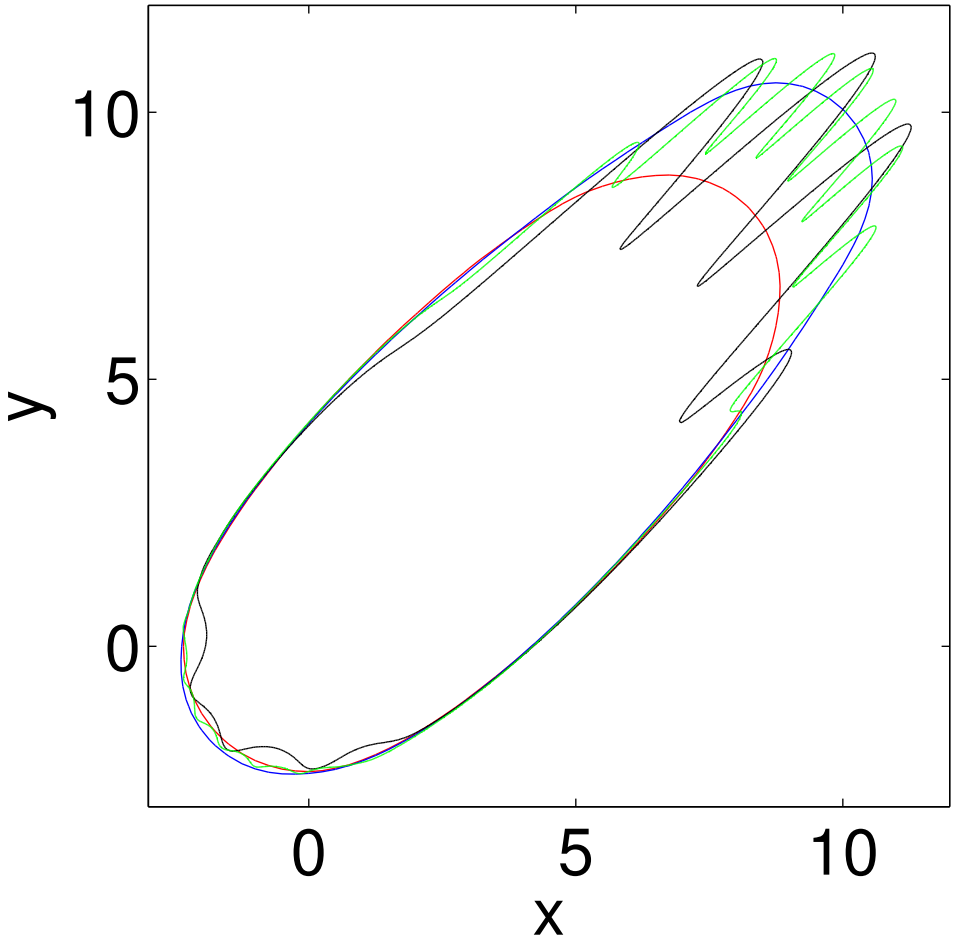


FIGURE 4.19. Comparison of different solutions for the case $\delta = 1$. Solution computed without perturbations (red), homogenized solution (blue), solution with $\epsilon = 2$(black) and solution with $\epsilon = 1$ (green).

The homogenized solution fits well with the high resolution solution. By taking ϵ to be smaller, the high resolution solution is expected to converge to the homogenized solution. Another important point is the solution computed with no small-scale perturbations has a smaller burnt area.

Consider now the case $\delta = 0.5$, which consists in a mix of shears and vortices. Figure 4.20 shows a high resolution solutions with the wind field for $\delta = 0.5$.

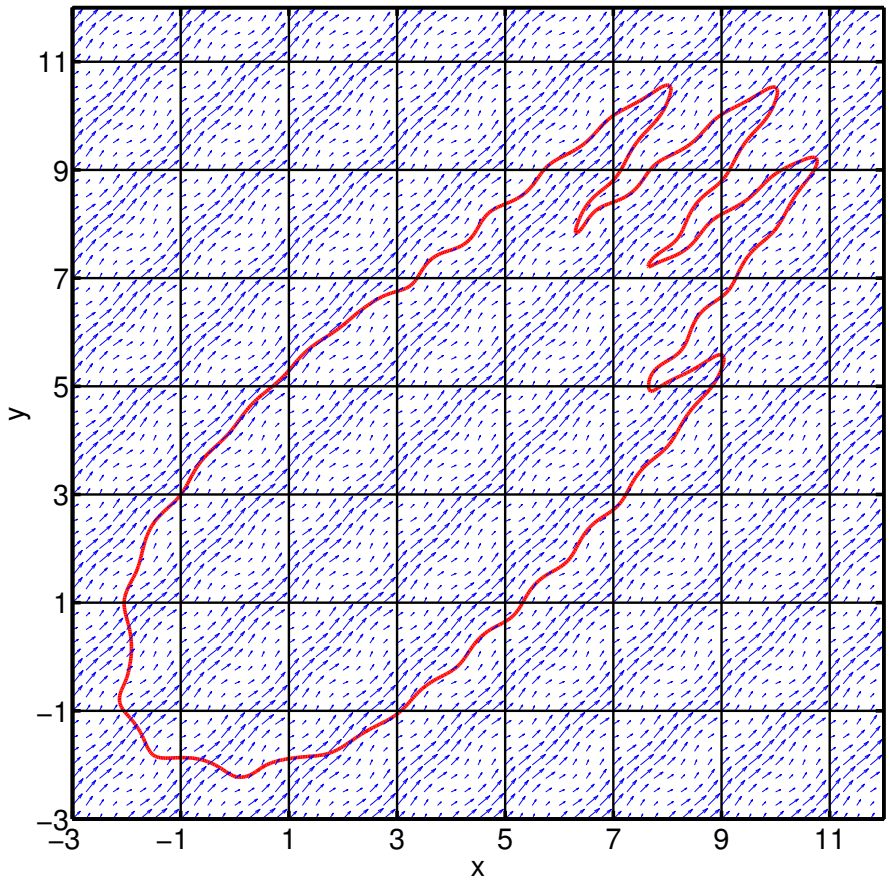


FIGURE 4.20. High resolution solution with the wind field for $\delta = 0.5$

Figure 4.21 shows the 3 types of solutions.

The fire spreads faster in the presence of perturbations. But, the fire spreads slower than in the case of the shear. These last examples show how a homogenized solution approximates a high resolution solution when the perturbations are periodic. The next section shows how to apply the conclusions drawn from periodic test cases to more realistic test cases.

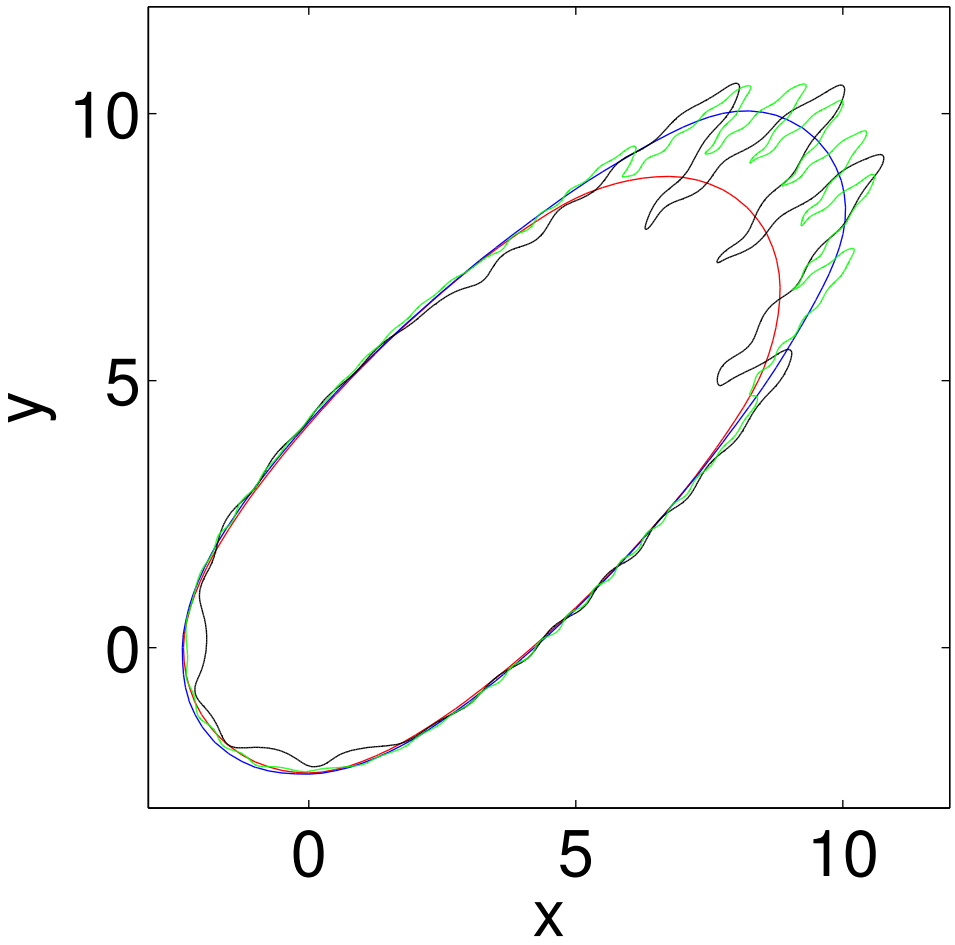


FIGURE 4.21. Comparison of different solutions for the case $\delta = 0.5$. Solution computed without perturbations (red), homogenized solution (blue), solution with $\epsilon = 2$ (black) and solution with $\epsilon = 1$ (green).

4.6. Approach for realistic test cases

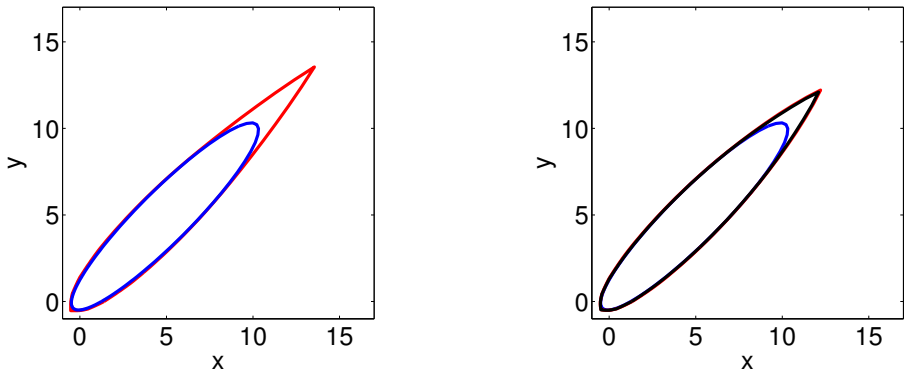
Directly applying the homogenization results of the previous sections would be difficult. In practice, the detail of the windspeed at small-scales is unknown. Periodic perturbations are an idealized form of the small-scale variation of the wind. Even if we wanted to use periodic perturbations, effective speed profiles would have to be computed for different values of all the important parameters (vegetation, wind speed, topography, etc...) and for different small-scale perturbation types. The amount of computations required would be too large.

The lessons learned from periodic homogenization can help us take into account the effect of small-scale perturbations without computing homogenized

speed profiles. It was shown in Section 4.5 that the speed increase resulting from the perturbations is a roughly linear function of the amplitude of the perturbations λ when the perturbations are aligned with the average wind. Another observation made in the previous sections is the fact that small-scale perturbations change the fire shape. But, this shape modification is slight. Using an ellipse which fits the effective Wulff shape is a good approximation. This means in practice, taking into account the effect of small-scale perturbations can be accomplished by modifying the ellipses governing the firespread. The effective rate of spread ($\overline{ROS}$) can be given by an equation of type $\overline{ROS} = ROS + K\lambda$ with K determined experimentally and depending on different parameters and ROS being the rate of spread in the case without perturbations. The effective $FROS$ and $BROS$ are the same as the original $FROS$ and $BROS$ in the cases considered in the previous sections. From these, the effective ellipse coefficients are obtained.

This section will begin by showing how small-scale perturbations add a correction to the original Wulff shape which scales as a function of λ. Then, the fact that effective Wulff shapes can be approximated by ellipses will be shown. These results will be applied to a test case generated with data from Prometheus. Solutions to propagation problems will be computed in three different ways : without small-scale perturbations, with small-scale perturbations and with effective propagation speeds determined experimentally. This will show how modifying the elliptical parameters to take into account small-scale perturbations can be an efficient way to take into account these perturbations.

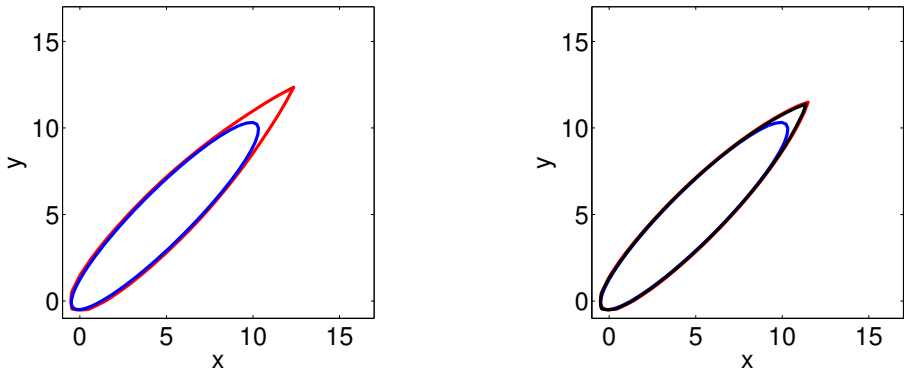
Consider again the case $\delta = 0.5$ and $\theta = \frac{\pi}{4}$. The homogenized speed profiles are ellipses, plus a correction aligned in direction $\frac{\pi}{4}$, as shown on Figure 4.22. This correction scales as a function of λ. Figure 4.22 shows how the correction computed for the case $\lambda = 0.5$ can be rescaled to obtain the correction for the case $\lambda = 0.3$.



(a) (b)

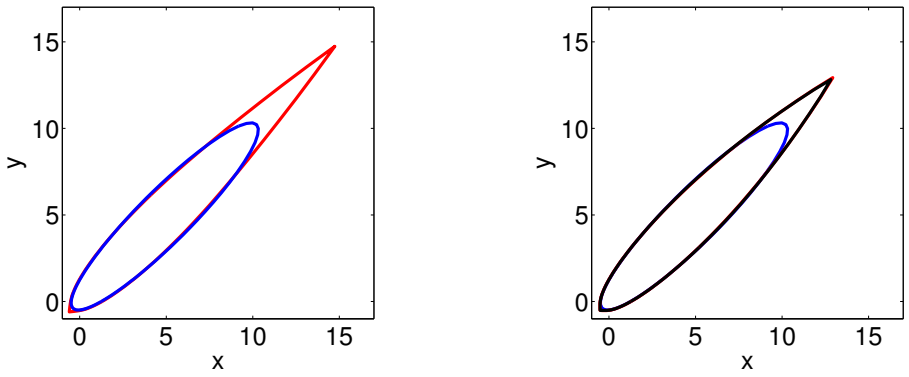
FIGURE 4.22. Correction to the elliptical shape when $\delta = 0.5$ and the average wind is in direction $\frac{\pi}{4}$. (a) shows the correction to the ellipse shape when $\lambda = 0.5$ and (b) shows the original ellipse (in blue), the Wulff shape for $\lambda = 0.3$ and the Wulff shape for $\lambda = 0.5$ rescaled by a factor of 0.6

This observation is also valid for other values of δ. Figure 4.23 shows what happens for the case $\delta = 0$ with the average wind toward $\frac{\pi}{4}$ and Figure 4.24 shows what happens for the case $\delta = 1$.



(a) (b)

FIGURE 4.23. Correction to the elliptical shape when $\delta = 0$ and the average wind is in direction $\frac{\pi}{4}$. (a) shows the correction to the ellipse shape when $\lambda = 0.5$ and (b) shows the original ellipse (in blue), the Wulff shape for $\lambda = 0.3$ and the Wulff shape for $\lambda = 0.5$ rescaled by a factor of 0.6



(a) (b)

FIGURE 4.24. Correction to the elliptical shape when $\delta = 1$ and the average wind is in direction $\frac{\pi}{4}$. (a) shows the correction to the ellipse shape when $\lambda = 0.5$ and (b) shows the original ellipse(in blue), the Wulff shape for $\lambda = 0.3$ and the Wulff shape for $\lambda = 0.5$ rescaled by a factor of 0.6

The next point to consider is how to replace the effective fire shape by the best fitting ellipse. The effective ROS is given by the maximal value of the speed profile. The effective $BROS$ is the minimal value of the speed profile and it is barely affected by the small-scale perturbations. The alignment of the corresponding ellipse is given by the direction corresponding to the effective ROS.

For example, for the case $\delta = 0.5$ with a main wind in direction $\frac{\pi}{4}$, the effective ROS is given by :

$$\overline{ROS} = ROS + 10.1\lambda$$

and this maximal propagagation speed is towards $\frac{\pi}{4}$. For the case $\delta = 0$, the effective ROS is given by equation

$$\overline{ROS} = ROS + 6.6\lambda.$$

For the case $\delta = 1$, the effective ROS is given by equation :

$$\overline{ROS} = ROS + 13.5\lambda.$$

Then, an ellipse having parameters a, b, c that closely fit the ROS, $BROS$ and $FROS$ of the homogenized speed profile can be chosen. Suppose then one wants to compute the propagation of a front in the presence of small-scale perturbations. The following procedure can be applied.

(1) Choose the perturbation amplitude and other propagation parameters such as the vegetation at every point of the domain.

(2) Compute the effective ROS,BROS and FROS by using experimental values that depend on the perturbation amplitude and the other propagation parameters.

(3) Choose the corresponding ellipse parameters at every point of the domain :

   (a) Use the ROS and $BROS$ to define $a = \frac{ROS+BROS}{2}$, $c = \frac{ROS-BROS}{2}$ and $b = FROS$.

Using this procedure for the study of the propagation of a front with average wind speed equal to 30, a wind alignment of $\frac{\pi}{4}$, $\delta = 0.5$ and perturbation amplitude $\lambda = 0.5$ gives the ellipse shown in Figure 4.25.

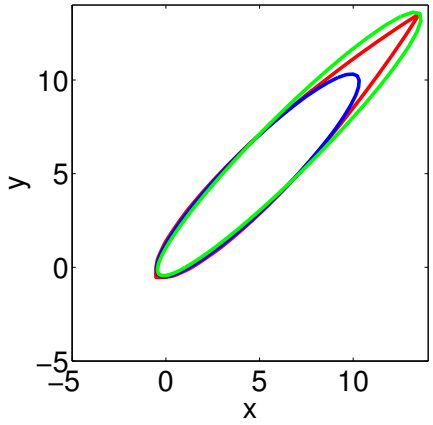


FIGURE 4.25. Original speed profile in blue, homogenized speed profile in red, and elliptical speed profile chosen to approximate the homogenized speed profile in green

4.7. Test cases with realistic data

The procedure described previously is now applied to a test case constructed with data from Prometheus (Tymstra et al., 2010). The vegetation is now heterogeneous and homogenized speed profiles for the different vegetation types are not available. Figure 4.26 shows the initial propagation domain of size 1250 m × 1250 m. The colors correspond to values of parameter a.

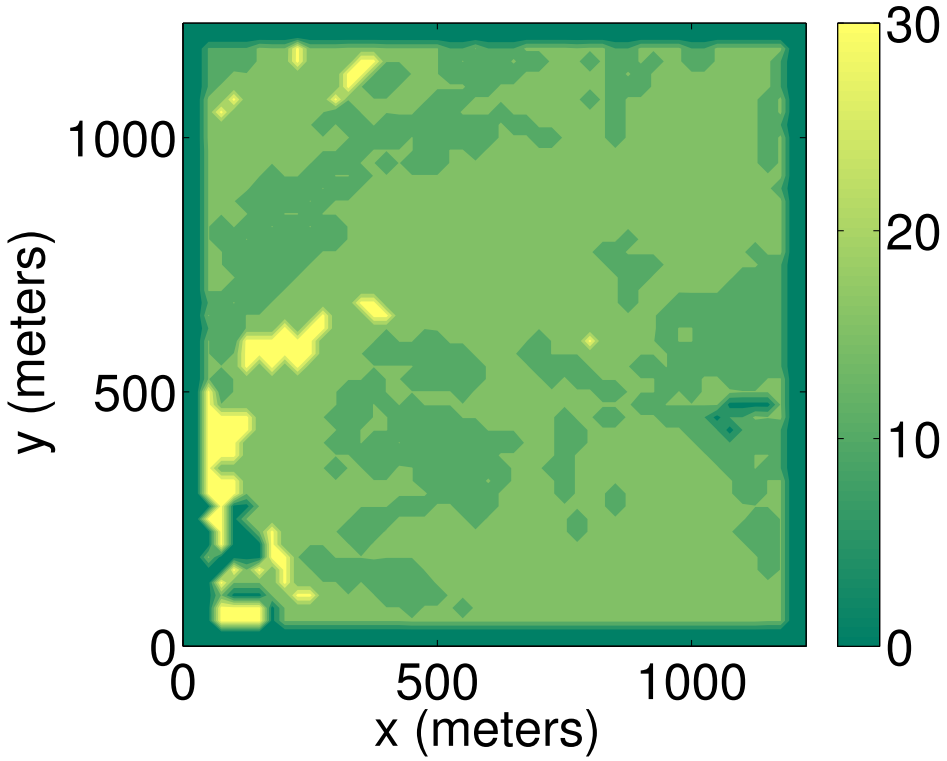


Figure 4.26. Fuel map used based on data from Prometheus

The vegetation varies every 25 m which gives a grid size of 50 points by 50 points. But, the goal is to consider smaller-scale heterogeneities in the windspeed. An average wind of speed 30 km/h aligned with $\theta = \frac{\pi}{4}$ is present. Small-scale perturbations on this average wind are represented by a periodic perturbation on the flow, with $\delta = 0.5$ and $\lambda = 0.5$. The period is 25 m and the wind varies at the scale of 1 m. This is a mix of a shear with vortices where the diameter of the vortices introduced is around 12 m. This gives a grid of size 1250 points by 1250 points.

Figure 4.27 shows the values of parameter a in the presence of small-scale perturbations.

The following three solutions are now compared :

(1) A solution to the propagation problem with no perturbations.

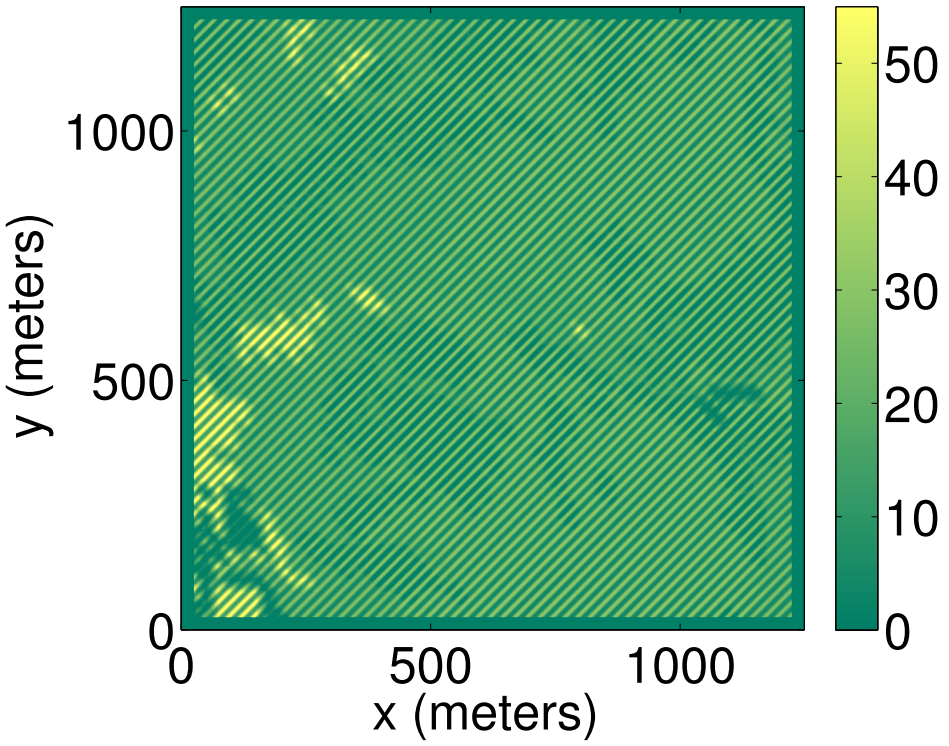


FIGURE 4.27. Values of coefficient a in the presence of perturbations

(2) A solution obtained with perturbations.

(3) A solution computed by using modified ellipse coefficients to take into account the effect of the perturbations. The new ellipse coefficients vary on the same scale as the vegetation.

For every type of vegetation, the modified ellipse coefficients are different. For $\lambda = 0.5$, the maximal wind speed is 45 km/h. From this maximal wind speed, maximal values over a periodic box for a and c can be computed (a_{max}, c_{max}) for every vegetation. Then, the effective ROS and $BROS$ can be computed with :

$$\overline{ROS} = K\lambda(a_{max} - a_0 + c_{max} - c_0) + a_0 + c_0$$
$$\overline{BROS} = a_0 - c_0$$

where $a_0 + c_0$ is the ROS when the wind is 30 km/h (to every vegetation type correspond different values of a_0 and c_0). The constant K was experimentally determined to be 0.2. Then, the effective a and c can be computed.

Figure 4.28 shows the difference between the increased value of parameter a and its value in the absence of perturbations. This increase corresponds approximately to a 10% increase to the original value.

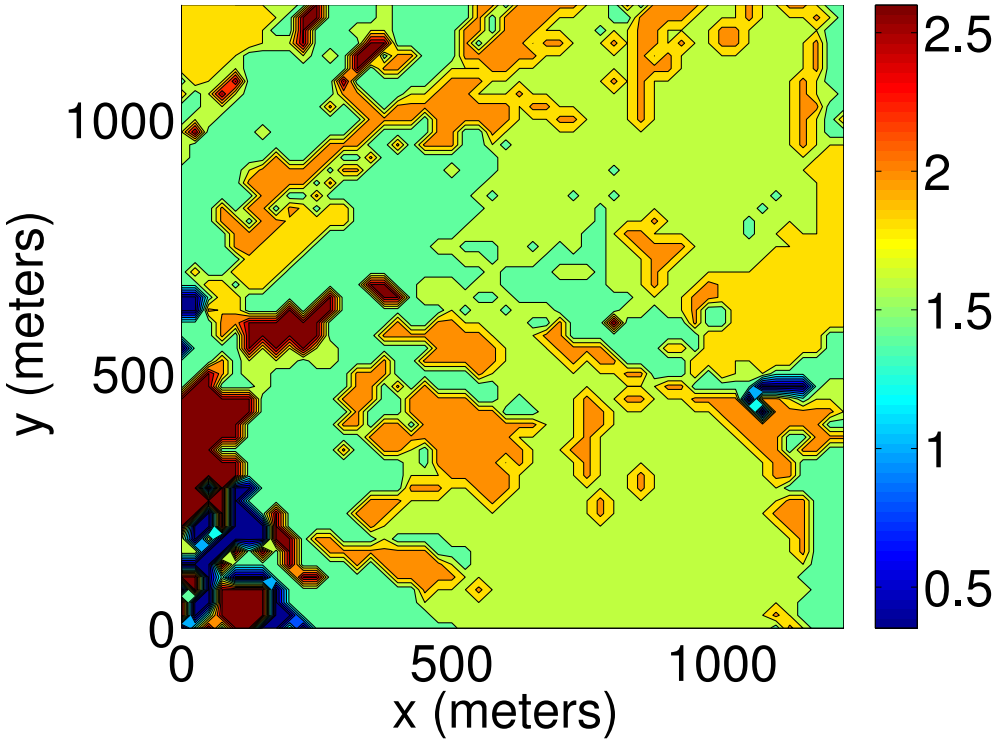


FIGURE 4.28. Increase in coefficient a in the presence of perturbations

Figure 4.29 shows the solutions obtained in the 3 different ways. The ignition point is $(x, y) = (50m, 50m)$. The flame front position after 50 minutes is shown.

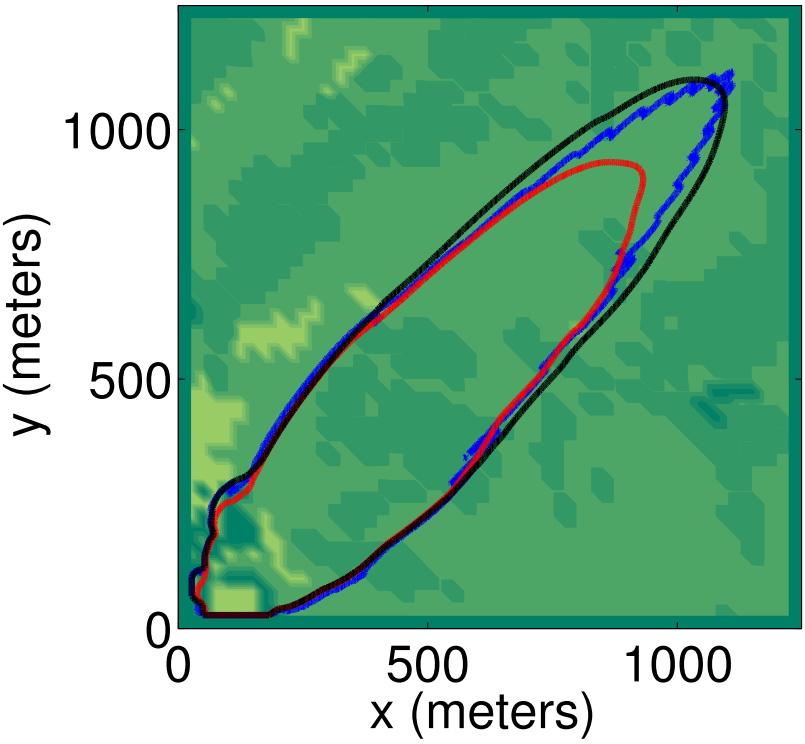


FIGURE 4.29. Solution computed with small-scale variations (in blue), with modified ellipse coefficients (in black) and without small-scale variations (in red)

The solution computed with modified ellipse coefficients is a better approximation of the solution computed with small-scale variations. It better captures the behavior of the fire at the head of the fire than the solution computed with no modifications to the ellipse coefficients.

Figure 4.30 shows another example on a domain of size 1750 m $\times$ 1750 m. The vegetation is defined with the same resolution as before, but now the vegetation grid is of size 70 points by 70 points. The perturbation on the wind is the same as before, so the fine grid with resolution 1m is of size 1750 points by 1750 points.

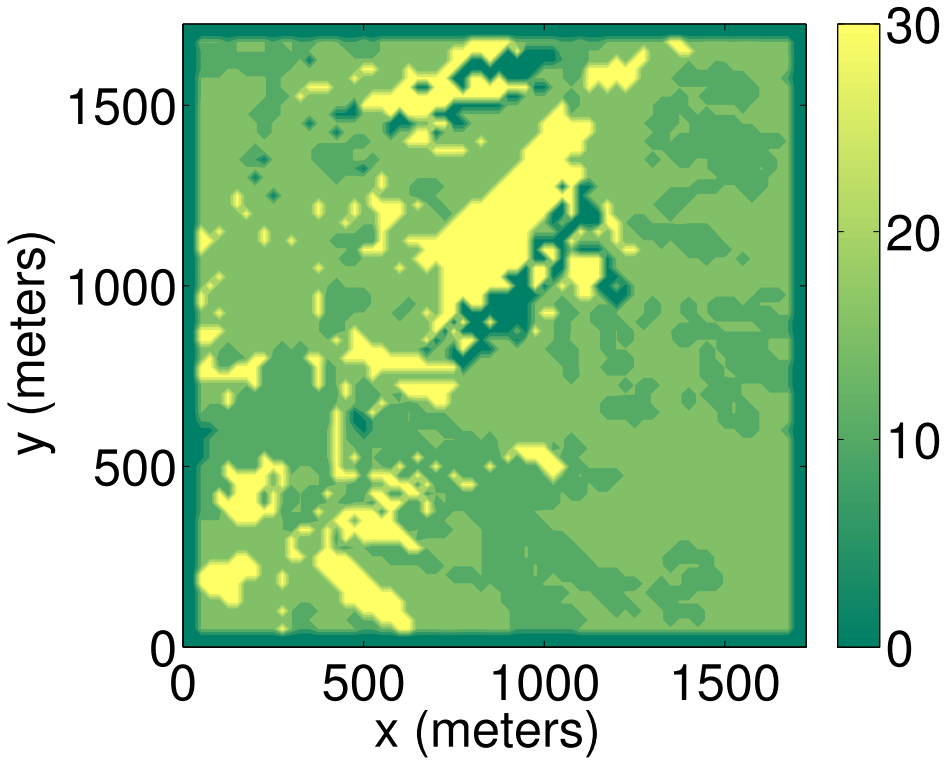


FIGURE 4.30. Fuel map used based on data from Prometheus for the second example

Figure 4.31 shows the small-scale variations.

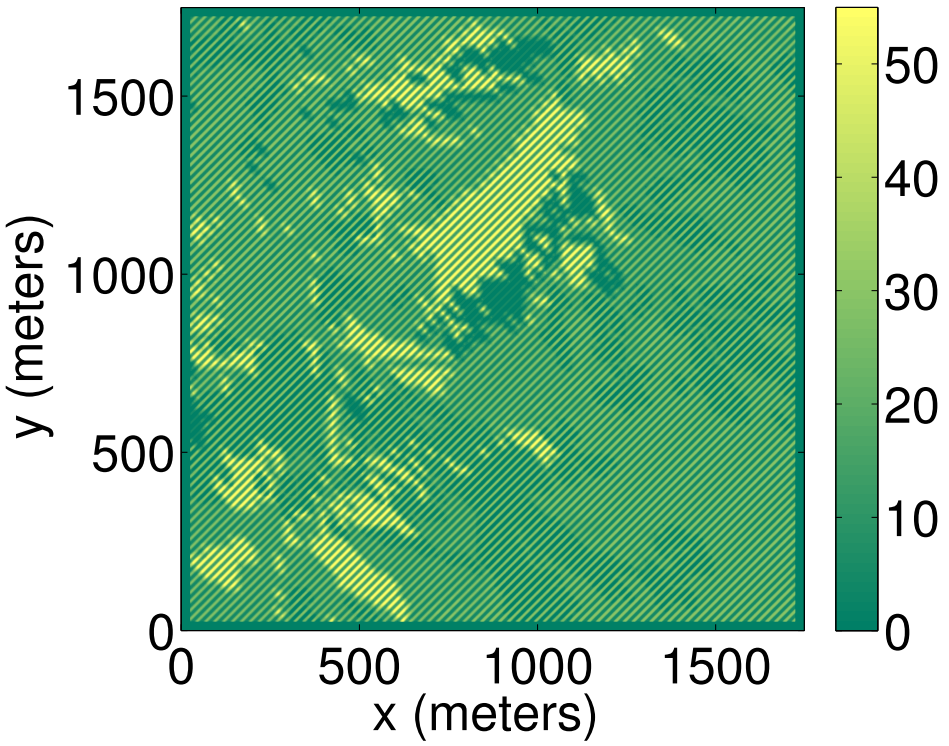


FIGURE 4.31. Values of coefficient a in the presence of perturbations

Figure 4.32 shows the speed increase. Constant K was chosen to be 1.2 in this case. The difference in the value of K comes from the fact the dominant vegetation type is not the same in the two cases. Ideally, this constant should vary with the vegetation since the equations giving the speed vary from one vegetation type to another.

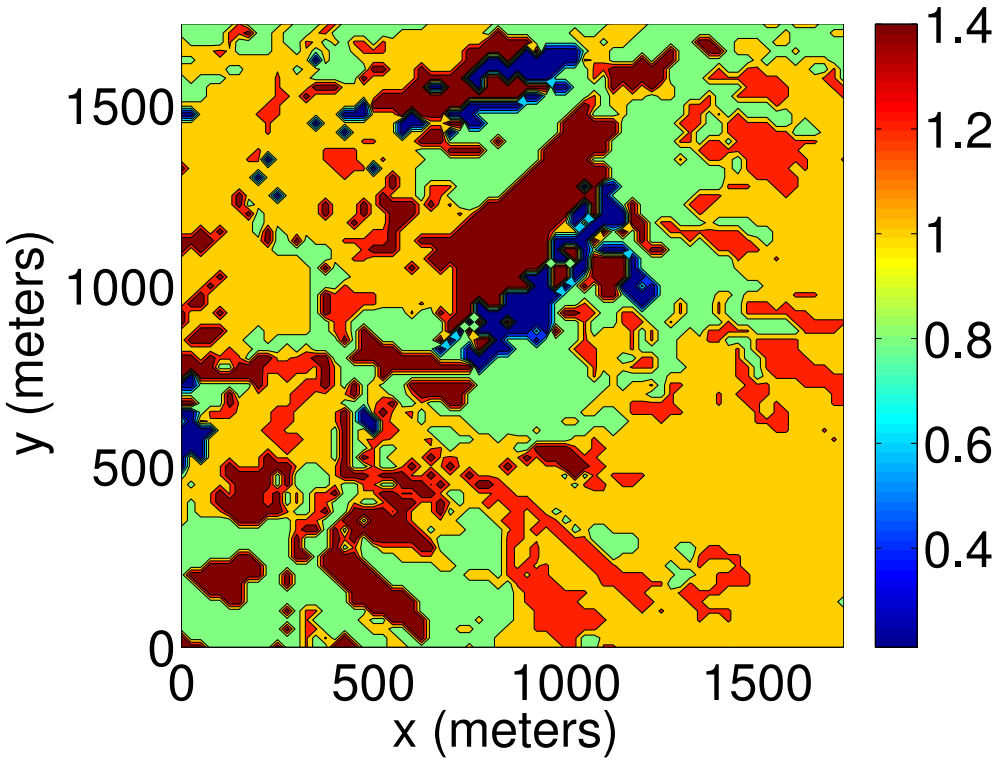


FIGURE 4.32. Increase in coefficient a in the presence of perturbations for the second domain

Figure 4.33 shows the different solutions for a final time of 55 minutes.

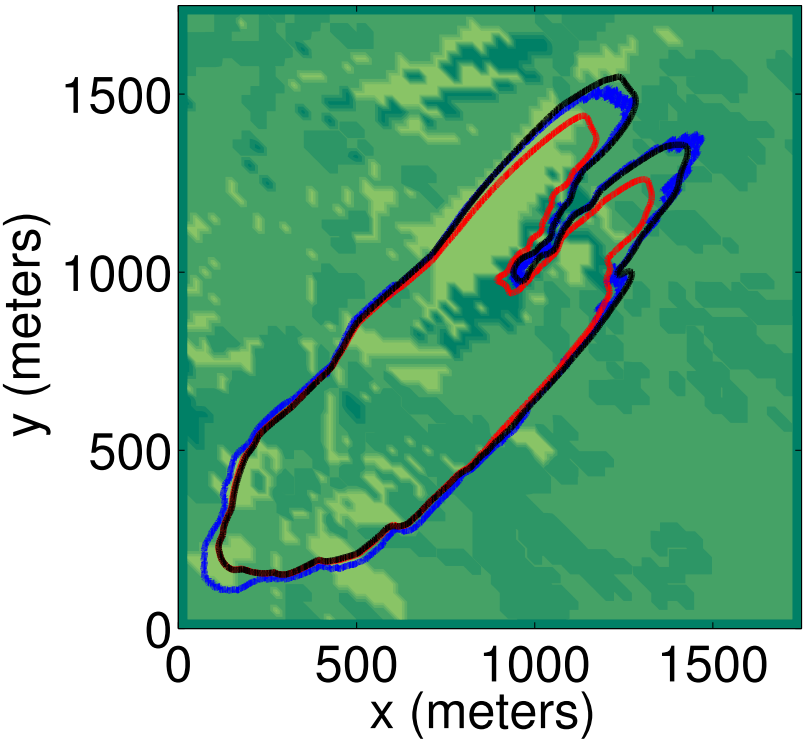


FIGURE 4.33. Solution computed with small-scale variations (in blue), with modified ellipse coefficients (in black) and without small-scale variations (in red)

These two examples show that in practice, the effect of small-scale variations can be captured by using modified ellipse coefficients that only vary on larger scales. In more realistic examples, the constant K would have to vary as a function of the vegetation.

4.8. Conclusion

The goal of this paper was to show how the theory of homogenization can be used in the context of forest fire simulation. Forest fires are an inherently multi-scale phenomenon. In particular, they generate variations of the wind speed on small-scales. These small-scale variations have a global effect on the fire propagation, but computing them in detail is too costly. Homogenization is a tool which allows the computation of the large-scale effect of these perturbations, without using a high resolution.

The small-scale variations on the wind were modeled as periodic perturbations given by either shears, vortices or a mix of both of these cases. These choices were made because it has been observed that forest fires can create small-scale vortices in the wind flow. Also, homogenization results are available for the periodic case.

The firespread was modeled using Richards' ellipse model and it was computed by solving a corresponding Hamilton-Jacobi partial differential equation. This choice of mathematical tool for the computation of the firespread was motivated by the existence of homogenization tools for this particular equation.

Homogenization allowed for the efficient computation of effective propagation speeds. The influence of the perturbation amplitude on the resulting effective speed was studied. In the cases studied, the effect of zero-mean perturbations was to increase the propagation speed. This increase was an approximately linear function of the amplitude of the perturbation. This increase is significant in the most realistic case studied (a perturbation consisting of shears and vortices aligned with the average wind).

Another conclusion is that perturbations can alter the shape of the fire from an ellipse, in some cases resulting in a teardrop shape that has been observed in practice. But, computing firespread using modified speed profiles is not necessary to adequately simulate the propagation. The numerical experiments conducted in this paper suggest modifying the ellipse coefficients instead of using modified shapes could be a good way to include the effect of small-scale variations in the wind speed on the firespread.

In order to apply the results of this paper to live situations, data on real fires would have to be collected in order to calibrate this modification to the ellipse parameters as a function of the various pertinent parameters.

4.9. Acknowledgments

I would like to thank professor Anne Bourlioux, my PHD advisor, for much-valued input about the contents of this paper and for financing this project.

Bibliographie

Y. Achdou, F. Camilli, and I. C. Dolcetta. Homogenization of Hamilton-Jacobi equations : Numerical methods. *Math. Models Methods Appl. Sci.*, 18, 2008.

J. Barber, C. Bose, A. Bourlioux, J. Braun, E. Brunelle, T. Garcia, T. Hillen, C. Lam, B. Ong, C. Poschl, R. Bryce, and C. Tymstra. *Prometheus - Canada's Wildfire Growth Simulator, Chapitre 1*, 2007.

G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic analysis*, 4 :271–283, 1991.

A. Bourlioux. Numerical turbulent combustion : an asymptotic view via an idealized test-case. *Modern Methods in Scientific Computing and Applications*, Kluwer Academic publishers :81–102, 2002.

D.L. Chopp. Computing minimal surfaces via level set curvature flow. *J. Comput. Phys.*, 106 :77–91, 1993.

T. L. Clark, M. A. Jenkins, J. Coen, and D. Packham. A coupled atmosphere-fire model : convective feedback on fireline dynamics. *Journal of Applied Meteorology*, 35 :875–901, 1996a.

T. L. Clark, M. A. Jenkins, J. L. Coen, and D. R. Packham. A coupled atmosphere-fire model : Role of the convective Froude number and dynamic fingering at the fireline. *Int j. Wildland Fire*, 6(4) :177–190, 1996b.

M. G. Crandall and P. L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43 :1–19, 1984.

M. G. Crandall and L. Tartar. Some relations between nonexpansive and order preserving mappings. *Proc. Amer. Math. Soc.*, 79 :74–80, 1979.

M.G. Crandall and P-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. AMS*, 277 :1–43, 1983.

M. Delfour and J. P. Zolesio. *Shapes and Geometries : Metrics, Analysis, Differential Calculus, and Optimization, Second Edition*. SIAM, Philadelphia, USA, 2011.

M. Fiedler and V. Ptak. On matrices with non-positive off-diagonal elements and positive principal minors. *Czechoslovak Mathematical Journal*, 12(3) :382–400,

1962.

A. Desfossés Foucault. Simulation numérique de feux de forêt avec réinitialisation et contournement d'obstacles. *Mémoire de maîtrise*, Université de Montréal, Montréal,Québec, 2008.

A. Desfossés Foucault. A level-set algorithm for the simulation of anisotropic firespread. *En préparation*, 2015a.

A. Desfossés Foucault. A semi-implicit scheme for steady anisotropic Hamilton-Jacobi pdes with applications to the simulation of forest fires. *En préparation*, 2015b.

J. Glasa and L. Halada. A note on mathematical modeling of elliptical fire propagation. *Computing and Informatics*, 30 :1303–1319, 2011.

D.G. Green. Shapes of simulated fires in heterogeneous fuels. *Ecol. Model.*, 20 : 21–32, 1983.

Forestry Canada Fire Danger Group. Development and structure of the Canadian forest fire behavior prediction system. *Information report ST-X-3*, Forestry Canada, 1992.

C. Kao, S. Osher, and Y. Tsai. Fast sweeping methods for static Hamilton-Jacobi equations. *SIAM J. Num. Anal.*, 42 :2612–2632, 2003.

B. Khouider. Modélisation asymptotique pour la simulation aux grandes échelles de la combustion turbulente prémélangée. *Thèse de doctorat*, Université de Montréal, Montréal, Québec, 2002.

P. L. Lions, G. Papanicolaou, and S.R.S. Varadhan. Homogenization of Hamilton-Jacobi equations. *Unpublished*, 1988.

S. Luo, Y. Yu, and H. Zhao. A new approximation for effective hamiltonians for homogenization of a class of Hamilton-Jacobi equations. *Multiscale model. simul.*, 9(2) :711–734, 2011.

V. Mallet, D.E. Keyes, and F.E. Fendell. Modeling wildland fire propagation with level set methods. *Computers and Mathematics with Applications*, 57(7) :1089 – 1101, 2009.

Mathworks. Documentation on the profile function, http ://www.mathworks.com/help/matlab/ref/profile.html, 2015-03-14. URL http://www.mathworks.com/help/matlab/ref/profile.html.

N. Z. Neyney. A numerical approach for modeling forest fire spread. *Rapport de stage de maîtrise*, Université de Montréal, 2008.

A. M. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations : Hamilton-Jacobi equations and free boundary problems. *SIAM J. Numer. Anal.*, 44 :879–895, 2006.

A. M. Oberman, R. Takei, and A. Vladimirsky. Homogenization of metric Hamilton-Jacobi equations. *Multiscale Model. Simul.*, 8(1) :269–295, 2009.

S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. New York, 2002.

S. Osher and B. Merriman. The wulff shape as the asymptotic limit of a growing crystalline interface. *Asian J. Math.*, 1(3) :560–571, 1997.

S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79 : 12–49, 1988.

J. Qian. Two approximations for effective hamiltonians arising from homogenization of Hamilton-Jacobi equations. *Preprint*, 2003.

G. Richards. A general mathematical framework for modelling 2 dimensional wildland fire spread. *Int. J. Wildland Fire*, 5(2), 1994.

E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal.*, 29 :867–884, 1992.

J. A. Sethian. *Level Set Methods and Fast Marching Methods :Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 1999.

J.A Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.*, 93(4) :1591–1595, 1996.

J.A Sethian and A.B. Vladimirsky. Ordered upwind methods for static Hamilton-Jacobi equations : Theory and algorithms. *SIAM J. Numer. Anal.*, 41(1) : 325–363, 2003.

R. Sun, S.K. Krueger, M. A. Jenkins, M. A. Zulauf, and J. J. Charney. The importance of fire-atmosphere coupling and boundary-layer turbulence to wildfire spread. *International Journal of Wildland Fire*, 18 :50–60, 2009.

M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114 :145–159, 1994.

J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40 :1528–1538, 1995.

C. Tymstra, R. Bryce, B.M. Wotton, S.W. Taylor, and O.B. Armitage. Development and structure of Prometheus : the Canadian wildland fire growth simulator. *Canadian forest service*, Ressources Naturelles Canada, 2010.

A. Vladimirsky and C. Zheng. A fast implicit method for time-dependent Hamilton-Jacobi PDEs. *arXiv :1306.3506[math.NA]*, 2014.

A.B. Vladimirsky. Fast methods for static Hamilton-Jacobi partial differential equations. *Lawrence Berkeley National Laboratory : Lawrence Berkeley National Laboratory*, Retrieved from : http ://escholarship.org/uc/item/8k28k9t9, 2001.

J. Xin. Front propagation in heterogeneous media. *Amer. Math. Soc.*, Providence, R. I, 1991.

H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250) :603–628, 2004.

Annexe A

FIRESPREAD EQUATION ON A TOPOGRAPHY

To obtain an equation describing the firespread on a topography, we follow the approach of (Neyney, 2008). Define the topography $z = g(x, y)$ and the level-set function $\psi(x, y, z)$. Our goal is to obtain an equation of type $\psi_t + \vec{V} \cdot \nabla\psi = 0$, where $\vec{V}$ is the propagation speed at a given point. Even though the curve representing the flame front is a 1D object moving in 3D, it must remain on the topography so it is effectively moving in 2D. The level-set function $\psi(x, y, z = g(x, y))$ can be explicitly written as a function of only 2 variables $\phi(x, y)$. We impose $\psi(x, y, z = g(x, y)) = \phi(x, y)$. Let us define $p = z_x$ and $q = z_y$. Using the chain rule :

$$\phi_x = \psi_x + \psi_z p$$
$$\phi_y = \psi_y + \psi_z q$$

Also force the gradient of ψ to be tangent to the topography :

$$-p\psi_x - q\psi_y + \psi_z = 0$$

This gives 3 linear equations in ψ_x,ψ_y and ψ_z to solve. The solution is

$$\psi_x = \frac{(1 + q^2)\phi_x - pq\phi_y}{n_1^2}$$
$$\psi_y = \frac{(1 + p^2)\phi_y - pq\phi_x}{n_1^2}$$
$$\psi_z = \frac{p\phi_x + q\phi_y}{n_1^2}$$

where

$$n_1 = \sqrt{1 + p^2 + q^2}$$

A-ii

To facilitate what is to come, define a new base for $\mathbb{R}^3$ relevant to the topography : $e_{\hat{x}}, e_{\hat{y}}, e_{\hat{z}}$. Choose $e_{\hat{z}}$ perpendicular to the topography and with norm 1 :

$$e_{\hat{z}} = \frac{1}{n_1}(-p, -q, 1)$$

Choose $e_{\hat{y}}$ tangent to the topography and in the direction of the wind :

$$e_{\hat{y}} = \frac{1}{n_2}(\sin\theta, \cos\theta, p\sin\theta + q\cos\theta)$$

where n_2 is given by

$$n_2 = \sqrt{1 + (p\sin\theta + q\cos\theta)^2}$$

Finally, take $e_{\hat{x}}$ perpendicular to the first 2 vectors :

$$e_{\hat{x}} = \frac{1}{n_1 n_2}\Big((1+q^2)\cos\theta + pq\sin\theta, -(1+p^2)\sin\theta - pq\cos\theta, -q\sin\theta + p\cos\theta\Big)$$

Now, define $\nabla\psi$ in this base :

$$\hat{\nabla}\psi = (\nabla\psi \cdot e_{\hat{x}}, \nabla\psi \cdot e_{\hat{y}}, \nabla\psi \cdot e_{\hat{z}})$$

Note that now, the third component of $\hat{\nabla}\psi$ is zero. Redefine $\hat{\nabla}\psi$ as this vector, but with no third component. The firespread equation can now be written :

$$\phi_t + \|A\hat{\nabla}\psi\| + \begin{pmatrix} 0 & c & 0 \end{pmatrix} \cdot \hat{\nabla}\psi = 0$$

for

$$A = \begin{pmatrix} b & 0 \\ 0 & a \end{pmatrix}$$

The b parameter represents the rate of spread in direction $e_{\hat{x}}$ and $a + c$ stands for the rate of spread in direction of $e_{\hat{y}}$. We can develop the term $\|A\hat{\nabla}\psi\|$ to get something of shape :

$$\|A\hat{\nabla}\psi\| = \sqrt{k_1\phi_x^2 + k_2\phi_y^2 - 2k_3\phi_x\phi_y}$$

The formulas for k_1,k_2 and k_3 are a bit more complicated than in the 2D case :

$$k_1 = \frac{a^2 n_1^2 \sin^2\theta + b^2((1+q^2)\cos\theta + pq\sin\theta)^2}{n_1^2 n_2^2}$$

$$k_2 = \frac{(a^2 n_1^2 + b^2 p^2 q^2)\cos^2\theta + b^2(1+p^2)\sin\theta(2pq\cos\theta + (1+p^2)sin\theta)}{n_1^2 n_2^2}$$

$$k_3 = \frac{b^2 pq((1+q^2)\cos^2\theta + (1+p^2)\sin^2\theta)}{n_1^2 n_2^2} + \frac{\frac{1}{2}(-a^2 n_1^2 + b^2(1+q^2+p^2(1+2q^2)))\sin 2\theta}{n_1^2 n_2^2}$$

Finally, the advection term becomes

$$\vec{C} \cdot \hat{\nabla}\psi_y = c\frac{(\phi_x \sin\theta + \phi_y \cos\theta)}{n_2}$$