Computers and Mathematics with Applications 57 (2009) 1089–1101

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Modeling wildland fire propagation with level set methods

V. Mallet [a,b,*], D.E. Keyes [c], F.E. Fendell [d]

[a] INRIA, Paris-Rocquencourt research center, BP 105, 78153 Le Chesnay cedex, France
[b] CEREA, joint laboratory ENPC - EDF R&D, Université Paris-Est, Marne la Vallée, France
[c] Columbia University, Appl Phys & Appl Math, 200 S. W. Mudd Bldg., MC 4701, 500 W. 120th Street, New York, NY, 10027, USA
[d] Northrop Grumman Space Technology, One Space Park, Redondo Beach, CA, 90278, USA

ARTICLE INFO

Article history:
Received 8 August 2007
Received in revised form 13 October 2008
Accepted 17 October 2008

Keywords:
Wildland fire spread
Front propagation
Level set methods
Multivac software
Hamilton–Jacobi equations

ABSTRACT

Level set methods are versatile and extensible techniques for general front tracking problems, including the practically important problem of predicting the advance of a fire front across expanses of surface vegetation. Given a rule, empirical or otherwise, to specify the rate of advance of an infinitesimal segment of fire front arc normal to itself (i.e., given the fire spread rate as a function of known local parameters relating to topography, vegetation, and meteorology), level set methods harness the well developed mathematical machinery of hyperbolic conservation laws on Eulerian grids to evolve the position of the front in time. Topological challenges associated with the swallowing of islands and the merger of fronts are tractable.

The principal goals of this paper are to: collect key results from the two largely distinct scientific literatures of level sets and fire spread; demonstrate the practical value of level set methods to wildland fire modeling through numerical experiments; probe and address current limitations; and propose future directions in the simulation of, and the development of, decision-aiding tools to assess countermeasure options for wildland fires. In addition, we introduce a freely available two-dimensional level set code used to produce the numerical results of this paper and designed to be extensible to more complicated configurations.

1. Introduction

Wildland fire modeling has received attention for decades, due to the sometimes disastrous consequences of large fires, and the tremendous costs of often ineffectual, possibly even counterproductive firefighting [1]. For the practically important scenario of wind-aided fire spread, one seeks a computationally efficient model, useful not only offline (for pre-crisis planning, e.g., placement of access roads, firebreaks, and reservoirs, and scoping of fuel-reduction burning, and post-crisis review, e.g., personnel training, litigation), but also during a crisis (i.e., real-time guidance for evacuation and firefighting). For computational efficiency, such that the benefits of ensemble forecasting [2] are readily accessible from a model, advantage should be taken of the inherent scale separation of: (1) the kilometer-and-larger, landscape-dominated scales of the local atmospheric dynamics; and (2) the one-meter-and-smaller scales of the local combustion dynamics. Even with advanced techniques and access to exceptional contemporary computing facilities, numerical simulations (of turbulent flows) that proceed from fundamental principles are challenged to resolve accurately in real time phenomena with spatial scales spanning much more than two orders of magnitude [H.R. Baum, private communication]. Thus, the feasibility of a direct numerical simulation encompassing the multivaried processes of wildland fire propagation [3] may be decades off

* Corresponding author at: INRIA, Paris-Rocquencourt research center, BP 105, 78153 Le Chesnay cedex, France.
E-mail addresses: vivien.mallet@inria.fr (V. Mallet), david.keyes@columbia.edu (D.E. Keyes), frank.fendell@ngc.com (F.E. Fendell).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.10.089

[4]. Moreover, at least many attempts (albeit usually problematic) at parameterization of sub-grid-scale phenomena in terms of grid-scale variables have been undertaken by meteorologists for cumulus convection, turbulent transport, and radiative transfer. However, meteorologists have extremely limited experience with the parameterization of combustion dynamics for weather-dependent wildland fire spread; even if such parameterization is possible, it remains unknown. Furthermore, data collection in wildland fires is so piecemeal, irregular, and of uncertain accuracy that, for many years to come, the data would better suit reinitialization of a simplistic model than assimilation into an ongoing calculation with a highly detailed model.

Accordingly, in this study, attention is focused on a minimalist treatment of the fire front, idealized as an interface between expanses of burned and unburned vegetation. This treatment is consistent with the typically limited, only gross characterization available for the vegetation at issue, since the vagaries of ignition events are difficult to anticipate, and maintaining an updated inventory for the huge area of wildlands in (say) the USA is daunting. This simplistic interfacial approach to the fire dynamics, easily executed in minutes on a laptop, given the requisite meteorological and other input fields, reserves computational resources for the difficult, more critical, and mostly yet-to-be-undertaken landscape-scale weather forecasting targeted for real-time wildfire applications.

The upshot is that simple persistence models are adopted for the wind field (and thermodynamic variables) in the study undertaken here. Also, attention is limited to a one-way interaction between the meteorology and the fire spread, though future extension to two-way interaction by use of an iterative procedure may be envisioned. Simplistic modeling still may provide the key macroscopic fire behavior sufficiently accurately for practical purposes (including estimates of smoke and pollutant generation), even for circumstances for which the simplification is not formally justifiable. In fact, observational data of wind-aided fire front progression in wildland are today typically sparse, so that not much more than the output of a simplistic model can be meaningfully validated and tuned. Moreover, the use of relevant mathematical methods to perform model selection, to carry out efficient parameter estimation, and to account for the uncertainty in predictions is facilitated by focusing on less detailed models with fewer parameters. In this paper, we mainly address the first step, which is to achieve proper forward simulations.

One of the most widely used models was devised by Rothermel [5] to predict the rate of fire spread, with focus on the head of a wind-aided fire. Because predictions of the Rothermel treatment have been found to be at odds with some observations, efforts to improve this spatially one-dimensional semi-empirical treatment, and to supplement the data upon which it is based, have been undertaken, especially in recent years [6]. Extension from a focus exclusively on the head of the fire, seeks to evolve the configuration of the entire fire perimeter, possibly of multiple fire perimeters. In this study and typically, the fire front, even a moderate fraction of an hour after a localized ignition in fire-prone vegetation, is taken to be a closed curve projected on a plane (the ground may not be flat). Such simulations of fire spread have been performed [7] with the so-called marker technique, which discretizes a front into a set of marker particles, and advances the front through updates of the particle positions. Parenthetically, as a problematic step, the updating by Finney takes each marker on the front to evolve identically to an idealization of how a front evolves from a single isolated ignition site in an unbounded expanse of vegetation, in the presence of a wind. In any case, even though applied projects have supported software development [7], still from a computational point of view, only a few, largely equivalent methodological developments have been undertaken [8, e.g.]. In this paper, we apply *level set methods* [9,10] to calculate fire front evolution.

In Section 2, we introduce wildland fire spread models, especially a semi-empirical, equilibrium-type model proposed in [11] for wind-aided fire spread across surface-layer, chaparral-type, burning-prone vegetation. (In commonly adopted equilibrium-type models, the fire spread rate depends on only the parametric values holding locally and instantaneously, so the fire spread rate is taken to adjust indefinitely rapidly to any temporal and spatial change.) Section 3 provides a brief introduction to level set methods. Section 4 describes the Multivac level set package that has been applied in this paper to the fire spread problem. A quick description of its performance is presented in Section 5. Finally, results of fire spread simulations with different idealized environmental conditions are reported in Section 6.

2. Front propagation functions for wildland fires

Even if theory and/or measurement furnished complete, perfect knowledge of the topography, vegetation, and meteorology at a site at a given time (e.g., furnished the locally pertinent values of all parameters in functional forms capable of representing these three types of input), one still currently possesses very incomplete, imperfect knowledge of the "rules" that would yield the physically observed rate of fire spread from the input. Achieving knowledge of fire spread "rules" sufficiently accurate for practical purposes may well lag emplacing means for observing and collecting exhaustive input data.

As already noted, a fire-growth simulation system such as FARSITE [7] seems unlikely to reach its potential as long as it seeks to describe the rate of fire spread at all orientations to the direction of the sustained low-level ambient wind from spread-rate modeling focused on the direction of the wind [5, e.g.]. On the other hand, posing a different rule for the spread rate at every possible orientation to the wind defeats the goal of simplicity.

2.1. Wind-aided wildland fire spread

Fendell and Wolff [11] addressed this dilemma in developing a model dedicated to wind-aided wildland fires that spread rapidly over level terrain with dry, moderately sparse fuel, taken here to be uniformly distributed to permit concentration

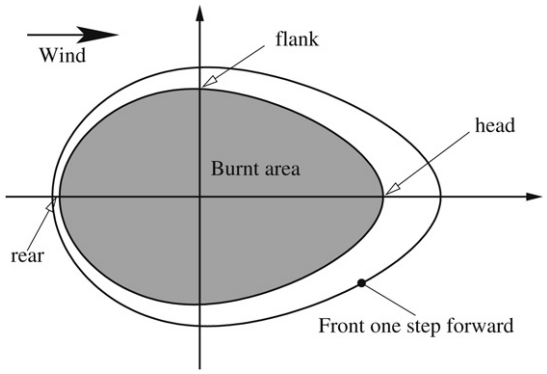

Fig. 1. Fendell and Wolff model introduces velocities at the rear (against the wind), at the head (in the wind direction) and at the flanks [11].

on wind effects. Parenthetically, for consistency with modeling in which the fire front is idealized as an interface moving according to a semi-empirical rule, only a minimal amount of information about the surface-layer fuel is required, mainly the mass loading consumed with fire front passage ("available"-fuel loading).

The Fendell and Wolff model focuses on front velocities at the rear of the front (where propagation is against the wind), at the head of the front (where propagation is with the wind), and on the flanks (where propagation is across the wind direction) — see Fig. 1. The fire spread velocities primarily depend on the wind velocity U. At the rear, the front advances relatively slowly against the oncoming wind, since hot combustion products tend to be blown over an already burned area. The velocity at the rear is denoted $\varepsilon(U)$. At the head, the velocity $h(U)$ is relatively large, since hot combustion products tend to be blown over a yet-to-burn area, in which discrete fuel elements are heated toward ignition by convective-conductive transfer. Both analytic modeling and laboratory experiments have shown that $h(U)$ is roughly proportional to $\sqrt{U}$ [12]. At the flanks, the (spread-aiding) wind component along the normal to the front is zero, but observationally the front advances faster than in the absence of wind, at the rate $f(U)$. As a speculation, a more meticulous treatment would find that, at the nominal flank, the configuration is convoluted, and fire spread is alternately with and against the wind. Of course, were the wind direction constant, limiting attention to the head would seem adequate but, in fact, change in wind direction may (rapidly) result in an interchange of the locations of the flank and head — an interchange sometimes associated with tragic consequence for firefighters.

The velocities (the terminology henceforth adopted, for brevity, in place of fire spread rates) proposed in [11] are

$$\varepsilon(U) = \varepsilon_0 \exp(-\varepsilon_1 U), \qquad f(U) = \varepsilon_0 + c_1 U \exp(-c_2 U), \qquad h(U) = \varepsilon_0 + a\sqrt{U}, \tag{1}$$

where ε_0, ε_1, c_1, c_2 and a are parameters (with readily inferred dimensionality) depending on the mass loading of fuel and other parameters characterizing the fuel bed, but independent of U.

The velocity is then provided at any point on the front through a "trigonometric interpolation":

$$\begin{aligned} F(U,\theta) &= f(U\sin^m\theta) + h(U\cos^n\theta) \quad \text{if } |\theta| \le \frac{\pi}{2}, \\ F(U,\theta) &= f(U\sin^m\theta) + \varepsilon(U\cos^2\theta) \quad \text{if } |\theta| > \frac{\pi}{2}, \end{aligned} \tag{2}$$

where θ is the angle between the wind direction and the normal to the front. The parameters m and n relate to the trigonometric interpolation among the locally fire spread-rate expressions for the head, flanks, and rear. The assignment is empirical, and, at present, based on plausibility (i.e., qualitative recovery of photographic monitoring of field tests). We set $m = 2$. In this paper, parameter n is set to 3 and is found to be significant since it determines the overall shape of the front from the flanks to the head.

To summarize, the velocity is, for all $U \ge 0$ and $\theta \in]-\pi, \pi]$,

$$F(U,\theta) = \varepsilon_0 + c_1 U \sin^2\theta \exp\left(-c_2 U \sin^2\theta\right) + \begin{cases} \varepsilon_0 + a\sqrt{U\cos^n\theta} & \text{if } |\theta| \le \dfrac{\pi}{2} \\ \varepsilon_0 \exp\left(-\varepsilon_1 U \cos^2\theta\right) & \text{if } |\theta| > \dfrac{\pi}{2}. \end{cases} \tag{3}$$

2.2. Simplified model

Based on the numerical experiments carried out with the level set code Multivac (Section 4), the model (3) proposed in [11] has been modified. First, the parameter n has been set to 3 instead of 1. Second, the model has been simplified without

losing its main features, primarily the overall shape of the fire front. The new model reads

$$\begin{aligned} F(U,\theta) &= \varepsilon_0 + a\sqrt{U\cos^n\theta} && \text{if } |\theta| \le \frac{\pi}{2}, \\ F(U,\theta) &= \varepsilon_0(\alpha + (1-\alpha)|\sin\theta|) && \text{if } |\theta| > \frac{\pi}{2}, \end{aligned} \tag{4}$$

where $\alpha \in [0, 1]$ is the ratio between the velocity at the rear ($\alpha\varepsilon_0$) and the velocity at the flanks (ε_0). Velocities at the rear and at the flanks no longer depend on the wind, since their dependence on the wind speed is hard to model accurately and has little impact on the overall front location. The velocity at the head is essentially the same as in the "full" model (3).

The simplified model is easier to tune, either via direct trials or with systematic methods for parameter estimation (which may require derivatives of the model with respect to its parameters). All results in this paper are for the simplified model. However, results for the "full" model would appear roughly the same.

3. Level set and fast marching methods

First introduced in [9], level set methods are Eulerian schemes for tracking fronts propagating according to a given speed function. In this section, we explain basic features of the level set methods used for fire spread modeling.

3.1. Mathematical basis and technique

3.1.1. Definitions

Assume the front evolves from the initial time $t = 0$ to the final time $t = T_f$. For all $t \in [0, T_f]$, the front at time t is the set of points $\Gamma(t)$ in $\mathbb{R}^N$ where $N = 2$ in case of the 2D simulations we carry out in this paper. We define $\Gamma_0 = \Gamma(0)$ as the initial front.

For all $t \in [0, T_f]$, each point $X \in \Gamma(t)$ with a well-defined normal moves in the direction normal to the front with a given speed $F(X, \Gamma, t)$. Notice that F may depend on the position, on the time and on local properties of the front itself (certainly the normal direction, not always defined, and possibly the local curvature or other properties).

The problem is to approximate Γ, given Γ_0 and F.

3.1.2. Strategy

The main idea is to evolve a function $\varphi : \mathbb{R}^N \times [0, T_f] \to \mathbb{R}$ such that

$$\forall t \in [0, T_f] \qquad \Gamma(t) = \left\{x \in \mathbb{R}^N / \varphi(x, t) = 0\right\}. \tag{5}$$

φ is called the level set function. At any time, the zero level set of φ is the front itself. φ could be any smooth (at least Lipschitz continuous) function satisfying Eq. (5).

It can be shown that φ obeys the equation

$$\forall x \in \mathbb{R}^N \quad \forall t \in [0, T_f] \qquad \varphi_t(x, t) + F(x, \varphi(\cdot, t), t)\|\nabla_x \varphi(x, t)\|_2 = 0, \tag{6}$$

where the velocity F is now defined everywhere in $\mathbb{R}^N$ and depends on the front through its dependence upon φ, and where $\|\cdot\|_2$ is the 2-norm. Details may be found in [10].

Practical issues (e.g., initialization of φ) make it convenient to replace the level set function φ with the signed distance to the front.

Then, if d is the Euclidean distance on $\mathbb{R}^N$, we define, for any given curve Υ, the distance d_Υ to Υ:

$$\forall x \in \mathbb{R}^N \qquad d_\Upsilon(x) = \min\{d(x, P)/P \in \Upsilon\}. \tag{7}$$

Hence the signed distance φ for all $x \in \mathbb{R}^N$ and $t \in [0, T_f]$:

$$\varphi(x, t) = \begin{cases} d_{\Gamma(t)}(x) & \text{if } x \text{ lies outside the front } \Gamma(t) \\ -d_{\Gamma(t)}(x) & \text{if } x \text{ lies inside the front } \Gamma(t). \end{cases} \tag{8}$$

Recall that $\varphi(\cdot, 0)$ is known as well as Γ_0; $\varphi(0)$ is the signed distance to Γ_0:

$$\forall x \in \mathbb{R}^N \quad \varphi(x, 0) = \begin{cases} d_{\Gamma_0}(x) & \text{if } x \text{ lies outside the front } \Gamma_0 \\ -d_{\Gamma_0}(x) & \text{if } x \text{ lies inside the front } \Gamma_0. \end{cases} \tag{9}$$

Eqs. (6) and (9) define the initial-value problem that is to be solved. Zero level sets of φ yield the front points.

This nonstationary problem involves the Hamilton–Jacobi equation (6). There may be multiple solutions to this equation. P.-L. Lions and M. G. Crandall defined the so-called "viscosity solution" of Hamilton–Jacobi equations [13,14], which turns out to be the unique physical solution for which we search. Under given assumptions (mainly on the speed function F), existence and uniqueness of the viscosity solution of the Eq. (6), with some initial conditions, can be proved.

3.2. Advantages and disadvantages of level set methods

Several methods may be relevant to simulate the propagation of fire fronts. One may want to use marker techniques, in which the front is discretized by a set of points. At each time step, each point is advanced according to the speed function. This Lagrangian methodology leads to low-cost computations, but requires care in the handling of topological changes.

Volume-of-fluid methods [15] represent the front by the amount of each grid-cell that is inside the front. In each cell, the front is approximated by a straight line (horizontal or vertical, in most methods). Such methods can deal with topological changes, but the front representation can be inaccurate. In wildland fire spread, the direction normal to the front is crucial because of the wind-direction-dependent speed function (see Section 2).

Level set methods automatically deal with topological changes that occur in wildland fire spread, such as fronts merging and front convergence (in connection with unburnt "islands"). The level set description enables a fair estimate of the normal to the front, making it well suited to the fire propagation problem.

However, level set methods have disadvantages. First, they embed the front in a higher-dimensional space. Helpfully, the narrow band level set method [16] is an efficient algorithm which almost decreases the problem dimension by one. Moreover, when it can be used, the fast marching method [17] provides a highly efficient algorithm.

The main reservation may be the lack of proof of convergence of numerical schemes for certain problems. For a given class of speed functions, the problem (6), with some initial conditions, may routinely be solved numerically [18]. However, no proof of convergence in mesh parameter or time step is yet available for some situations.

3.3. Quick review of numerical approximations

Numerical approximation to solutions of Hamilton–Jacobi equations is closely related to numerical approximation to hyperbolic conservation laws.[1] The point is to introduce a numerical Hamiltonian to approximate the Hamiltonian $H = F \cdot \|\nabla_x \varphi\|_2$.

Crandall and Lions have proven that, for given Hamiltonians and initial conditions, a consistent, monotonic and locally Lipschitzian numerical Hamiltonian yields a solution that converges to the viscosity solution. Formal results may be found in [18] and [19].

In one dimension, $\varphi_t + H(\nabla_x \varphi) = 0$ may lead to the following approximation:

$$\varphi_j^{n+1} = \varphi_j^n - \Delta t\, g\left(\frac{\varphi_{j+1} - \varphi_j}{\Delta x}, \frac{\varphi_j - \varphi_{j-1}}{\Delta x}\right). \tag{10}$$

For instance, if the Hamiltonian is not convex, the Lax–Friedrichs scheme may be used [18]; then, the numerical Hamiltonian is

$$\forall a, b \in \mathbb{R} \qquad g(a, b) = H\left(\frac{a+b}{2}\right) - \vartheta \frac{b-a}{2}, \tag{11}$$

where the monotonicity is satisfied on $[-R, R]$ if $\vartheta = \max_{-R \le a \le R} |H'(a)|$.

Several schemes have been developed, from simple and efficient schemes as that of Engquist–Osher to high-order essentially nonoscillatory schemes [20].

3.4. Overview of complexity issues

Let the mesh (in $\mathbb{R}^N$) be orthogonal with M points along each direction. Assume that the front is described by $\mathcal{O}(M^{N-1})$ points. The narrow band level set method makes it sufficient to update the level set function only in a narrow band (of width k) around the front. For each time step, the complexity of the algorithm is therefore $\mathcal{O}(kM^{N-1})$.

For an explicit temporal discretization the number of iterations is related to the Courant–Friedrichs–Lewy condition. Along x, the Courant number must be less than 1:

$$\frac{\max\left|H'\right| \Delta t}{\Delta x} \le 1. \tag{12}$$

Usually, controlling the accuracy of approximation is subordinate to space discretization, which means that the time step is adjusted so that the Courant number is taken close to 1.

Calculations may sometimes be sped up by reformulating the level set problem as a stationary problem. This leads to the so-called fast marching method [17]. Nevertheless, restrictions on the Hamiltonian prevent the use of this technique for some applications. The work of Sethian and Vladimirsky has overcome some limitations [21], but restrictive conditions still remain.

[1] Notice that, from Eq. (6), φ_x satisfies a hyperbolic conservation law in the one-dimensional case.

4. Code

4.1. Introduction to the Multivac level set package

Multivac is a level set package freely available (under the GNU GPL license) at http://vivienmallet.net/fronts/. It is designed to be both efficient and extensible, so that it may be used for a large range of applications. To achieve these goals, Multivac is built as a fully object-oriented library in C++.

Multivac was designed independently of the fire spread application described herein, but easily enabled fire spread simulations, and is presently distributed with fire-spread-motivated functions. It has also been used in modeling the growth of Si-based nanofilms [22] and image segmentation.

The latest stable version available at the time of submission is Multivac 1.10.

4.2. Structure

The modularity of Multivac comes from its object-oriented framework, in which the main components of a simulation have been split into an equal number of objects. A simulation is defined by the following objects:

- the *mesh*;
- the *level set function*;
- the *velocity*, which provides the propagation rate of the front according to its position, its normal, its curvature, and the time;
- the *initial front*;
- the *initializer*, which manages first initializations and initializations required by level set methods (e.g., the narrow band reconstruction);
- the *numerical scheme*, which advances the front in time;
- the *output management*.

For each item, a set of classes[2] with a common interface is available. For instance, several speed (i.e., propagation rate) functions are available through several classes, e.g., `CConstantSpeed` or `CFireModel`. All speed functions have the same interface, which allows users to define their own speed function on the same basis. The user principally provides speed rates as a function of the position, the time, the normal to the front and the curvature (these values are computed by Multivac itself).

4.3. Calling sequence

The whole is managed by an object of the class `CSimulator`. This object simply calls the *initializer* to perform the first initializations. Then it manages the loop in time (or iterations, in the case of the fast marching method) into which the *numerical scheme* is called to advance the front. The *initializer* is called again to reinitialize the signed distance function for the new step, and the object dedicated to post-processing requirements is called to save any needed data.

In each step, objects communicate with one another through methods (i.e., functions) of their interface. For example, the *velocity* object provides speed rates to the *numerical scheme*.

A simplified overview of the architecture is shown in Fig. 2.

4.4. Overview of available classes

Multivac package (version 1.10) includes several classes which are listed in Table 1.

4.5. Other strengths, limitations and future work

Multivac takes advantage of C++ exceptions to track errors, and several debugging levels are defined, from a safe mode, in which all is checked, to a fast mode, in which performance is the primary concern.

There are currently two main limitations. First, Multivac deals only with uniform orthogonal meshes. However, extensions of level set methods to unstructured meshes exist (e.g., [24]) and they could be implemented within the Multivac framework. Adaptively refined meshes are also accommodated with additional mathematical complexity, though the implementation effort would be substantial. Second, Multivac deals only with two-dimensional problems.

Work is planned to allow inverse modeling (parameter estimation based on data assimilation) within the framework of Multivac. The main idea is to replace the class `CSimulator` with a class dedicated to inverse modeling. Preliminary results show the framework extendibility, but this capability is not yet available in distributed versions. Future versions should include this feature, based on an innovative method for integrating sensitivities along with the front itself.

[2] A class is a user-defined type, in the manner of structures in C. Classes encapsulate data (called attributes) and functions (called methods).

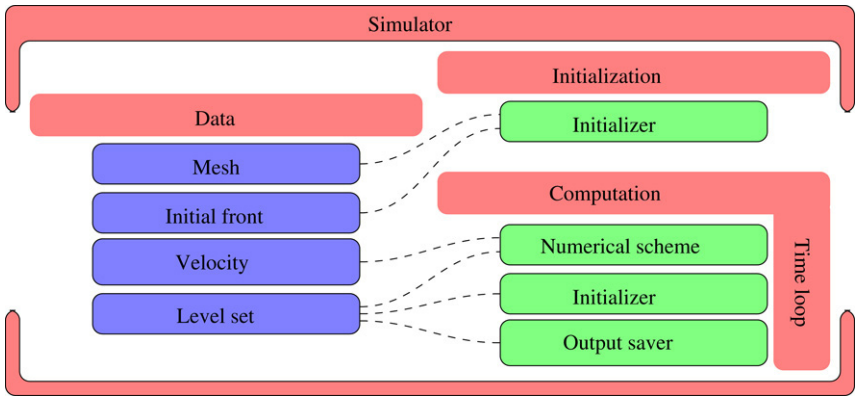

Fig. 2. Simplified overview of the architecture: the main communications between the objects within the class `CSimulator`.

Table 1
Basic classes available in Multivac 1.10.

Category	Available classes
Mesh	Orthogonal mesh
Level set function	Defined on an orthogonal mesh
Velocity	Constant speed Piecewise constant speed Fire model Simplified fire model Image intensity Image gradient
Initial front	Circle Two or three circles One or two circles with an island inside Front defined by any set of points
Initializer	Basic initialization (no velocity extension) Extends the velocity with the closest neighbor on the front
Numerical scheme (narrow band)	Engquist–Osher, first order Lax–Friedrichs, first order Engquist–Osher, ENO, second order Chan-Vese algorithm [23]
Numerical scheme (fast marching)	Engquist–Osher, first order

Table 2
Simulation test-case.

Data	Value	Comment
Domain	$\Omega = [0, 3] \times [0, 3]$	
Initial front	Circle	
Circle center	$(x_c, y_c) = (1.5, 1.5)$	Domain center
Initial circle radius	$r_{\text{initial}} = 0.5$	
Final circle radius	$r_{\text{final}} = 0.9$	
Velocity	$F = 1.0$	Constant
Duration	$T_f = 0.4$	
Time step	$\Delta t = 10^{-4}$	

5. Complexity and convergence studies

5.1. Convergence studies

In this section, we report convergence studies that are necessary to validate the code. As in [25], tests are carried out for a circle that expands in time with a unitary velocity. Details of the simulation are summarized in Table 2.

We introduce three norms. The first is

$$e^1_{\text{spatial}} = |r_{\text{simulated}} - r_{\text{final}}|, \tag{13}$$

Table 3
Errors versus spatial discretization.

$\Delta x = \Delta y$	$N_x = N_y$	$e^1_{\text{spatial}}(\times 10^3)$	$e^2_{\text{time}}\ (\times 10^3)$	$e^\infty_{\text{time}}\ (\times 10^3)$
0.01	301	1.634	1.753	2.377
0.005	601	0.855	0.901	1.191
0.0025	1201	0.460	0.474	0.600
0.00125	2401	0.244	0.247	0.299

Table 4
Timings versus spatial discretization.

$\Delta x = \Delta y$	$N_x = N_y$	Timings (s)
0.03	101	0.4
0.015	201	0.9
0.01	301	1.6
0.0075	401	2.6
0.006	501	4.0
0.005	601	5.6
0.004285714	701	7.4
0.00375	801	9.5
0.003333333	901	11.9
0.003	1001	14.1

where $r_{\text{simulated}}$ is the simulated radius, estimated as follows:

$$r_{\text{simulated}} = \frac{1}{\text{card}(\Gamma_d)} \sum_{(x,y)\in\Gamma_d} \text{d}\left((x,y),(x_c,y_c)\right), \tag{14}$$

where Γ_d is the discretized front as returned by the simulation (at time T_f), card is the cardinal (number of points) and d is the Euclidian distance.

Additionally, if $T_{\text{true}}(x,y)$ is the time at which the front is supposed to reach the point (x,y):

$$e^2_{\text{time}} = \sqrt{\frac{1}{\text{card}(\Gamma_d)} \sum_{(x,y)\in\Gamma_d} \left(T_f - T_{\text{true}}(x,y)\right)^2}. \tag{15}$$

The last norm is an infinity norm:

$$e^\infty_{\text{time}} = \max_{(x,y)\in\Gamma_d} \left|T_f - T_{\text{true}}(x,y)\right|. \tag{16}$$

Table 3 shows results for the first-order Engquist–Osher scheme with the narrow-band method. The width of the band is 12 cells and the front lies within a band whose width is 6 cells.

The first-order Lax–Friedrichs scheme and the second ENO Engquist–Osher scheme were also checked successfully. As for the second-order scheme, the full-matrix method (without the narrow-band restriction) was used because the front reconstruction destroys the second-order accuracy.

5.2. Complexity issues

Multivac was compiled under Linux with GNU/g++ 3.3, and the reference simulation (see Table 2) was launched on a Pentium 4 running at 2.6 GHz. The width of the narrow band was 12 cells and the width of the inner band, in which the front lies, was 6 cells. If $N_x = N_y = 1001$ (one million cells), the 4,000 iterations were achieved in 14 s.

The complexity of the narrow-band level set method is close to $\mathcal{O}(N)$, where $N = N_x = N_y$. Table 4 shows that linear complexity of the method is not observed. Instead, the complexity seems to be $\mathcal{O}(N^2)$. This is the complexity of the suboptimal algorithm currently used to rebuild the front. Moreover, the number of front reconstructions increases with the mesh refinement since the width of the narrow band does not change.

6. Applying level set methods to fire spread applications

6.1. Method and numerical scheme

The speed function (3) introduced in the level set Eq. (6) provides a Hamiltonian with nontrivial dependencies. Because of these dependencies (particularly the non-convexity of the Hamiltonian), neither the fast marching method nor its extension to anisotropic problems can be applied. The narrow-band level set method is more relevant.

Table 5
Parameters and their default values.

Parameter	Value	Parameter	Value
n	3	Domain	$\Omega = [0, 3] \times [0, 3]$
U	100	Initial front	Circle
a	0.5	Circle center	(1.5, 1.0)
ε_0	0.2	Initial circle radius	$r_{\text{initial}} = 0.5$
α	0.5	Velocity	$F = 1.0$
Δx	3×10^{-3}	Duration	$T_f = 0.1$
Δy	3×10^{-3}	Time step	$\Delta t = 5 \times 10^{-5}$
Δt	10^{-4}	Spatial discretization	$N_x = N_y = 1001$
T_f	0.1		

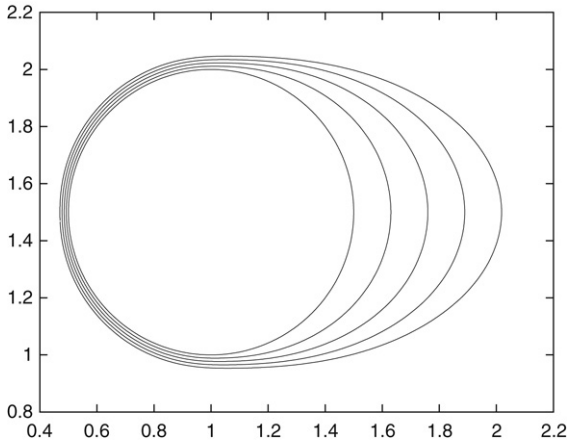

Fig. 3. Basic simulation described by Table 5.

A highly accurate numerical scheme is not required for the investigations reported here. The discrepancies between the numerical simulation and the exact solution should be considered in the context of other approximations: the model itself is simplistic; input parameters such as wind speed or fuel density are typically not accurately estimated; the location of the initial front introduces further uncertainties. A first-order scheme suffices for our purposes.

Since the Hamiltonian involved is not convex with respect to spatial derivatives of the level set function, the first-order Lax–Friedrichs scheme (refer to Eq. (11)) is well suited. To minimize introduction of diffusivity, a local Lax–Friedrichs scheme may be used as well.

As previously advocated, the timestep Δt is chosen according to the Courant–Friedrichs–Lewy condition (12):

$$\Delta t = \frac{\alpha \Delta x}{\max |H'|}, \tag{17}$$

where $\alpha \leq 1$; α is not kept constant in the tests that we undertake. Nevertheless, the Courant–Friedrichs–Lewy condition is estimated at every iteration with an (*a priori*) approximation to $\max \left|H'\right|$ along x and y, which leads to:

$$\Delta t \leq \frac{\Delta x}{a(\frac{n}{2} + 1)\sqrt{U}}. \tag{18}$$

The main characteristics of the simulation, including model parameters (refer to Eq. (3)), are gathered in Table 5.

6.2. Results

The simulation described by Table 5 is shown in Fig. 3. The figure shows snapshots of the front, initially circular, at subsequent times, under a constant-magnitude wind blowing from left to right. Since thoroughly burnt areas cannot be burnt again (on the time scale of the simulation), the area enclosed by the front increases with time. The rear, the flanks and the head of the front are clearly identifiable.

The reference simulation is slightly modified to show the ability to deal with multiple fronts — Fig. 4. It demonstrates the capability to deal with the merging of fronts (two main fronts), and to deal with the so-called islands, i.e., an unburnt area surrounded by a burnt area.

In Figs. 5 and 6, we use the same parameters as in Table 5 but $\Delta t = 2.5 \times 10^{-5}$, and a depends on x, a being equal to 0.5 if $x < 1.7$, and $a = 0.25$ (Fig. 5) or $a = 1.0$ (Fig. 6) if $x > 1.8$, and a being linearly interpolated for intermediate

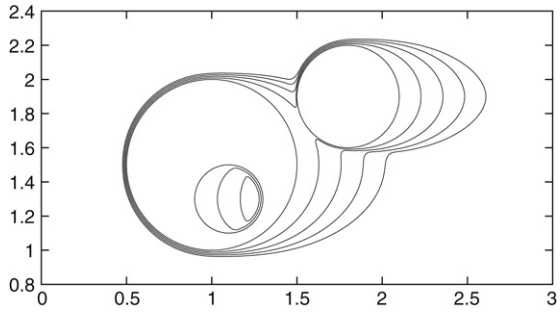

Fig. 4. Two main fronts merge, and an island – the unburnt area within the biggest front – is burnt.

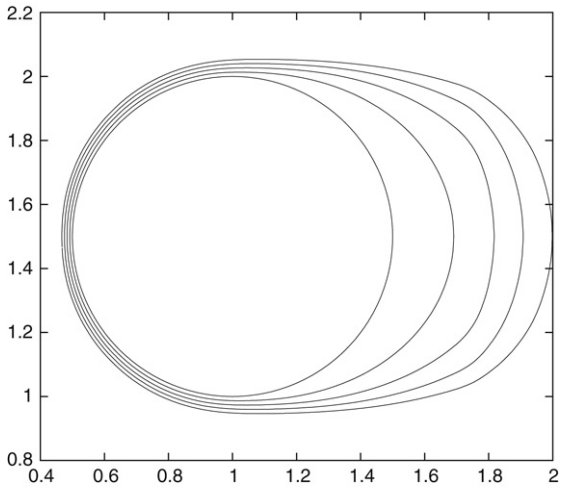

Fig. 5. The front slows down at the head for $a = 0.25$ if $x > 1.8$. The final time is changed to $T_f = 1.5$.

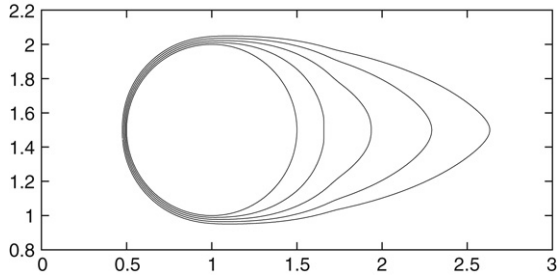

Fig. 6. The front advances faster at the head for $a = 1.0$ if $x > 1.8$.

values of x. Since a takes into account the available fuel loading, these two simulations roughly show the influence of the inhomogeneous available fuel loading, should it increase (Fig. 5) or decrease (Fig. 6). The inherent decrease of the radius of curvature at the head for a constant-direction wind suggests that some vacillation of wind direction contributes when the head broadens under otherwise uniform conditions (including uniform fuel load).

Fig. 7 shows the impact of a rotating wind direction. If north is toward the top of the figure, then the wind is oriented first west-to-east and tends later to south-to-north.

The next two Figs. 8 and 9 show the behavior of two fronts subject to a simple-counterflow wind, i.e., a wind defined as:

$$\overrightarrow{U}(x, y) = \begin{pmatrix} -ux \\ uy \end{pmatrix} \tag{19}$$

where u is set to 100. A counterflow exemplifies wind conditions well suited for setting a backfire, to pre-burn the vegetation in the path of a wind-aided fire.

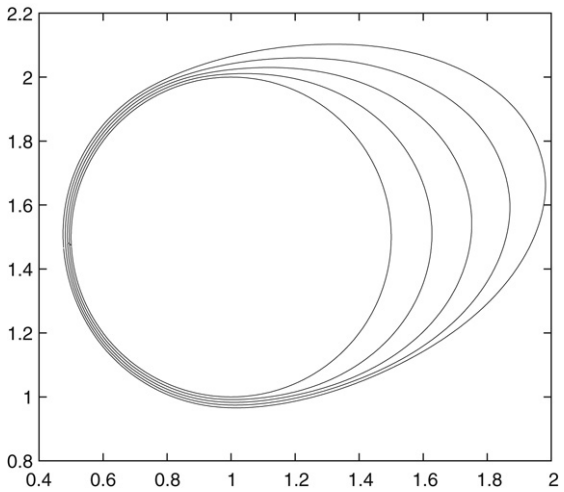

Fig. 7. Same as the reference simulation, but with a changing wind direction.

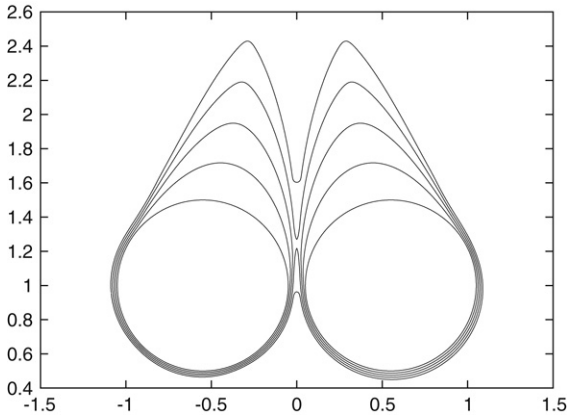

Fig. 8. Evolution of the merged front from initially two mirror-image fronts, one to each side of the stagnation line for a converging x-component wind, but both to one side of the stagnation line for a diverging y-component wind.

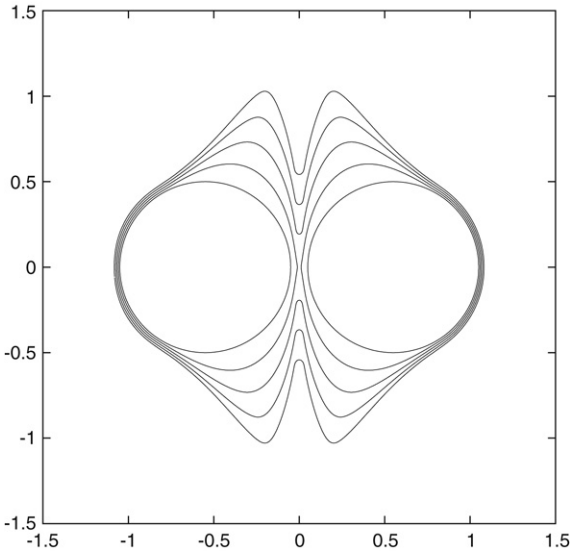

Fig. 9. Evolution of the merged front from initially two mirror-image fronts, here symmetrically sited relative to a simple counterflow wind.

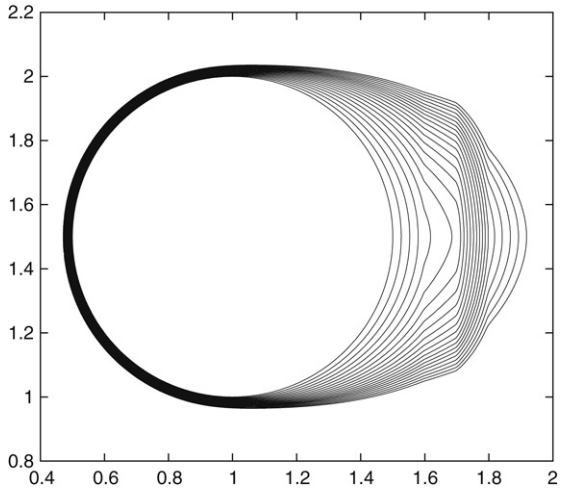

Fig. 10. Taking into account topography: the front propagates over an idealized hill.

The last Fig. 10 shows a front that propagates over an idealized hill. Where the slope is positive (between $x = 1.6$ and $x = 1.7$), the fire front typically advances faster. Downhill the front typically slows down [26, pp. 94–97]. The speed function is therefore modified to take into account the slope s:

$$F_{\text{topography}} = F \times \mathrm{e}^{2s}, \tag{20}$$

where s is in radians.

7. Conclusion and future prospects

A semi-empirical, equilibrium-type fire spread rate has been used to model a wind-aided fire front propagation across wildland surface vegetation. In this formulation, the rate depends primarily on the wind speed, and the angle between the wind direction and the normal to the fire front (idealized as a one-dimensional interface). In scenarios arising in practice, the front may consist of several closed curves (possibly nested) that can merge as they propagate.

Level set methods appear capable of treating the model formulated to simulate wildland fire evolution. They readily treat the topological changes that may occur to the fire front, and they are known to converge to the physical solution of the front tracking problem.

They were applied via the Multivac package. This open-source library is designed to handle a wide range of applications without loss of computing performance. It includes several algorithms and numerical schemes, primarily for the narrow-band level set method, which is more computationally efficient than the full level set method.

A possible direction for future work is to focus on parameter estimation within the context of the simple model illustrated here. A cost is introduced to measure the distance between the simulated front and ground, aerial, and/or satellite observations. The discrepancy between the simulated and observed positions of the front may be based either on the front arrival times (at monitored locations), or on distances between the simulated front and the monitored locations (at arrival times). For gradient-based optimization methods, the main challenge is to compute the derivative of the cost function with respect to the parameters. An adjoint code being difficult to construct, alternative methods should be sought.

This work could help guide fire-control tactics. The objective function would then penalize front advance into societal assets, and penalize the cost of the firefighting activity. The parameters would be the model variables modifiable by firefighting countermeasures. The links between this optimization problem and shape optimization should be investigated.

Acknowledgments

The support of the National Science Foundation under grant CCF-03-52334 and the US Department of Agriculture Forest Service under grant SFES 03-CA-11272169-33, administered by the Riverside Forest Fire Laboratory, a research facility of the Pacific Southwest Research Station, is gratefully acknowledged. The authors are particularly indebted to Dr. Francis M. Fujioka of Riverside for enhancing the relevance of our research through his technical advice, and for his support for the training of summer students in the computational technology of fire spread and fire imaging.

References

[1] Stephen Pyne, Tending fire — Coping with America's Wildland Fires, Island Press, 2004.
[2] T.N. Palmer, G.J. Shutts, R. Hagedorn, F.J. Doblas-Reyes, T. Jung, M. Leutbecher, Representing model uncertainty in weather and climate prediction, Annual Review of Earth and Planetary Sciences 33 (2005) 163–193.
[3] J.L. Coen, Encyclopedia of Atmospheric Sciences, Academic Press, 2003, 2,586–2,596.
[4] M.A. Jenkins, T.L. Clark, J.L. Coen, Forest fires — Behavior and Ecological Effects, Academic Press, 2001, 257–302.
[5] R.C. Rothermel, A mathematical model for predicting fire spread in wildland fuels, USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden, Research Paper, INT-115, 1972.
[6] D.W. Carlton, The impact of changing the surface fire spread model, JFSP, 2003.
[7] Mark A. Finney, FARSITE: Fire area simulator-model development and evaluation, Technical report USDA Forest Service, 1998.
[8] J.C.S. André, J.M. Urbano, D.X. Viegas, Forest fire spread models: the local quasi-equilibrium approach, Combustion Science and Technology 178 (12) (2006) 2,115–2,143.
[9] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1988) 12–49.
[10] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.
[11] F.E. Fendell, M.F. Wolff, Forest Fires — Behavior and Ecological Effects, Academic Press, 2001, Chapter Wind-aided fire spread, 171–223.
[12] M.F. Wolff, G.F. Carrier, F.E. Fendell, Wind-aided fire spread across arrays of discrete fuel elements. II. Experiment, Combustion Science and Technology 77 (1991) 261–289.
[13] P.-L. Lions, Generalized Solutions of Hamilton-Jacobi Equations, in: Research Notes in Mathematics, vol. 69, Pitman, 1982.
[14] M.G. Crandall, P.-L. Lions, Viscosity solutions of Hamilton-Jacobi equations, Transactions of the American Mathematical Society 277 (1983) 1–42.
[15] W.F. Noh, P. Woodward, A simple line interface calculation, in: Fith International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, 1976, pp. 330–340.
[16] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, Journal of Computational Physics 118 (2) (1995) 269–277.
[17] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proceedings of the National Academy of Sciences 93 (4) (1996) 1591–1595.
[18] M.G. Crandall, P.-L. Lions, Two approximations of solutions of Hamilton-Jacobi equations, Mathematics of Computation 167 (43) (1984) 1–19.
[19] P.E. Souganidis, Approximation schemes for viscosity solutions of Hamilton-Jacobi equations, Journal of Differential Equations 59 (1) (1985) 1–43.
[20] S. Osher, C.-W. Shu, High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations, SIAM Journal on Numerical Analysis 28 (1991) 907–922.
[21] J.A. Sethian, A. Vladimirsky, Ordered upwind methods for static Hamilton-Jacobi equations, Proceedings of the National Academy of Sciences 98 (20) (2001) 11,069–11,074.
[22] Anh-Vu Phan, Vivien Mallet, Modeling the growth of Si-based nanofilms by coupling the boundary contour method and level set Multivac, in: Proceedings of the Tenth International Conference on Composite Engineering, 2003, pp. 575–576.
[23] Tony F. Chan, Luminita A. Vese, Active contours without edges, IEEE Transactions on Image Processing 10 (2) (2001) 266–277.
[24] T.J. Barth, J.A. Sethian, Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains, Journal of Computational Physics 145 (1998) 1–40.
[25] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, Journal of Computational Physics 148 (1998) 2–22.
[26] R.H. Luke, A.G. McArthur, Bushfires in Australia, Australian Government Publishing Service, Canberra, 1978.