Advances in Computational Mathematics
https://doi.org/10.1007/s10444-018-9650-4

Numerical modeling of wildland surface fire propagation by evolving surface curves

Martin Ambroz[1] · Martin Balažovjech[1] · Matej Medl'a[1] · Karol Mikula[1,2]

Received: 1 December 2017 / Accepted: 9 November 2018 / Published online: 26 November 2018

Abstract

We introduce a new approach to wildland fire spread modeling. We evolve a 3-D surface curve, which represents the fire perimeter on the topography, as a projection to a horizontal plane. Our mathematical model is based on the empirical laws of the fire spread influenced by the fuel, wind, terrain slope, and shape of the fire perimeter with respect to the topography (geodesic and normal curvatures). To obtain the numerical solution, we discretize the arising intrinsic partial differential equation by a semi-implicit scheme with respect to the curvature term. For the advection term discretization, we use the so-called inflow-implicit/outflow-explicit approach and an implicit upwind technique which guarantee the solvability of the corresponding linear systems by an efficient tridiagonal solver without any time step restriction and also the robustness with respect to singularities. A fast treatment of topological changes (splitting and merging of the curves) is described and shown on examples as well. We show the experimental order of convergence of the numerical scheme, we demonstrate the influence of the fire spread model parameters on a testing and real topography, and we reconstruct a simulated grassland fire as well.

Keywords Curve evolution · Surface curve · Topological changes · Wildland fire modeling · Geodesic curvature · Normal curvature

Mathematics Subject Classification (2010) 35R01 · 65M08 · 53Z05 · 68U20

1 Introduction

Wildland fires cause considerable ecological and economical loss to the wildland environment. In the twentieth century, the wildland fires presented a serious threat

Communicated by: Jean-Frédéric Gerbeau

✉ Martin Ambroz
ambroz.martin.ml@gmail.com

Extended author information available on the last page of the article.

mainly in Australia, North America, and South Europe. Due to global warming, this threat has spread in other areas in the new millennium. To avoid future loss caused by fire, there is an effort to predict fire behavior, e.g., to predict the key places in order to slow the fire down and to put it down efficiently, safely, and with the smallest possible loss.

Nowadays, there exist several mathematical fire propagation models and simulation softwares. We can divide them according to the way they simulate the fire propagation processes to (quasi-) physical, e.g., Wildland Fire Dynamics Simulator, and (quasi-) empirical, e.g., FARSITE. In general, the simulations based on the physical models are more time-consuming, compared to those of the empirical models. In an effort to simplify the physical model of the fire development, the empirical laws of fire behavior are a suitable choice. A number of wildland fire simulation models are built in this way (Calchas, Farsite, Prometheus, SiroFire, Vesta, Flire, Ignite, Firemap, Firestation, Pyrocart) [36].

The Wildland-Urban Interface Fire Dynamics Simulator (WFDS) [20] is an extension of the Fire Dynamic Simulator [19]. This two- and three-dimensional model is based on a formulation of the equations of motion for buoyant flows. The combustion is considered using the stoichiometric relation only for the gas phase, arising from evaporation and pyrolysis. The low-speed, thermally driven flow with an emphasis on smoke and heat transport from fire is obtained as a numerical solution of an appropriate form of the Navier-Stokes equations. However, the computational time is a big disadvantage, taking dozens of hours to obtain 100 s of simulation [36]. Perhaps the best known and used software, FARSITE [12], is based on the evolution of a two-dimensional discretized curve as a boundary between the burnt and unburned fuel (fire perimeter). The information about fuel, topography, and weather is necessary to compute the fire propagation. Computations are based on existing mathematical models like Rothermel's model for surface fire spread, Van Wagner's model for crown fire initiation, Rothermel's model for crown fire spread, and Nelson's dead fuel moisture model. Since the fuel models are built for the climate in the USA, application of FARSITE is not straightforward in any other country. In Australia and Canada, other softwares, Prometheus and SiroFire, were developed. They are based on their own climate, adjusted empirical laws, and fire perimeter evolution algorithms [9]. The Greek simulator Calchas [37] considers terrain slope, vegetation parameters, and current meteorological conditions. The evolution is based on a variation of the Dijkstra algorithm and a fuzzy/neural system to estimate the combined effect of the mentioned factors. As a result, the spatial curve is evolving in the normal direction. Firestation [18] is a semi-empirical simulator driven by the terrain slope, fuel describing parameters and wind direction and velocity. Moreover, two models for wind field simulations are included. A simple and fast one, based on the mass conservation principle, and a more complex model based on the three-dimensional Navier-Stokes equations. Similarly to FARSITE, the curve is evolving by the Huygens principle, considering ellipses with variable semi-major and semi-minor axis. Vesta [1] is a simulator considering the terrain slope, the wind velocity and direction, and fuel properties (the vegetation type and the slope orientation). It offers two different evolution models. The first one requires no parameter on input, since all values are given. To obtain the model parameters, hundreds of fire situations were simulated and evaluated on a physics-based simulator FIRETEC. The second one was developed for testing purposes and basically is driven by the wind velocity and direction.

In this paper, we develop a new wildland fire propagation model. The model is based on the surface curve evolution and empirical principles of fire propagation on Earth topography. In the mathematical literature, there exist a number of studies about the evolution of planar and surface curves with many applications (see, e.g., [2–8, 11, 13, 14, 23–28, 31, 35]). We distinguish two main approaches to handle the curve evolution problems, the so-called Lagrangian (direct) approach (see, e.g., [2, 11, 24]) and the so-called Eulerian (level-set) approach (see, e.g., [31, 35]). In the Eulerian level-set approach, one solves the problem of curve evolution in a 2-D computational domain which is usually discretized by a uniform grid and the number of discrete unknowns is proportional to the number of the 2-D grid points. The evolving curve is then obtained implicitly, as the zero isoline of 2-D + time level set function. In the Lagrangian approach, one evolves directly the curve discretization points, so it is spatially a 1-D problem and thus computationally much more simple and faster than the level-set method. However, the Lagrangian approaches need the so-called tangential grid point redistribution [4, 13, 14, 23–28] and an efficient algorithm for the detection and treatment of topological changes during the evolution [3, 5–8, 27], which are automatically handled by the level-set method [31, 35]. When the Lagrangian methods are tangentially stabilized and are able to treat the topological changes quickly, they represent really efficient approaches to 2-D or surface curve evolution. In the fire modeling literature, the simulators based on the Lagrangian approach are denoted as vector-based (Calchas, Farsite, Prometheus, SiroFire) and the simulators based on the Eulerian are called raster-based (Vesta, Flire, Ignite, Firemap, Firestation, Pyrocart) [36].

In our Lagrangian forest fire propagation model, the fire perimeter is represented by a surface curve, although it is evolved numerically as a projected planar curve. The surface curve evolution is driven by the empirical mathematical model, which considers variable fuel rate of spread (ROS), wind speed and direction, topography slope, and the fire perimeter shape with respect to the topography (geodesic and normal curvatures). Such a complex mathematical model is new in the literature devoted to wildland fire modeling as well as the curve evolution. This, together with a new and efficient numerical treatment of the mathematical model, represents the main contributions of our paper, which is organized as follows. We define a fire perimeter as a surface curve in Section 2. We design the outer normal velocity and derive the external driving forces for the surface curve evolution in Section 3. We also employ the tangential velocity in order to obtain an asymptotically uniform grid point redistribution. Numerical discretization is presented in Section 4. In Section 5, we discuss the efficient treatment of topological changes in the fire perimeter, splitting and merging of the curves. In the last Section 6, the results of our model on representative examples are presented.

2 A surface curve

In this paper, we will use the so-called Lagrangian approach for the evolution of a 3-D surface curve, representing a fire perimeter. For numerical computations, we will use its projection on a 2-D planar curve, where we follow [24, 26].

Let us have the closed $\Gamma = \{\mathbf{x}(u);\ u \in [0,1],\ \mathbf{x}(0) = \mathbf{x}(1)\}$, where $\mathbf{x}(u) = (x(u), y(u)) \in \mathbb{R}^2$ is the position vector of the curve Γ for the parameter u. In the discretized form, we can imagine it as shown in Fig. 1.

We suppose that the curve will expand due to an external force. In other words, we suppose the motion of the curve in the outer normal direction. If the curve Γ is parametrized in a counterclockwise direction, the unique definition of the unit tangent $\mathbf{T}$ and the outer unit normal $\mathbf{N}$ vectors to the plane curve Γ can be done as follows: $\mathbf{T} = \mathbf{x}_s$, $\mathbf{N} = \mathbf{x}_s^{\perp}$, and $\mathbf{T} \wedge \mathbf{N} = -1$, where $\mathbf{x}_s = \frac{\partial \mathbf{x}}{\partial s}$ and $\mathbf{T} \wedge \mathbf{N}$ denotes the determinant of the matrix with columns $\mathbf{T}$ and $\mathbf{N}$, where s is the arc-length parametrization of the curve Γ: $ds = g du$, where $g = |\mathbf{x}_u| = \left|\frac{\partial \mathbf{x}}{\partial u}\right| > 0$. If $\mathbf{T} = (x_s, y_s)$, then $\mathbf{N} = (y_s, -x_s)$.

Such curve could represent the fire perimeter on a horizontal plane [3]. Unfortunately, in most cases, the wildland fires do not occur on a flat terrain. Therefore, we define the surface $\mathcal{M}$ that represents a local Earth topography, given, e.g., by a digital terrain model. Let $\mathcal{M}$ be a two-dimensional surface in $\mathbb{R}^3$, $\mathcal{M} = \left\{(x, y, \varphi(x, y)) \in \mathbb{R}^3,\ (x, y) \in \Omega\right\}$, represented by a graph of the function $\varphi : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ defined on a domain $\Omega \subset \mathbb{R}^2$. Let the curve $\mathcal{G}$, parametrized by $u \in [0, 1]$, $(x(0), y(0), z(0)) = (x(1), y(1), z(1))$, be the smooth closed surface curve on $\mathcal{M}$ that represents the fire perimeter on the surface $\mathcal{M}$. Let us denote by p the arc-length parametrization of the curve $\mathcal{G}$: $dp = G du$, where $G = |\mathcal{G}_u| > 0$.

Furthermore, we suppose that the relation between the planar curve Γ and the surface curve $\mathcal{G}$ is $\mathcal{G} = \left\{(x(u), y(u), z(u) = \varphi(x(u), y(u))) \in \mathbb{R}^3,\ (x(u), y(u)) \in \Gamma\right\}$, what means the curve Γ is the vertical projection of the surface curve $\mathcal{G}$ (see Fig. 2).

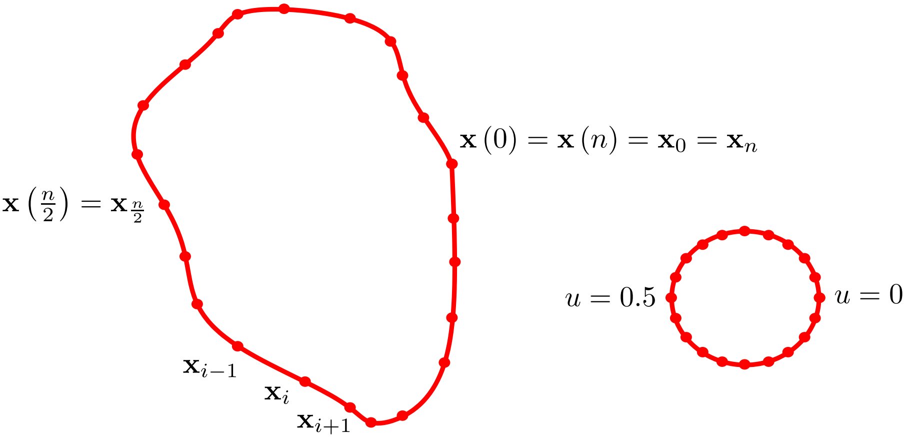

$\mathbf{x}(0) = \mathbf{x}(n) = \mathbf{x}_0 = \mathbf{x}_n$
$\mathbf{x}\left(\frac{n}{2}\right) = \mathbf{x}_{\frac{n}{2}}$
$\mathbf{x}_{i-1}$
$\mathbf{x}_i$
$\mathbf{x}_{i+1}$
$u = 0.5$
$u = 0$

Fig. 1 Discretization of a planar curve Γ (left), where n is the number of curve grid points. The closed curve is parametrized by $u \in [0, 1]$ and $\mathbf{x}(0) = \mathbf{x}(1)$, e. g., the circle with unit length (right)

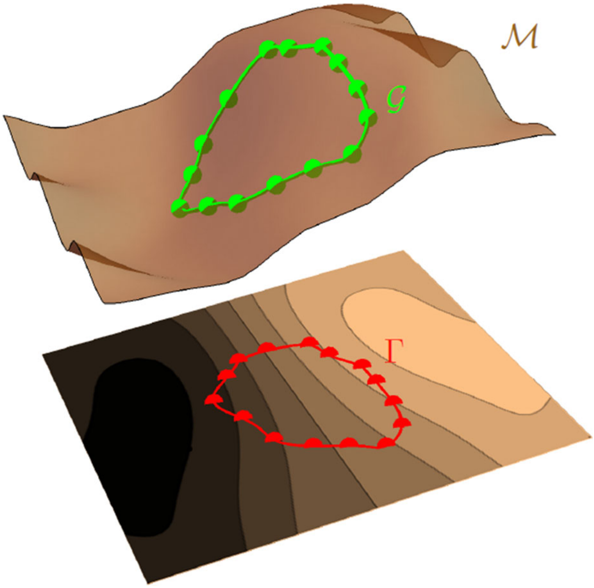


Fig. 2 The surface curve $\mathcal{G}$ on the surface $\mathcal{M}$ and the planar curve Γ as the vertical projection of $\mathcal{G}$ on the horizontal plane

Now, we can define the formula for the unit tangent $\mathcal{T}$ and the unit normal vectors $\mathcal{N}$ in the tangent plane to the surface $\mathcal{M}$. For the unit tangent vector $\mathcal{T}$, we obtain

$$\mathcal{T} = \frac{(\mathbf{T}, \nabla\varphi \cdot \mathbf{T})}{\sqrt{1 + (\nabla\varphi \cdot \mathbf{T})^2}}. \tag{1}$$

Note that here and in further notations, we denote a 3-D vector by $(\mathbf{a}, b)$, where $\mathbf{a} \in \mathbb{R}^2$ and $b \in \mathbb{R}$. We find $\mathcal{N}$ as the cross product of the unit tangent vector $\mathcal{T}$ and the upward-pointing unit normal vector $\mathcal{N}_{\mathcal{M}}$ to the surface $\mathcal{M}$. Vector $\mathcal{N}_{\mathcal{M}}$ is given as the cross product of two vectors from the tangent plane to the surface $\mathcal{M}$, e.g., $\mathbf{v_1} = (1,\ 0,\ \varphi_x)$ and $\mathbf{v_2} = \left(0,\ 1,\ \varphi_y\right)$, which is then normalized and we get

$$\mathcal{N}_{\mathcal{M}} = \frac{\left(-\varphi_x,\ -\varphi_y,\ 1\right)}{\sqrt{1 + |\nabla\varphi|^2}} = \frac{(-\nabla\varphi,\ 1)}{\sqrt{1 + |\nabla\varphi|^2}}. \tag{2}$$

Then, the outer unit normal vector $\mathcal{N}$ is derived as follows

$$\mathcal{N} = \frac{\left(\mathbf{N} + (\nabla\varphi)^{\perp}(\nabla\varphi \cdot \mathbf{T}),\ \nabla\varphi \cdot \mathbf{N}\right)}{\sqrt{\left(1 + |\nabla\varphi|^2\right)\left(1 + (\nabla\varphi \cdot \mathbf{T})^2\right)}}, \tag{3}$$

where $(\nabla\varphi)^{\perp} = \left(\varphi_y, -\varphi_x\right)$. Since $(\nabla\varphi)^{\perp} = (\nabla\varphi \cdot \mathbf{T})\,\mathbf{N} - (\nabla\varphi \cdot \mathbf{N})\,\mathbf{T}$, we finally get

$$\mathcal{N} = \frac{\left(\left(1 + (\nabla\varphi \cdot \mathbf{T})^2\right)\mathbf{N} - (\nabla\varphi \cdot \mathbf{T})(\nabla\varphi \cdot \mathbf{N})\,\mathbf{T},\ \nabla\varphi \cdot \mathbf{N}\right)}{\sqrt{\left(1 + |\nabla\varphi|^2\right)\left(1 + (\nabla\varphi \cdot \mathbf{T})^2\right)}}. \tag{4}$$

For the surface curve $\mathcal{G}$, we find its curvature vector $\mathcal{K}$ as the second derivative of $\mathcal{G}$ with respect to the arc-length parametrization p,

$$\mathcal{K} = \mathcal{G}_{pp} = \mathcal{T}_p = \frac{g}{G}\mathcal{T}_s = \left(\frac{\mathbf{x}_{ss}\left(1+\varphi_s^2\right) - \mathbf{x}_s\varphi_s\varphi_{ss}, \varphi_{ss}\left(1+\varphi_s^2\right) - \varphi_s^2\varphi_{ss}}{\left(1+\varphi_s^2\right)^2}\right). \tag{5}$$

Since $\varphi_{ss} = \left(\varphi_{xx}x_s x_s + \varphi_{xy}x_s y_s + \varphi_x x_{ss} + \varphi_{yx}y_s x_s + \varphi_{yy}y_s y_s + \varphi_y y_{ss}\right)$ and $-k\mathbf{N} = \mathbf{x}_{ss}$, where k is the curvature of the planar curve, we obtain $\varphi_{ss} = \mathbf{T}^T H(\varphi)\mathbf{T} - k(\nabla\varphi\cdot\mathbf{N})$, where $H(\varphi)$ is the Hessian (a square matrix with the second-order partial derivatives) of the terrain function φ and we obtain the final equation for the curvature vector $\mathcal{K}$,

$$\mathcal{K} = \left(\frac{-k\mathbf{N}\left(1+(\nabla\varphi\cdot\mathbf{T})^2\right) - \mathbf{T}(\nabla\varphi\cdot\mathbf{T})\mathbf{T}^T H(\varphi)\mathbf{T} - k(\nabla\varphi\cdot\mathbf{N})}{\left(1+(\nabla\varphi\cdot\mathbf{T})^2\right)^2}, \frac{\mathbf{T}^T H(\varphi)\mathbf{T} - k(\nabla\varphi\cdot\mathbf{N})}{\left(1+(\nabla\varphi\cdot\mathbf{T})^2\right)^2}\right). \tag{6}$$

From the Darboux frame, which is an analogue of the Frenet-Serret frame, we know that $\mathcal{K} = K_g(-\mathcal{N}) + K_n\mathcal{N}_{\mathcal{M}}$, where K_g is the geodesic curvature and K_n is the normal curvature. The splitting of the curvature $\mathcal{K}$ to K_g and K_n (see Fig. 3), is important for simulation of the surface fire spread. If we know the curvature in the tangent plane, K_g, we know how the fire perimeter shape influences the local normal velocity. The second part, K_n, expresses the rugged surface contribution to the local normal velocity, especially in canyons, in valleys, and on ridges. Since the geodesic curvature is the projection of $\mathcal{K}$ to the inner unit normal vector $-\mathcal{N}$, we get

$$K_g = k\frac{\sqrt{1+|\nabla\varphi|^2}}{\left(1+(\nabla\varphi\cdot\mathbf{T})^2\right)^{\frac{3}{2}}} - \frac{(\nabla\varphi\cdot\mathbf{N})\mathbf{T}^T H(\varphi)\mathbf{T}}{\sqrt{1+|\nabla\varphi|^2}\left(1+(\nabla\varphi\cdot\mathbf{T})^2\right)^{\frac{3}{2}}}. \tag{7}$$

The normal curvature is the component of $\mathcal{K}$ in the direction of the unit upward-pointing normal vector to the surface $\mathcal{N}_{\mathcal{M}}$,

$$K_n = \frac{\mathbf{T}^T H(\varphi)\mathbf{T}}{\left(1+(\nabla\varphi\cdot\mathbf{T})^2\right)\sqrt{1+|\nabla\varphi|^2}}. \tag{8}$$

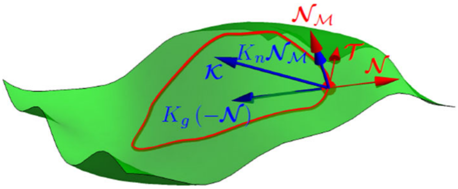


Fig. 3 Splitting of the curvature $\mathcal{K}$ to the geodesic (K_g) and normal (K_n) curvatures is useful in modeling of the surface fire perimeter evolution. K_g represents the influence of a local fire perimeter shape and K_n represents the influence of a topography shape

3 The wildland surface fire spread mathematical model

In this section, we describe the forces influencing the surface curve evolution. We characterize the outer normal velocity $\mathcal{V}$ of the surface curve $\mathcal{G}$, the external force $\mathcal{F}$, and its influencing factors, such as fuel, wind, and terrain slope. We also derive the general form of the planar (projected) curve evolution, where we split the planar velocity into the normal velocity β and the tangential velocity α. While the normal velocity changes the curve shape, the tangential velocity does not in the continuous form. In the discrete form, we use the tangential velocity for the redistribution of curve grid points that stabilizes the numerical model.

3.1 Normal velocity of the surface curve

In our approach, we evolve the surface curve $\mathcal{G}$ representing the fire perimeter and we need to define the driving forces. The overall fire spread rate is expressed by the external force $\mathcal{F}$. Besides that, the fire spread is influenced by the shape of the fire perimeter too. In other words, the curvatures of the surface curve influence the local normal velocity (see Fig. 4). The geodesic curvature in the tangent plane to the surface, K_g, smoothes the curve. Moreover, the shape of the topography influences the normal velocity. The normal curvature K_n of the curve evolving in a valley (or on a ridge) can increase (or decrease) the normal velocity $\mathcal{V}$. Such an evolution of the curve $\mathcal{G}$ can be described by the following velocity $\mathcal{V}$ in the outer normal direction,

$$\mathcal{V} = \mathcal{F}\left(\delta_{\mathcal{F}} - \delta_g K_g + \delta_n K_n\right), \tag{9}$$

where $\mathcal{F}$ is the external force, $\delta_{\mathcal{F}}$ is the weight of the external force, δ_g is the weight of the geodesic curvature, and δ_n is the weight of the normal curvature influence on the fire spread. Such formula expresses the dominant role of the external force that can be accelerated or slowed down by the geodesic and normal curvatures. In the case of the flat terrain ($K_n = 0$) and the linear fire perimeter ($K_g = 0$), the fire spread

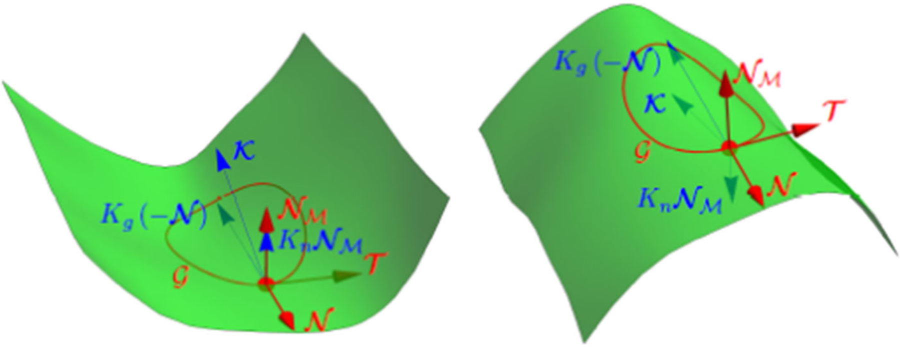


Fig. 4 Geodesic (K_g) and normal (K_n) curvatures in a valley and on a ridge. In both cases, K_g is the same, since the local fire perimeter is the same on a tangent plane and it decreases the local normal velocity. However, K_n differs. In a valley, $K_n > 0$; therefore, the normal velocity will increase. On the contrary, on a ridge, $K_n < 0$ and the normal velocity will decrease

depends only on the external force $\mathcal{F}$. Considering a valley or a ridge, $K_n \neq 0$, and it is used to express the topography influence on the normal velocity. In the valley (see Fig. 4 left) $K_n > 0$ and causes the fire spread acceleration due to the accumulation of the radiation heat. On the other hand, on the ridge (see Fig. 4 right) $K_n < 0$ and it slows the fire spread down, since the radiation heat is dissipated. Considering a non-linear fire perimeter, i.e., the curve has convex and concave parts in the tangent plane, the geodesic curvature $K_g \neq 0$. In parts with $K_g < 0$, the fire perimeter is accelerated due to radiation heat accumulation and in parts with $K_g > 0$, the radiation heat dissipation slows down the fire perimeter propagation.

Our design of an external force influencing the fire behavior is based on the empirical laws of the wildland fire perimeter propagation. Research indicates that the wildland fire propagation is influenced by the fuel parameters, weather conditions, and surrounding topography slope. Considering such most important factors, we suggest the following formula for the external force,

$$\mathcal{F} = f \, f_w(\mathbf{w} \cdot \boldsymbol{\mathcal{N}}) \, f_s(\mathbf{s} \cdot \boldsymbol{\mathcal{N}}), \tag{10}$$

where f is the fuel influence, $f_w(\mathbf{w} \cdot \boldsymbol{\mathcal{N}})$ is the wind influence, and $f_s(\mathbf{s} \cdot \boldsymbol{\mathcal{N}})$ is the terrain slope influence on the rate of spread, with $\mathbf{w}$ being the three-dimensional wind vector, $\mathbf{s}$ being the three-dimensional slope vector, and $\boldsymbol{\mathcal{N}}$ being the unit normal vector in the tangent plane to the surface curve.

The fuel influence In the literature [29], ground, surface, and aerial fuels are distinguished in the modeling of the wildland fires.

In our model, we shall consider a surface fuel. It is usually not homogeneousn and therefore we assume a heterogeneous fuel flammability on a topographic surface. The rate of spread (ROS) of the fire perimeter in the normal direction usually depends on the local fuel properties. Therefore, the spatial variability in ROS is expressed by a 2-D ROS map, $f(\mathbf{x})$, $\mathbf{x} \in \Omega \subset \mathbb{R}^2$, and $f(\mathbf{x})$ gives the rate of spread in a point $(\mathbf{x}_i, \varphi(\mathbf{x}_i))$ on the surface. According to the literature [16], we suppose that the ROS map is given by a weighted combination of the most important factors, such as species, age, bulk density, fuel moisture, vertical arrangement, fuel loading, and compactness. Some of these factors can be determined by typological forestry maps, like the species, age, or bulk density, and their combination creates the ROS map (see Fig. 5), where the young, dense coniferous forest is considered as a fuel with the highest spread rate. Moreover, it is possible to adapt the ROS map $f(\mathbf{x})$ to the existing fire propagation models, e.g., Rothermel's fire spread model.

The wind influence Wind can dramatically increase (or decrease) the fire spread rate if the fire spreads in (or against) the wind direction. It may also change the fire spread direction. The wind increases the fuel preheating and drying, and it supplies the oxygen to the fire.

Measuring the wind speed and direction on a topography with sufficient spatial resolution during the fire is not possible. Therefore, we consider that wind has the same speed and direction on the whole topography given by a two-dimensional vector $\mathbf{w}^{2D}$. Such vector can be provided, e.g., by a weather forecast service. However, we evolve curve on a variable surface; hence, we need a 3-D wind vector $\mathbf{w}$. While we

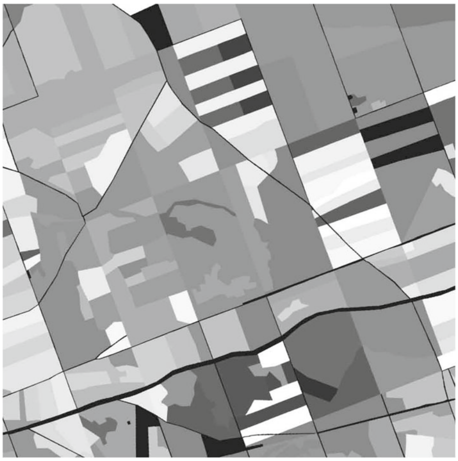


Fig. 5 The example of the scalar rate of spread (ROS) map f, created as a weighted combination of selected factors, such as species, age, and bulk density

retain the length, $|\mathbf{w}| = |\mathbf{w}_{2D}|$, we construct the third coordinate as the directional derivative of the terrain function φ along a vector $\mathbf{w}^{2D}$, i.e., the vector $\mathbf{w}$ lies in the tangent plane, mathematically expressed as

$$\mathbf{w} = \left(\mathbf{w}^{2D}, \nabla\varphi \cdot \mathbf{w}^{2D}\right) \frac{\left|\mathbf{w}^{2D}\right|}{\sqrt{\left|\mathbf{w}^{\mathbf{2D}}\right|^2 + \left(\nabla\varphi \cdot \mathbf{w}^{2D}\right)^2}} \tag{11}$$

for $\mathbf{w}_{2D} \neq \mathbf{0}$. In the case of no wind, $\mathbf{w}_{2D} = \mathbf{0}$, we set $\mathbf{w} = \mathbf{0}$.

According to the results in [34, 38], the wind influences the rate of spread exponentially, so we consider the scalar product of the wind vector $\mathbf{w}$ and the outer normal vector $\boldsymbol{\mathcal{N}}$ as an exponent of the function f_w in the form

$$f_w\left(\mathbf{w} \cdot \boldsymbol{\mathcal{N}}\right) = e^{\lambda_w (\mathbf{w} \cdot \boldsymbol{\mathcal{N}})}, \tag{12}$$

where λ_w is a positive parameter. If those vectors are perpendicular, $\mathbf{w} \cdot \boldsymbol{\mathcal{N}} = 0$, the external force $\mathcal{F}$ is not influenced by the wind, because $f_w = 1$. If the vectors are parallel, with the same orientation, $\mathbf{w} \cdot \boldsymbol{\mathcal{N}} = |\mathbf{w}|$, $f_w = e^{\lambda_w |\mathbf{w}|}$, the influence of the wind is the strongest. In all other cases, the exponent of the function f_w is given by the projection of the wind vector $\mathbf{w}$ onto the outer normal vector $\boldsymbol{\mathcal{N}}$ (see Fig. 6).

The topography slope influence The slope, similarly to wind, can increase (or decrease) the fire spread or change the spread direction. The slope increases the radiation and convection heat transfer up the slope. From the digital terrain model (the topography function φ), we can easily obtain a vector function $\nabla\varphi$ characterizing the topography slope. Now, we adjust the 2-D vector function $\nabla\varphi$ to the tangent plane to get a slope vector $\mathbf{s}$ the same way as we did for the wind vector $\mathbf{w}$. We obtain the third coordinate of the vector $\mathbf{s}$ as the directional derivative of the terrain function φ along $\nabla\varphi$, i.e.,

$$\mathbf{s} = \left(\nabla\varphi, |\nabla\varphi|^2\right) \frac{|\nabla\varphi|}{\sqrt{|\nabla\varphi|^2 + \left(|\nabla\varphi|^2\right)^2}} = \frac{\left(\nabla\varphi, |\nabla\varphi|^2\right)}{\sqrt{1 + |\nabla\varphi|^2}}. \tag{13}$$

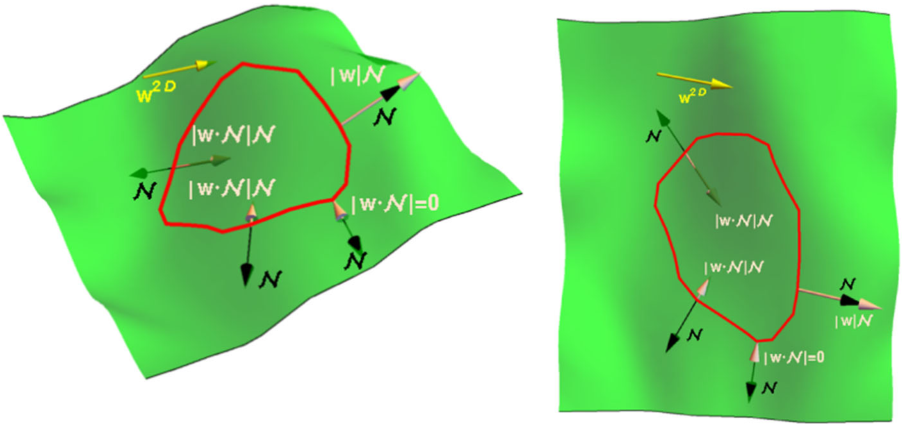

$\mathbf{w}^{2D}$
$|\mathbf{w}|\mathcal{N}$
$\mathcal{N}$
$|\mathbf{w}\cdot\mathcal{N}|\mathcal{N}$
$\mathcal{N}$
$|\mathbf{w}\cdot\mathcal{N}|\mathcal{N}$
$|\mathbf{w}\cdot\mathcal{N}|=0$
$\mathcal{N}$
$\mathcal{N}$
$\mathbf{w}^{2D}$
$\mathcal{N}$
$|\mathbf{w}\cdot\mathcal{N}|\mathcal{N}$
$|\mathbf{w}\cdot\mathcal{N}|\mathcal{N}$
$\mathcal{N}$
$\mathcal{N}$
$|\mathbf{w}|\mathcal{N}$
$|\mathbf{w}\cdot\mathcal{N}|=0$
$\mathcal{N}$

Fig. 6 An example of the projection of the wind vector $\mathbf{w}$ (white) to the normal vector $\mathcal{N}$ (black). The dot product $(\mathbf{w} \cdot \mathcal{N})$ can lead to various results. If the wind blows in the outer normal direction $|\mathbf{w} \cdot \mathcal{N}| > 0$. If the wind blows in the opposite direction $|\mathbf{w} \cdot \mathcal{N}| < 0$ and if the vectors $\mathcal{N}$ and $\mathbf{w}$ are perpendicular $|\mathbf{w} \cdot \mathcal{N}| < 0$. The wind vector $\mathbf{w}$ is computed from a measured two-dimensional wind vector $\mathbf{w}^{2D}$ (yellow)

for $\nabla\varphi \neq \mathbf{0}$. In the case of flat terrain, $\nabla\varphi = \mathbf{0}$, we set $\mathbf{s} = \mathbf{0}$.

According to the results in [10, 38], the slope influences the rate of spread exponentially, depending on the projection of $\mathbf{s}$ to $\mathcal{N}$; therefore, we consider

$$f_s\left(\mathbf{s} \cdot \mathcal{N}\right) = e^{\lambda_s (\mathbf{s} \cdot \mathcal{N})}, \tag{14}$$

where λ_s is a positive parameter.

3.2 Evolution of the projected planar curve

We suppose that the projected planar curve Γ of the surface curve $\mathcal{G}$, which models the fire spread, moves in time by a general planar velocity vector field v. We can split such general motion of any point $\mathbf{x}$ of the curve Γ into the normal and tangential directions, so we consider a general form of the planar curve evolution in the following form

$$\mathbf{x}_t = v = \beta\mathbf{N} + \alpha\mathbf{T}, \tag{15}$$

where β is the velocity in the normal direction $\mathbf{N}$ and α is the tangential velocity of the planar curve Γ. In Section 3, we designed the normal velocity $\mathcal{V}$ for the surface curve $\mathcal{G}$. Now, we want to relate the normal velocity $\mathcal{V}$ in the tangent plane to the projected curve Γ normal velocity β. Following [24], we subsequently get that

$$\mathcal{V} = \mathcal{G}_t \cdot \mathcal{N} = \left(x_t, y_t, \varphi_t\left(x, y\right)\right) \cdot \mathcal{N} = \left(\mathbf{x}_t, \mathbf{x}_t \cdot \nabla\varphi\right) \cdot \mathcal{N} = \sqrt{\frac{1 + |\nabla\varphi|^2}{1 + (\nabla\varphi \cdot \mathbf{T})^2}}\beta, \tag{16}$$

from where we obtain

$$\beta = \mathcal{V}\sqrt{\frac{1 + (\nabla\varphi \cdot \mathbf{T})^2}{1 + |\nabla\varphi|^2}}. \tag{17}$$

Employing the equation for the normal velocity of the surface curve (9), the geodesic curvature (7), and the normal curvature (8), taking into account the Frenet equation $\mathbf{x}_{ss} = -k\mathbf{N}$, the equation for the normal velocity of the planar curve (17) can be written in the form $\beta = (-\varepsilon k + w)$. Now, we rewrite the general form of the planar curve evolution (15) in terms of the intrinsic PDE for the evolution of the position vector $\mathbf{x}$ of the planar curve Γ, where we split the normal velocity β into the curvature term ε and the force term w

$$\mathbf{x}_t = \varepsilon \mathbf{x}_{ss} + \alpha \mathbf{x}_s + w \mathbf{x}_s^{\perp}, \tag{18}$$

where

$$\varepsilon = \frac{\mathcal{F}\delta_g}{1 + (\nabla\varphi \cdot \mathbf{T})^2}, \tag{19}$$

$$\begin{aligned} w = \mathcal{F} \Bigg(& \delta_{\mathcal{F}} \sqrt{\frac{1+\left(\nabla\varphi\cdot\mathbf{T}\right)^2}{1+|\nabla\varphi|^2}} + \delta_g \frac{\mathbf{T}^T H(\varphi)\mathbf{T}(\nabla\varphi\cdot\mathbf{N})}{\left(1+(\nabla\varphi\cdot\mathbf{T})^2\right)\left(1+|\nabla\varphi|^2\right)} \\ & + \delta_n \frac{\mathbf{T}^T H(\varphi)\mathbf{T}}{\sqrt{1+(\nabla\varphi\cdot\mathbf{T})^2}\left(1+|\nabla\varphi|^2\right)} \Bigg) \end{aligned} \tag{20}$$

and a suitable choice of α is discussed in the following subsection.

3.3 The choice of the tangential velocity

Although it is well known that a tangential motion does not change the shape of the evolving curve, we know that it is helpful in stabilization of the numerical algorithms based on Lagrangian approaches [23, 25]. If we want to preserve a distribution of points, we must conserve the ratio $\frac{g}{L}$, where g represents the local length $g = |\mathbf{x}_u|$ and L the global curve length [2, 3, 23–28]. Considering the numerical discretization, we obtain $\frac{g}{L} \approx \frac{\frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{h}}{L} = \frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{Lh} = \frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{\frac{L}{n}}$, where $h = \frac{1}{n}$, n is the number of the curve grid points (see Fig. 1). The numerator denotes the distance between two neighbouring points, and the denominator denotes the average distance between neighbouring points (since n denotes the total number of curve points and its segments). One can get the curve with uniformly distributed discrete grid points, if the ratio $\frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{\frac{L}{n}} \to 1$ in time for all discrete segments, so in the continuous formulation, we should have $\frac{g}{L} \to 1$ with increasing time [25]. For the time evolution of the curve local length g, we obtain

$$\begin{aligned} g_t &= |\mathbf{x}_u|_t = \frac{\mathbf{x}_u}{|\mathbf{x}_u|} \cdot (\mathbf{x}_u)_t = \frac{g\mathbf{x}_s}{g} \cdot (\mathbf{x}_t)_u = \mathbf{T} \cdot (\beta\mathbf{N} + \alpha\mathbf{T})_u = \mathbf{T} \cdot g\,(\beta\mathbf{N} + \alpha\mathbf{T})_s \\ &= \mathbf{T} \cdot g\,(\beta_s\mathbf{N} + \beta\mathbf{N}_s + \alpha_s\mathbf{T} + \alpha\mathbf{T}_s) = \mathbf{T} \cdot g\,((\beta_s - \alpha k)\,\mathbf{N} + (\beta k + \alpha_s)\,\mathbf{T}) \\ &= gk\beta + g\alpha_s = gk\beta + \alpha_u. \end{aligned} \tag{21}$$

By integrating the previous equation, we obtain the formula for the evolution of the global curve length L

$$L_t = \int_0^1 g_t du = \int_0^1 gk\beta du + \int_0^1 \alpha_u du = \int_0^1 gk\beta du + \alpha\,(1) - \alpha\,(0)\,. \qquad (22)$$

Due to the periodic boundary condition, we have $\alpha\,(1) = \alpha\,(0)$. Then, we rewrite the formula as follows

$$L_t = \int_0^1 gk\beta du = \int_\Gamma k\beta\, ds = L\,\langle k\beta\rangle_\Gamma\,, \qquad (23)$$

where $\langle k\beta\rangle_\Gamma = \frac{1}{L}\int_\Gamma k\beta\, ds$. For the time evolution of the ratio $\dfrac{g}{L}$, we obtain

$$\left(\frac{g}{L}\right)_t = \frac{(gk\beta + \alpha_u)\,L - gL\,\langle k\beta\rangle_\Gamma}{L^2} = \frac{g}{L}\left(k\beta + \alpha_s - \langle k\beta\rangle_\Gamma\right). \qquad (24)$$

If we choose $\left(\dfrac{g}{L}\right)_t = 0$, we get $\alpha_s = \langle k\beta\rangle_\Gamma - k\beta$ and with this tangential velocity, the curve preserves the initial distribution of grid points [23]. If we impose $\left(\dfrac{g}{L}\right)_t = \omega\left(1 - \dfrac{g}{L}\right)$, where ω is a parameter determining how fast the redistribution becomes uniform, we get $\frac{g}{L} \to 1$ and we obtain the formula for the tangential velocity which guarantees the asymptotically uniform redistribution of grid points of the planar curve Γ [25],

$$\alpha_s = \langle k\beta\rangle_\Gamma - k\beta + \omega\left(\frac{L}{g} - 1\right). \qquad (25)$$

4 The numerical scheme

In the previous section, we derived the intrinsic PDE for the planar curve Γ position vector $\mathbf{x}$ evolution,

$$\mathbf{x}_t - \alpha\mathbf{x}_s = \varepsilon\mathbf{x}_{ss} + w\mathbf{x}_s^\perp, \qquad (26)$$

where $\varepsilon,\ w,$ andα are given by (19), (20), and (25), and here we present its numerical discretization. First, we perform the spatial discretization, which is based on the flowing finite volume method in space [23], and then we discuss the semi-implicit discretization [3, 23] in time. To guarantee the solvability of arising cyclic tridiagonal linear systems for any choice of time step, we use the so-called inflow-implicit/outflow-explicit scheme [3, 21, 22] in the approximation of the advection term.

Integrating (26) over the finite volume $\mathbf{p}_i = [\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_{i+\frac{1}{2}}]$ (see Fig. 7), where $\mathbf{x}_{i-\frac{1}{2}}$ represents the middle point between the points $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$, i.e.,

$$\mathbf{x}_{i-\frac{1}{2}} = \frac{\mathbf{x}_{i-1} + \mathbf{x}_i}{2}, \qquad (27)$$

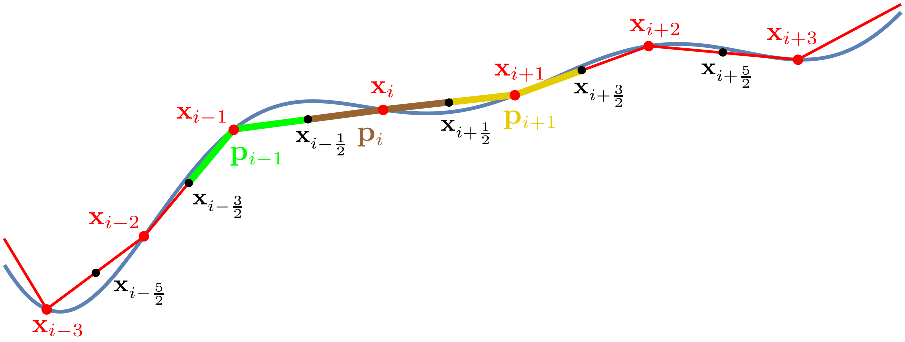


Fig. 7 Visualization of the curve discretization, curve grid points (red), and their midpoints. Finite volumes $\mathbf{p}_{i-1}$, $\mathbf{p}_i$, and $\mathbf{p}_{i+1}$ are highlighted by green, brown, and yellow color. Note that p_i is not a straight line given by $\mathbf{x}_{i-\frac{1}{2}}$ and $\mathbf{x}_{i+\frac{1}{2}}$, but a broken line given by $\mathbf{x}_{i-\frac{1}{2}}$, $\mathbf{x}_i$ and $\mathbf{x}_{i+\frac{1}{2}}$

we get

$$\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{x}_t ds - \alpha \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{x}_s ds = \epsilon \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{x}_{ss} ds + w \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{x}_s^{\perp} ds, \tag{28}$$

where the values ϵ, α, and w are considered constant, with values ϵ_i, α_i, andw_i on the discrete curve segment $\mathbf{p}_i$ around the point $\mathbf{x}_i$. We define $h_i = |\mathbf{x}_i - \mathbf{x}_{i-1}|$, then the measure of the segment $\mathbf{p}_i$ is equal to $\frac{h_i + h_{i+1}}{2}$, and using the Newton-Leibniz formula, we get the following approximation of (28), for $i = 1, ..., n$,

$$\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i \left[\mathbf{x}\right]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} = \epsilon_i \left[\mathbf{x}_s\right]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} + w_i \left(\left[\mathbf{x}\right]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \right)^{\perp} \tag{29}$$

$$\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) = \epsilon_i \left[\mathbf{x}_s\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + w_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right)^{\perp}. \tag{30}$$

Approximating the arc-length derivative $\mathbf{x}_s$ in the first bracket on the right-hand side by a finite difference, we obtain the semi-discrete flowing finite volume scheme,

$$\begin{aligned} &\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) \\ &= \epsilon_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right)^{\perp}. \end{aligned} \tag{31}$$

Now, we split the advection term (involving the tangential velocity α) as follows,

$$-\alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) = \alpha_i \left(\mathbf{x}_i - \mathbf{x}_{i+\frac{1}{2}} \right) - \alpha_i \left(\mathbf{x}_i - \mathbf{x}_{i-\frac{1}{2}} \right). \tag{32}$$

Using (27), we obtain

$$\begin{aligned} &\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t + \frac{\alpha_i}{2} (\mathbf{x}_i - \mathbf{x}_{i+1}) - \frac{\alpha_i}{2} (\mathbf{x}_i - \mathbf{x}_{i-1}) \\ &= \epsilon_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^{\perp}. \end{aligned} \tag{33}$$

If $\alpha_i < 0$, i.e., the velocity of advection $(-\alpha)$ in (26) is positive in the segment p_i, it represents the inflow in $\mathbf{x}_{i-\frac{1}{2}}$ and the outflow in $\mathbf{x}_{i+\frac{1}{2}}$. On the other hand, if $\alpha_i > 0$, it represents the inflow in $\mathbf{x}_{i+\frac{1}{2}}$ and the outflow in $\mathbf{x}_{i-\frac{1}{2}}$. Let us define

$$b^{in}_{i-\frac{1}{2}} = \max\left(-\alpha_i, 0\right), b^{\mathrm{out}}_{i-\frac{1}{2}} = \min\left(-\alpha_i, 0\right),$$

$$b^{\mathrm{in}}_{i+\frac{1}{2}} = \max\left(\alpha_i, 0\right), b^{\mathrm{out}}_{i+\frac{1}{2}} = \min\left(\alpha_i, 0\right) \tag{34}$$

and rewrite the equation (33) as follows

$$\begin{aligned}
&\frac{h_i + h_{i+1}}{2}(\mathbf{x}_i)_t + \frac{1}{2}\left(b^{\mathrm{in}}_{i+\frac{1}{2}} + b^{\mathrm{out}}_{i+\frac{1}{2}}\right)(\mathbf{x}_i - \mathbf{x}_{i+1}) + \frac{1}{2}\left(b^{\mathrm{in}}_{i-\frac{1}{2}} + b^{\mathrm{out}}_{i-\frac{1}{2}}\right)(\mathbf{x}_i - \mathbf{x}_{i-1}) \\
&= \epsilon_i\left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i}\right) + w_i\left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2}\right)^{\perp}.
\end{aligned} \tag{35}$$

In order to perform the time discretization, let us denote by m the time step index and by τ the length of the discrete time step. Let us approximate the time derivative by the finite difference $(\mathbf{x}_i)_t = \frac{\mathbf{x}_i^{m+1} - \mathbf{x}_i^m}{\tau}$, taking the inflow part of the advection term and the curvature term implicitly and the outflow part of the advection term and the force term explicitly. We obtain

$$\begin{aligned}
&\frac{h^m_{i+1} + h^m_i}{2}\frac{\mathbf{x}^{m+1}_i - \mathbf{x}^m_i}{\tau} + \frac{1}{2}b^{\mathrm{in}^m}_{\ \ i+\frac{1}{2}}\left(\mathbf{x}^{m+1}_i - \mathbf{x}^{m+1}_{i+1}\right) + \frac{1}{2}b^{\mathrm{in}^m}_{\ \ i-\frac{1}{2}}\left(\mathbf{x}^{m+1}_i - \mathbf{x}^{m+1}_{i-1}\right) \\
&- \varepsilon_i\left(\frac{\mathbf{x}^{m+1}_{i+1} - \mathbf{x}^{m+1}_i}{h^m_{i+1}} - \frac{\mathbf{x}^{m+1}_i - \mathbf{x}^{m+1}_{i-1}}{h^m_i}\right) \\
&= -\frac{1}{2}b^{\mathrm{out}^m}_{\ \ i+\frac{1}{2}}\left(\mathbf{x}^m_i - \mathbf{x}^m_{i+1}\right) - \frac{1}{2}b^{\mathrm{out}^m}_{\ \ i-\frac{1}{2}}\left(\mathbf{x}^m_i - \mathbf{x}^m_{i-1}\right) + w^m_i\left(\frac{\mathbf{x}^m_{i+1} - \mathbf{x}^m_{i-1}}{2}\right)^{\perp},
\end{aligned} \tag{36}$$

from where we get the fully discrete scheme in the form of a cyclic tridiagonal system

$$\begin{aligned}
&\mathbf{x}^{m+1}_{i-1}\left(-\frac{\varepsilon^m_i}{h^m_i} - \frac{b^{\mathrm{in}^m}_{\ \ i-\frac{1}{2}}}{2}\right) + \mathbf{x}^{m+1}_{i+1}\left(-\frac{\varepsilon^m_i}{h^m_{i+1}} - \frac{b^{\mathrm{in}^m}_{\ \ i+\frac{1}{2}}}{2}\right) \\
&+\mathbf{x}^{m+1}_i\left(\frac{h^m_{i+1} + h^m_i}{2\tau} + \frac{\varepsilon^m_i}{h^m_i} + \frac{\varepsilon^m_i}{h^m_{i+1}} + \frac{b^{\mathrm{in}^m}_{\ \ i-\frac{1}{2}}}{2} + \frac{b^{\mathrm{in}^m}_{\ \ i+\frac{1}{2}}}{2}\right) = \mathbf{x}^m_i\frac{h^m_{i+1} + h^m_i}{2\tau} \\
&-\frac{b^{\mathrm{out}^m}_{\ \ i+\frac{1}{2}}}{2}\left(\mathbf{x}^m_i - \mathbf{x}^m_{i+1}\right) - \frac{b^{\mathrm{out}^m}_{\ \ i-\frac{1}{2}}}{2}\left(\mathbf{x}^m_i - \mathbf{x}^m_{i-1}\right) + w^m_i\left(\frac{\mathbf{x}^m_{i+1} - \mathbf{x}^m_{i-1}}{2}\right)^{\perp},
\end{aligned} \tag{37}$$

$i = 1, ..., n$, where n is the number of the curve grid points.

In complex fire simulations, e.g., when curves are merging or splitting, the evolving fire perimeter can be locally sharp, even with singularities. Therefore, in such singular points, we use just the first-order implicit upwind scheme instead of the second-order inflow-implicit/outflow-explicit method in the discretization of the advection term. This first-order upwind scheme is applied when the angle between two consecutive curve segments $(\mathbf{x}_{i-1}, \mathbf{x}_i)$ and $(\mathbf{x}_i, \mathbf{x}_{i-1})$ is less than 120°, i.e., $\measuredangle(\mathbf{x}_{i-1}\mathbf{x}_i\mathbf{x}_{i+1}) < 120°$. In the upwind scheme, we use only the inflow velocities $b^{\text{in}}_{i-\frac{1}{2}}, b^{\text{in}}_{i+\frac{1}{2}}$, and points $\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_{i+\frac{1}{2}}$ in (32) are approximated by the neighbouring values $\mathbf{x}_{i-1}$ or $\mathbf{x}_{i+1}$, depending on the inflow direction. In this way, we get instead of (35) that

$$
\begin{aligned}
&\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t + b^{\text{in}}_{i+\frac{1}{2}} (\mathbf{x}_i - \mathbf{x}_{i+1}) + b^{\text{in}}_{i-\frac{1}{2}} (\mathbf{x}_i - \mathbf{x}_{i-1}) \\
&= \epsilon_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^{\perp}
\end{aligned}
\tag{38}
$$

and

$$
\begin{aligned}
&\mathbf{x}^{m+1}_{i-1} \left(-\frac{\varepsilon^m_i}{h^m_i} - {b^{\text{in}}}^m_{i-\frac{1}{2}} \right) + \mathbf{x}^{m+1}_{i+1} \left(-\frac{\varepsilon^m_i}{h^m_{i+1}} - {b^{\text{in}}}^m_{i+\frac{1}{2}} \right) \\
&+\mathbf{x}^{m+1}_i \left(\frac{h^m_{i+1} + h^m_i}{2\tau} + \frac{\varepsilon^m_i}{h^m_i} + \frac{\varepsilon^m_i}{h^m_{i+1}} + {b^{\text{in}}}^m_{i-\frac{1}{2}} + {b^{\text{in}}}^m_{i+\frac{1}{2}} \right) = \mathbf{x}^m_i \frac{h^m_{i+1} + h^m_i}{2\tau} \\
&+w^m_i \left(\frac{\mathbf{x}^m_{i+1} - \mathbf{x}^m_{i-1}}{2} \right)^{\perp}
\end{aligned}
\tag{39}
$$

instead of (37). This replacement of (37) by (39) occurs rarely, but in case it arises, the usage of (39) makes the scheme robust with respect to singularities.

The above systems (37) and (39) are both strictly diagonally dominant; thus, it is always solvable by the efficient cyclic tridiagonal solver (a modification of the Thomas algorithm) without any restriction on time step length τ [3].

In the numerical schemes (37) and (39), there are two parameters, ε^m_i and w^m_i, given by (19)–(20), which are evaluated as follows,

$$
\varepsilon^m_i = \frac{\mathcal{F}^m_i \delta_g}{1 + \left(\nabla\varphi^m_i . \mathbf{T}^m_i\right)^2},
\tag{40}
$$

$$
\begin{aligned}
w^m_i = \mathcal{F}^m_i \Bigg(&\delta_{\mathcal{F}} \sqrt{\frac{1+\left(\nabla\varphi^m_i . \mathbf{T}^m_i\right)^2}{1+|\nabla\varphi^m_i|^2}} + \delta_g \frac{{\mathbf{T}^m_i}^T H(\varphi^m_i) \mathbf{T}^m_i \ \nabla\varphi^m_i . \mathbf{N}^m_i}{\left(1+(\nabla\varphi^m_i . \mathbf{T}^m_i)^2\right)\left(1+|\nabla\varphi^m_i|^2\right)} \\
&+ \delta_n \frac{{\mathbf{T}^m_i}^T H(\varphi^m_i) \mathbf{T}^m_i}{\sqrt{1+(\nabla\varphi^m_i . \mathbf{T}^m_i)^2 \left(1+|\nabla\varphi^m_i|^2\right)}} \Bigg),
\end{aligned}
\tag{41}
$$

where $\mathbf{T}^m_i = \frac{\mathbf{x}^m_{i+1} - \mathbf{x}^m_{i-1}}{h^m_{i+1} + h^m_i}$, $\mathbf{N}^m_i = {\mathbf{T}^m_i}^{\perp}$, and $\mathcal{F}^m_i$, $\nabla\varphi^m_i$, and $H\left(\varphi^m_i\right)$ are discrete values of the external force, the topography slope, and the square matrix of the second-order partial derivatives of the topography function φ. The value $\nabla\varphi^m_i =$

$\nabla\varphi\left(\mathbf{x}_i^m\right)$ is obtained as the bilinear interpolation of the discrete gradient given by the central differences of the topography function φ given on the regular square grid. The discretization of $H\left(\varphi_i^m\right)$ is obtained similarly.

The external force $\mathcal{F}_i^m$ is discretized by

$$\mathcal{F}_i^m = \mathcal{F}\left(\mathbf{x}_i^m\right) = f_i^m \, e^{\lambda_s\left(\mathbf{s}_i^m \cdot \boldsymbol{\mathcal{N}}_i^m\right)} \, e^{\lambda_w\left(\mathbf{w}_i^m \cdot \boldsymbol{\mathcal{N}}_i^m\right)}, \tag{42}$$

where the values $f_i^m = f\left(\mathbf{x}_i^m\right)$ are obtained by bilinear interpolation of the rate of spread values given by a pixel map, and

$$\mathbf{w}_i^m = \mathbf{w}\left(\mathbf{x}_i^m\right) = \left(\mathbf{w}^{2D}, \nabla\varphi_i^m \cdot \mathbf{w}^{2D}\right) \frac{\left|\mathbf{w}^{2D}\right|}{\sqrt{\left|\mathbf{w^{2D}}\right|^2 + \left(\nabla\varphi_i^m \cdot \mathbf{w}^{2D}\right)^2}}, \tag{43}$$

$$\mathbf{s}_i^m = \mathbf{s}\left(\mathbf{x}_i^m\right) = \frac{\left(\nabla\varphi_i^m, \left|\nabla\varphi_i^m\right|^2\right)}{\sqrt{1 + \left|\nabla\varphi_i^m\right|^2}}. \tag{44}$$

In order to discretize the tangential velocity α_i, we first set $\alpha_0^m = 0$, which causes the point $\mathbf{x}_0$ will move only in the normal direction. Then, we get α_i^m for $i = 1, 2, ..., n-1$ by

$$\alpha_i^m = \alpha_{i-1}^m + h_i^m k_i^m \beta_i^m - h_i^m \left\langle k\beta \right\rangle_\Gamma^m + \omega\left(\frac{L^m}{n} - h_i^m\right), \tag{45}$$

where the curvature k_i^m, the normal velocity β_i^m, for $i = 1, 2, ..., n$, the mean value $\langle k\beta\rangle_\Gamma$, and the total curve length L^m are given by following formulas

$$k_i^m = \operatorname{sgn}\left(\mathbf{h}_{i-1}^m \wedge \mathbf{h}_{i+1}^m\right) \frac{1}{2h_i^m} \arccos\left(\frac{\mathbf{h}_{i+1}^m \cdot \mathbf{h}_{i-1}^m}{h_{i+1}^m h_{i-1}^m}\right), \tag{46}$$

$$\beta_i^m = -\frac{\varepsilon_{i-1}^m + \varepsilon_i^m}{2} k_i^m + \frac{w_{i-1}^m + w_i^m}{2}, \tag{47}$$

$$\langle k\beta\rangle_\Gamma = \frac{1}{L^m}\sum_{l=1}^{n} h_l^m k_l^m \beta_l^m, \tag{48}$$

$$L^m = \sum_{l=1}^{n} h_l^m, \tag{49}$$

where $\mathbf{h}_i^m = \mathbf{x}_i^m - \mathbf{x}_{i-1}^m$, $h_i^m = \left|\mathbf{h}_i^m\right|$.

Setting $\alpha_0^m = 0$ can cause an unnecessary large value of the tangential velocity in an effort to redistribute the curve points uniformly. To minimize the tangential velocity, we find the average tangential velocity $\alpha_{avg}^m = \sum\limits_{i=0}^{n} \frac{\alpha_i^m}{n}$. It is clear that α_{avg}^m

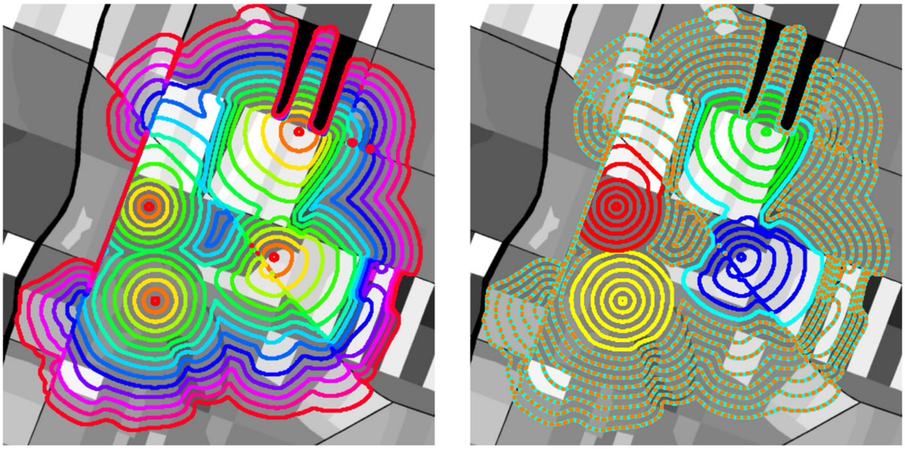


Fig. 8 An example of merging and splitting of curves on a ROS map. On the left, four red circles representing the initial fire perimeters and their time evolution are plotted. Curves with the same color represent the fire perimeter position at the same time. On the right, there is the same situation, but the evolution of different fire perimeters is plotted in different colors—red, yellow, green, and blue—until their merging. After the merging, the color changes to a combination, e.g., green and blue curves merge to a cyan curve

is an unnecessary tangential velocity, and therefore we find new minimized values as $\overline{\alpha_i^m} = \alpha_i^m - \alpha_{avg}^m$ for $i = 0, 1, ..., n-1$, by which we redefine α_i^m.

5 Topological changes treatment

By a topological change, we mean merging of several fire perimeters and/or splitting of the evolving curve into several separate curves (Fig. 8). Such splitting can occur when the curve velocity is locally slowed down significantly (e.g., nonburnable regions). Detecting and solving the topological changes in the Lagrangian approach is usually highly time-consuming [12], because the standard approaches have computational complexity $O\left(n^2\right)$, where n is the number of curve points. Such a high complexity is due to the strategy for the topological changes detection, which consists of computing pairwise distances between all grid points of the curve [12, 32]. The number of operations in such approach is $(n-1)^2 = n^2 - 2n + 1$ steps that must be performed in every computational time step and it slows down the overall computing time significantly. In this section, we present our $O(n)$ approach for the detection and processing of the topological changes in curve evolution which makes our overall computational method fast and applicable in a complex situation of the fire perimeter evolution.

In the sequel, we omit the time step index m, since we detect the topological changes before every time step and we will use following notations:

- NC is the number of curves
- n_j is the number of grid points of the jth curve Γ_j
- $n_{\min}$ is the minimal authorized number of grid points of a curve

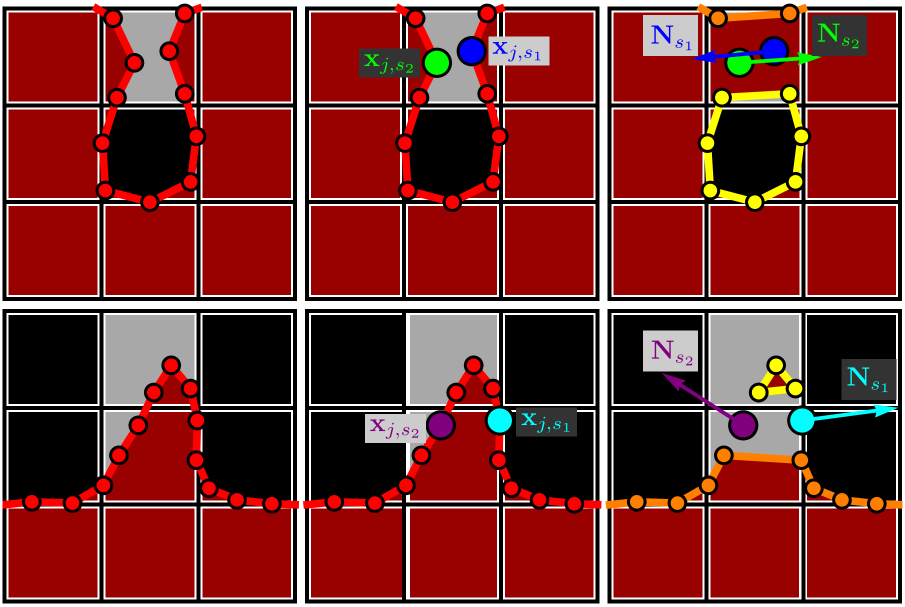


Fig. 9 Condition for splitting curves from Algorithm 3 line 18. The curve (red) is evolving on a heterogeneous ROS map. The first situation (upper row), when the curve encircles the nonburnable area (black), indicates splitting into inner and outer curves, since the normal N_{s_1} (blue) is pointing towards N_{s_2} (green). In this case, we split the curve (see also Fig. 8). The second situation (bottom row) illustrates the curve entering a narrow strip between nonburnable fuel. In this case, the normal N_{s_1} (cyan) is pointing away from N_{s_2} (purple); splitting and deleting the smaller curve would not allow the curve to spread through a narrow area. In this case, we do not split the curve

- $\overline{h}_j$ is the mean segment length of the curve Γ_j, i.e. $\overline{h}_j = L_j/n_j$, L_j is the length of curve Γ_j
- h_d is the desired segment length during the evolution
- $L_{\text{dif}}(i,k)$ is the local difference between the sum of the desired segment lengths h_d and the sum of the real segment lengths from the kth to the ith segment, i.e., $L_{\text{dif}}(i,k) = \sum_{j=k}^{j=i}\left(h_j - h_d\right)$
- $\mathbf{x}_{j,i}$ is the ith grid point of the jth curve Γ_j
- Deletecurve(j) is the function, that deletes the curve Γ_j. If $NC > j$, Γ_j is replaced by Γ_{NC} and NC is reduced by 1, otherwise only NC is reduced by 1.

Maintaining the desired segment length h_d is required for reliable topological changes detection. In our method, every curve is asymptotically uniformly discretized, i.e., all segment lengths of curves are close to their mean value $\overline{h}_j$. However, $\overline{h}_j$ can differ between different curves, and it may increase for expanding curves or decrease for shrinking curves. Such difference in $\overline{h}_j$ is non-desirable, especially in merging of curves, so we maintain the desired segment length h_d for all curves by adding or removing points by the Algorithm 2 after which the asymptotically uniform redistribution is quickly obtained by the tangential velocity included in the numerical method. The algorithm for adding and removing points is built on the idea of traversing the curve and summing the segment lengths, i.e., summing the distances between grid points. We check whether the difference between this sum and the corresponding sum of the desired segment length h_d is in absolute value larger than the desired length h_d of one segment. If this difference is larger than 0, we add a point to the longest segment. If this difference is less than 0, we remove the point between the shortest neighboring segments.

Our overall computing strategy can be described by the Algorithm 1.

Our main idea for the topological changes detection (see also [3]) is to create a narrow strip of cells along the curves. Let us have an array $p[I][J]$, where $I = 0, ..., py, \quad J = 0, ..., px$, covering the whole computational domain Ω. It is crucial to set the coupling between the cell size and the desired curve segment length h_d, especially when treating the curve splitting. To that goal, we set the cell size to $2h_d \times 2h_d$. Ideally, a smooth enough curve, as desired, should not have more than three points (two whole segments) in one cell, because the cell diagonal length is less than $3h_d$.

Algorithm 1 Overall computing strategy

```
foreach Time step do
    Topological changes detection (splitting) - Algorithm 3
    Topological changes detection (merging) - Algorithm 4
    foreach Curve $\Gamma_j$ do
        if $\left|h_d - \overline{h}_j\right| > \epsilon_h$ then
            Adding and removing grid points - Algorithm 2
        Numerical computation by the scheme (37), (39)
```

In our approach, we will split the curve if two non-neighboring curve grid points belong to the same cell (see Fig. 9 top). We also require that the difference of indices of such two non-neighboring grid points should be greater than or equal to three. This means that the curve should leave the cell and come back to the cell following the curve trajectory between those two points, because four consecutive points cannot belong to one cell. In Algorithm 3, the splitting is detected by traversing all curve grid points and marking the cells with the grid point numbers. If a point i belongs to a cell already marked by the j-th point and $i - j > 2$, we look for the nearest pair of points in a local neighborhood of the i-th and the j-th points. If the distance between the found nearest points is less than the mean segment length $\overline{h}_j$, we split the curve (see Fig. 9 again). However, in the situation of the curve entering the narrow strip between the nonburnable fuel (see Fig. 9 bottom), the curve would undesirably split. We prevent the splitting in such situations by checking the orientation of the outer normal vectors in the curve points, where the splitting should proceed. If the normal vectors are pointing away from each other, we do not split the curve. We deal with the merging detection similarly to the splitting detection. In Algorithm 4, we traverse through the points of all curves. In traversing, we mark the cell, to which

Algorithm 2 Adding and removing grid points

```
/* k is the segment number from which we measure
   $L_{dif}(i,k)$ */
$k \leftarrow 0$
$i \leftarrow 0$
$L_{dif}(i,k) \leftarrow 0$
for $i \leftarrow 1$ to $n_j$ do
    $L_{dif}(i,k) \leftarrow L_{dif}(i,k) + h_i - h_d$
    if $|L_{dif}(i,k)| > h_d$ then
        if $L_{dif}(i,k) > 0$ then
            /* find longest segment $h_l$, $k < l < i$ */
            $l \leftarrow$LongestSegment($k$, $i$)
            $N \leftarrow \text{floor}\left(L_{dif}(i,k)/h_d\right)$
            /* add $N$ point(s) on the longest segment */
            AddPoints($N$, $\mathbf{x}_l$)
            $L_{dif}(i,k) \leftarrow L_{dif}(i,k) - Nh_d$
            $k \leftarrow i$
        else
            /* find shortest neighboring segments $h_s$ and
               $h_{s+1}, k < s < i$ */
            $s \leftarrow$ ShortestNeighoringSegments($k$, $i$)
            /* remove the point between the shortest
               neighboring segments */
            RemovePoint($\mathbf{x}_s$)
            $L_{dif}(i,k) \leftarrow L_{dif}(i,k) + h_d$
            $k \leftarrow i$
    if $n_j < n_{min}$ then
        DeleteCurve($j$)
```

the curve point belongs, by the curve number. If the current point belongs to the cell already marked by a point of another curve, we detect merging. Finally, we look for the nearest pair of points in a local neighborhood of those two detected points and we perform the merging of the curves in this nearest pair.

In further notations, let us denote by $p[\mathbf{x}_{j,i}]$ the cell $p[\text{floor}\left(y_{j,i}/2h_d\right)]$ $[\text{floor}\left(x_{j,i}\right)/2h_d]$, where the point $\mathbf{x}_{j,i}$ belongs. To determine the cells $p[\mathbf{x}_{j,i}]$, we traverse all curve grid points instead of checking every cell. The topological changes are processed with the functions SplitCurve(j, s_1, s_2) in Algorithm 3 and MergeCurves(c_1, c_2, m_1, m_2) in Algorithm 4. SplitCurve(j, s_1, s_2) takes the curve number j and two points indexes s_1, s_2 as argument and splits the curve Γ_j at points $\mathbf{x}_{j,s_1}$ and $\mathbf{x}_{j,s_2}$ into two closed curves. The first point set $\mathbf{x}_{j,1}$, $\mathbf{x}_{j,2}$, ..., $\mathbf{x}_{j,s_1-1}$, $\mathbf{x}_{j,s_2+1}$, ..., $\mathbf{x}_{j,n}$ will replace the original curve by the number j. The second point set $\mathbf{x}_{j,s_1+1}$, $\mathbf{x}_{j,s_1+2}$, ..., $\mathbf{x}_{j,s_2-2}$, $\mathbf{x}_{j,s_2-1}$ will be stored as Γ_{NC+1} and the number of curves NC

Algorithm 3 Topological changes detection (splitting)

```
for $j \leftarrow 1$ to NC do
    for $i \leftarrow 1$ to $n_j$ do
        /* mark every cell $p[\mathbf{x}_{j,i}]$, where $i$th grid point
           of the curve $\Gamma_j$ lies by 0 */
        $p[\mathbf{x}_{j,i}] \leftarrow 0$
    for $i \leftarrow 1$ to $n_j$ do
        if $p[\mathbf{x}_{j,i}] = 0$ then
            $p[\mathbf{x}_{j,i}] \leftarrow i$
        else
            /* at least one grid point already belongs
               to the cell $p[\mathbf{x}_{j,i}]$ */
            if $i - p[\mathbf{x}_{j,i}] > 2$ then
                /* the grid points $\mathbf{x}_{j,i}$ and $\mathbf{x}_{j,p[\mathbf{x}_{j,i}]}$ lie in
                   the same cell and there are more than 2
                   curve grid points between them */
                $s_1 \leftarrow i$
                $s_2 \leftarrow p[\mathbf{x}_{j,i}]$
                /* find points $\mathbf{x}_{j,k}$ and $\mathbf{x}_{j,l}$ with shortest
                   distance */
                min $\leftarrow$ BIG
                for $k \in \{s_1 - 2; s_1 - 1; s_1; s_1 + 1; s_1 + 2\}$ do
                    for $l \in \{s_2 - 2; s_2 - 1; s_2; s_2 + 1; s_2 + 2\}$ do
                        if $\left|\mathbf{x}_{j,k} - \mathbf{x}_{j,l}\right| <$ min then
                            min $\leftarrow \left|\mathbf{x}_{j,k} - \mathbf{x}_{j,l}\right|$
                            $s_1 \leftarrow k$
                            $s_2 \leftarrow l$
                if $\left|\mathbf{x}_{j,s_1} - \mathbf{x}_{j,s_2}\right| < \overline{h}_j$ then
                    /* test, whether the normal vector $\mathbf{N}_{s_2}$
                       at point $\mathbf{x}_{j,s_2}$ is towards the normal
                       vector at point $\mathbf{x}_{j,s_1}$, $\epsilon$ is a small
                       number, see also Fig. 9 */
                    if $\left|\mathbf{x}_{j,s_1} - \mathbf{x}_{j,s_2}\right| > \left|\mathbf{x}_{j,s_1} - \left(\mathbf{x}_{j,s_2} + \epsilon\mathbf{N}_{s_2}\right)\right|$ then
                        SplitCurve($j$, $s_1$, $s_2$)
                        /* repeat the splitting detection
                           for $j$th curve $\Gamma_j$ */
                        $j--$
                        break
```

will be increased by 1. MergeCurves(c_1, c_2, m_1, m_2) takes two curve numbers c_1, c_2 and two points indices m_1, m_2 as an argument. The newly merged curve will consist of $n_{c_1} + n_{c_2} - 2$ points, namely $\mathbf{x}_{c_1,1}$, ..., $\mathbf{x}_{c_1,m_1-1}$, $\mathbf{x}_{c_2,m_2+1}$, $\mathbf{x}_{c_2,n_{c_2}}$, $\mathbf{x}_{c_2,1}$, ...,

Algorithm 4 Topological changes detection (merging)

```
1  if NC $>$ 1 then
2      for $j \leftarrow 2$ to NC do
3          for $i = 1$ to $n_j$ do
               /* mark every cell $p[\mathbf{x}_{j,i}]$, where $i$th grid
                  point of $\Gamma_j$ lies by 0 */
4              $p[\mathbf{x}_{j,i}] \leftarrow 0$
5      for $i \leftarrow 1 \rightarrow n_1$ do
           /* mark every cell $p[\mathbf{x}_{1,i}]$, where $i$th grid point
              of $\Gamma_1$ lies by 1 */
6          $p[\mathbf{x}_{1,i}] \leftarrow 1$
7      for $j \leftarrow 2$ to $NC$ do
8          for $i \leftarrow 1$ to $n_j$ do
9              if $p[\mathbf{x}_{j,i}] = 0$ then
10                 $p[\mathbf{x}_{j,i}] \leftarrow j$
11             else
12                 if $p[\mathbf{x}_{j,i}] < j$ then
                       /* point of curve $\Gamma_{p[\mathbf{x}_{j,i}]}$ already belongs
                          to the cell $p[\mathbf{x}_{j,i}]$ */
13                     $c_1 \leftarrow p[\mathbf{x}_{j,i}]$
14                     $c_2 \leftarrow j$
15                     $m_2 \leftarrow i$
16                     $p \leftarrow -1$
17                     for $k \leftarrow 1$ to $n_{c_1}$ do
18                         if $p[\mathbf{x}_{c_1,k}] = -1$ then
19                             $m_1 \leftarrow p[\mathbf{x}_{c_1,k}]$
20                     min $\leftarrow$ BIG
                       /* find the nearest points $\mathbf{x}_{c_1,k}$ and $\mathbf{x}_{c_2,l}$
                          */
21                     for $k \in \{m_1 - 2; m_1 - 1; m_1; m_1 + 1; m_1 + 2\}$ do
22                         for $l \in \{m_2 - 2; m_2 - 1; m_2; m_2 + 1; m_2 + 2\}$ do
23                             if $\left|\mathbf{x}_{c_1,k} - \mathbf{x}_{c_2,l}\right| <$ min then
24                                 min $\leftarrow \left|\mathbf{x}_{c_1,k} - \mathbf{x}_{c_2,l}\right|$
25                                 $m_1 \leftarrow k$
26                                 $m_2 \leftarrow l$
27                     if $\left|\mathbf{x}_{c_1,m_1} - \mathbf{x}_{c_2,m_2}\right| < \left(\overline{h}_{c_1} + \overline{h}_{c_2}\right)/2$ then
                           /* merge $\Gamma_{c_1}$ at point $\mathbf{x}_{c_1,m_1}$ with $\Gamma_{c_2}$ at
                              point $\mathbf{x}_{c_2,m_2}$ */
28                         MergeCurves($c_1, c_2, m_1, m_2$)
29                         Adding and removing grid points algorithm for curve
                           $\Gamma_{c_1}$
30                         goto line 1
```

$\mathbf{x}_{c_2,m_2-1}$, $\mathbf{x}_{c_1,m_1+1}$, $\mathbf{x}_{c_1,n_{c_1}}$, and will be stored as the curve Γ_{c_1}. The curve Γ_{c_2} will be deleted by DeleteCurve(c_2). We also note that due to the topological changes detection, we use the time step $\tau \leq 2h_d/\beta$ which prevents the grid points from skipping a cell in one time step.

6 Numerical experiments

In this section, we describe various numerical experiments showing the properties of our mathematical model and the numerical scheme. First, we show the experimental order of convergence of the numerical scheme on a circle evolving by an external force and the geodesic curvature. Then, we show how the curve evolution is influenced by the external force (ROS, the wind speed and direction and the topography slope), the geodesic and normal curvatures on an artificial topography. Finally, we show a reconstruction of a real surface fire and complex evolution of the fire perimeter with topological changes on a real variable topography of central Slovakia mountain area.

6.1 Experimental order of convergence of the numerical scheme

The experimental order of convergence (EOC) is calculated as EOC $= \log_2 \frac{\|E_n\|}{\|E_{2n}\|}$, where $\|E_n\|$ is the norm of the space-time error for the curve with n grid points, which is computed by (50) and (54).

To find the EOC, it is necessary to know the exact solution. If the circle on the inclined plane is shrinking by the unit external force, its radius in time is given by $r(t) = 1 - t$. If such circle is evolved by the geodesic curvature, its radius is given by $r(t) = \sqrt{1-2t}$. In the case of a circle on a unit sphere evolving by the geodesic curvature, the radius is given by $r(t) = \sqrt{1 - \left(1 - r_0^2\right) e^{2t}}$, where r_0 is the radius of the initial circle [4]. Time step τ is chosen proportionally either to h_d or to ${h_d}^2$.

In the following paragraphs, we compute the EOC of a circle on an inclined plane and on a sphere. For a circle on an inclined plane, the EOC is computed for both evolutions, driven by a unit external force and by the geodesic curvature. In the second paragraph, we compute the EOC of the same evolution as we did in the first paragraph by using a different measure. We project the numerically computed curves to the horizontal plane and we use the ellipses in the horizontal plane as the exact solution for the computation of EOC. In the last paragraph, we compute the EOC for a circle on a unit sphere evolving by the geodesic curvature.

EOC for a circle evolving on an inclined plane Let us consider the curve evolution on the inclined plane $\varphi(x, y) = x$ [24]. The initial projected curve is given as the ellipse $(x, y) = (a\cos(\gamma), b\sin(\gamma))$, $\gamma \in [0, 2\pi]$, with the semi-major axis length $b = 1$ and the semi-minor axis length $a = \cos\left(\frac{\pi}{4}\right)$. Such curve represents a unit circle (circle with unit radius) on the inclined plane and let it be discretized by a uniform grid. On the other hand, the initial curve on the horizontal plane is an ellipse with non-uniformly distributed points. Therefore, we employ the tangential velocity and we set the parameter $\omega = 15$.

First, we compare the exact radius $r(m\tau)$ and the numerical radius $\left|\mathbf{x}_i^m\right|$ of the shrinking circle (Fig. 10). The discrete L_2-norm is given by

$$\|E_n\| = \sqrt{\sum_{m=1}^{\text{NTS}} \tau \sum_{i=1}^{n} \left(\left|\mathbf{x}_i^m\right| - |r(m\tau)|\right)^2 \frac{h_i^m + h_{i+1}^m}{2}}, \tag{50}$$

where NTS is the number of time steps. Tables 1 and 2 show the EOC of the circle driven by a unit external force $\left(\delta_{\mathcal{F}} = -1,\ \delta_g = 0\right)$ and by the geodesic curvature $\left(\delta_{\mathcal{F}} = 0,\ \delta_g = 1\right)$ considering the coupling $\tau \approx h_d$ and $\tau \approx {h_d}^2$.

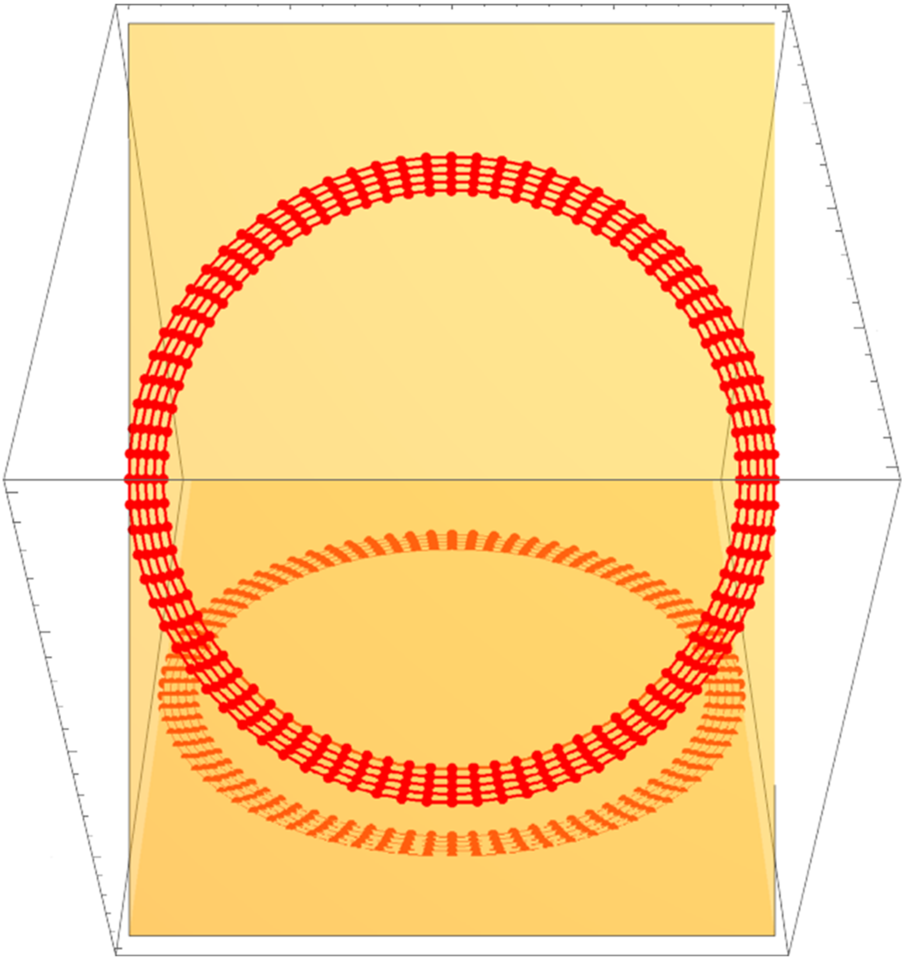


Fig. 10 Shrinking circle with a non-uniform point distribution on the inclined plane and its projection to the horizontal plane with a uniform point distribution

Table 1 The EOC for the circle evolving on the inclined plane with coupling $\tau \approx h_d$

			$\mathcal{F}=-1,\ \delta_g=0$		$\mathcal{F}=0,\ \delta_g=1$	
n	τ	NTS	$\Vert E_n \Vert$	EOC	$\Vert E_n \Vert$	EOC
10	1.00000e-1	1	3.302470e-3	–	6.284157e-3	–
20	5.00000e-2	2	9.045722e-4	1.868238	2.846399e-3	1.142582
40	2.50000e-2	4	2.912492e-4	1.634982	1.336523e-4	1.090653
80	1.25000e-2	8	1.098624e-4	1.406557	6.443453e-4	1.052579
160	6.25000e-3	16	4.662980e-5	1.236373	3.158991e-4	1.028370
320	3.12500e-3	32	2.132691e-5	1.128577	1.563418e-4	1.014760
640	1.56250e-3	64	1.017795e-5	1.067228	7.776370e-5	1.007534
1280	7.81250e-4	128	4.969091e-6	1.034393	3.877938e-5	1.003807

EOC for an ellipse evolving in the horizontal plane This computation of EOC is based on the fact that an ellipse on the horizontal plane is just a vertical projection of a circle on the inclined plane. The exact solution is known to be the ellipse with semi-major axis $b = r(t)$ and semi-minor axis $a = r(t) \cos\left(\frac{\pi}{4}\right)$. The distance d_i^m of $\mathbf{x}_i^m = (x, y)$ and the ellipse can be found by transforming point $\mathbf{x}_i^m$ to adapted ellipsoidal coordinates N, d, θ [15]. To do so, we omit the time index m in the following steps to maintain simplicity. In the first step, we compute the initial value of the angle θ

$$\theta_0 = \arctan\left(\frac{y}{\left(1-\mathrm{e}^2\right)x}\right), \tag{51}$$

where $\mathrm{e}^2 = \frac{\mathrm{a}^2-\mathrm{b}^2}{\mathrm{a}^2}$. In the next steps, we iteratively compute the improved values of θ, distance from the ellipse d_i and the radius of curvature N:

$$N^k = \frac{a}{\sqrt{1-\mathrm{e}^2\sin^2\theta^{\mathrm{k}-1}}},\ (d_i)^k = \frac{x}{\cos\left(\theta^{k-1}\right)} - N^k, \tag{52}$$

Table 2 The EOC for the circle evolving on the inclined plane with coupling $\tau \approx {h_d}^2$

			$\mathcal{F}=-1,\ \delta_g=0$		$\mathcal{F}=0,\ \delta_g=1$	
n	τ	NTS	$\Vert E_n \Vert$	EOC	$\Vert E_n \Vert$	EOC
10	1.00000e-1	1	3.302470e-3	–	6.284157e-3	–
20	2.50000e-2	4	6.284567e-4	2.393660	1.231436e-3	2.351378
40	6.25000e-2	16	1.413407e-4	2.044901	2.887643e-4	2.092378
80	1.56250e-2	64	3.425238e-4	2.012837	7.071423e-4	2.029820
160	3.90625e-3	256	8.487239e-5	2.003887	1.754468e-4	2.010967
320	9.76563e-3	1024	2.116100e-5	2.001303	4.372529e-4	2.004493
640	2.44141e-3	4096	5.285475e-5	2.000491	1.091621e-5	2.001996
1280	6.10352e-4	16384	1.320919e-6	2.000205	2.727283e-5	2.000935

Table 3 The EOC for the projected ellipse evolution using the coupling $\tau \approx h_d$

			$\delta_{\mathcal{F}} = -1,\ \delta_g = 0$		$\delta_{\mathcal{F}} = 0,\ \delta_g = 1$	
n	τ	NTS	$\|E_n\|$	EOC	$\|E_n\|$	EOC
10	1.00000e-1	1	2.813507e-3	–	4.816043e-3	–
20	5.00000e-2	2	7.793186e-4	1.852084	2.219950e-3	1.117321
40	2.50000e-2	4	2.509626e-4	1.634741	1.046447e-3	1.085027
80	1.25000e-2	8	9.457092e-5	1.408004	5.050892e-4	1.050889
160	6.25000e-3	16	4.009806e-5	1.237864	2.477384e-4	1.027721
320	3.12500e-3	32	1.832639e-5	1.129610	1.226322e-4	1.014479
640	1.56250e-3	64	8.742325e-6	1.067834	6.100230e-5	1.007403
1280	7.81250e-4	128	4.267217e-6	1.034721	3.042210e-5	1.003744

$$\theta^k = \arctan\left(\frac{y}{\left(1 - \mathrm{e}^2 \frac{\mathrm{N}^\mathrm{k}}{\mathrm{N}^\mathrm{k} + (\mathrm{d}_\mathrm{i})^\mathrm{k}}\right) x}\right). \tag{53}$$

The limit distance $d_i^m = \lim\limits_{k \to \infty} (d_i)^k$ is used for computing the L_2-norm of the space-time error:

$$\|E_n\| = \sqrt{\sum_{m=1}^{\mathrm{NTS}} \tau \sum_{i=1}^{n} \left(d_i^m\right)^2 \frac{h_i^m + h_{i+1}^m}{2}}. \tag{54}$$

Tables 3 and 4 show the EOC, where the error is computed for the projected curves and the circle on the inclined plane is driven by a unit external force $\left(\delta_{\mathcal{F}} = -1,\ \delta_g = 0\right)$ and the geodesic curvature $\left(\delta_{\mathcal{F}} = 0,\ \delta_g = 1\right)$, considering the coupling $\tau \approx h_d$ and $\tau \approx {h_d}^2$.

EOC for a circle evolving on a unit sphere [4] The evolution on a unit sphere is driven by the geodesic curvature (Fig. 11). The initial projected curve is given as the circle

Table 4 The EOC for the projected ellipse evolution using the coupling $\tau \approx {h_d}^2$

			$\delta_{\mathcal{F}} = -1,\ \delta_g = 0$		$\delta_{\mathcal{F}} = 0,\ \delta_g = 1$	
n	τ	NTS	$\|E_n\|$	EOC	$\|E_n\|$	EOC
10	1.00000e-1	1	2.813507e-3	–	4.816043e-3	–
20	2.50000e-2	4	5.476276e-4	2.361102	9.627732e-4	2.322580
40	6.25000e-3	16	1.233453e-4	2.150492	2.266572e-4	2.086684
80	1.56250e-3	64	2.990653e-5	2.044171	5.556107e-5	2.028366
160	3.90625e-4	256	7.412363e-6	2.012455	1.378814e-5	2.010647
320	9.76563e-5	1024	1.848366e-6	2.003683	3.436444e-6	2.004439
640	2.44141e-5	4096	4.617082e-7	2.001197	8.579233e-7	2.001996
1280	6.10352e-6	16384	1.153921e-7	2.000437	2.143409e-7	2.000942

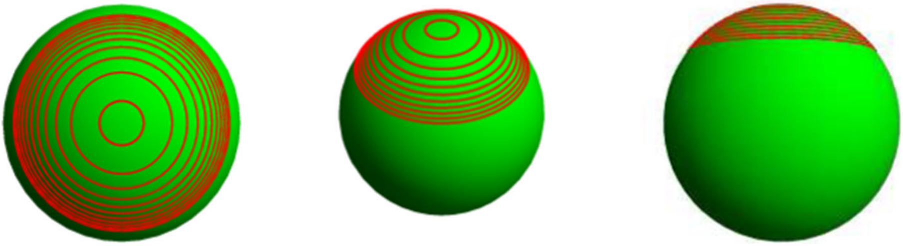


Fig. 11 A circle evolving by its geodesic curvature with the initial radius 0.8 on the unit sphere

$(x, y) = (r_0 \cos(\gamma), r_0 \sin(\gamma))$, $\gamma \in [0, 2\pi]$, with the initial radius $r_0 = 0.8$. The L_2-norm of the space-time error is computed given by (50). Table 5 shows the EOC, where the coupling $\tau \approx {h_d}^2$ was used.

From the numerical experiments, we see that by using the coupling $\tau \approx {h_d}^2$, the method is second-order accurate for both the normal motion $\left(\delta_{\mathcal{F}} = -1,\ \delta_g = 0\right)$ and the (geodesic) curvature-driven motion $\left(\delta_{\mathcal{F}} = 0,\ \delta_g = 1\right)$.

6.2 The influence of the model parameters on the evolution on an artificial topography

The following examples illustrate how the external force (ROS, wind, and terrain slope) and the geodesic and normal curvatures influence the curve expanding on a surface. The surface is given by $\varphi(x, y) = 0.1x^2 + y$. We obtain the initial discrete curve as $\mathbf{x}_i^0 = (4\cos(\gamma_i), 5\sin(\gamma_i))$, where $\gamma_i = \frac{i}{2\pi}$, $i = 1, \ldots, n$. The number of grid points was set to $n = 100$, the time step was chosen as $\tau = \frac{10}{n^2} = 10^{-3}$ min for the number of time steps NTS $= 10,000$ and for the tangential velocity, we set $\omega = 15$. In the examples, we set $f = 1$, $\delta_{\mathcal{F}} = 1$ and we vary δ_g, δ_n, λ_s, and λ_w.

The first example (Fig. 12) compares three curves, green, red, and blue, considering neither the slope nor the wind influence. The green curve is driven only by the unit external force, $f = 1$ and $\delta_{\mathcal{F}} = 1$, the other parameters are equal to zero. In this case, the geodesic distance between the curves in the subsequent time

Table 5 The EOC for the circle evolving on the unit sphere with coupling $\tau \approx {h_d}^2$

			$\delta_{\mathcal{F}} = -1,\ \delta_g = 0$	
n	τ	NTS	$\|E_n\|$	EOC
100	1.00000e-3	5.000e2	1.962130e-3	–
200	2.50000e-4	2.000e3	5.079713e-4	1.949602
400	6.25000e-5	8.000e3	1.282249e-4	1.986070
800	1.56250e-5	3.200e4	3.213590e-5	1.996419
1600	3.90625e-6	1.280e5	8.039000e-6	1.999098
3200	9.76563e-7	5.120e5	2.010070e-6	1.999770

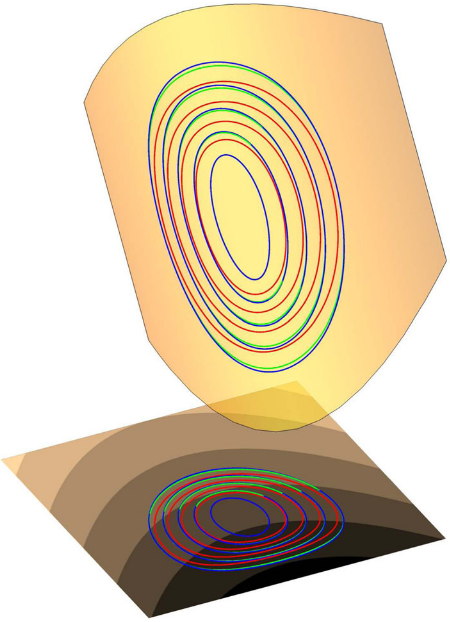


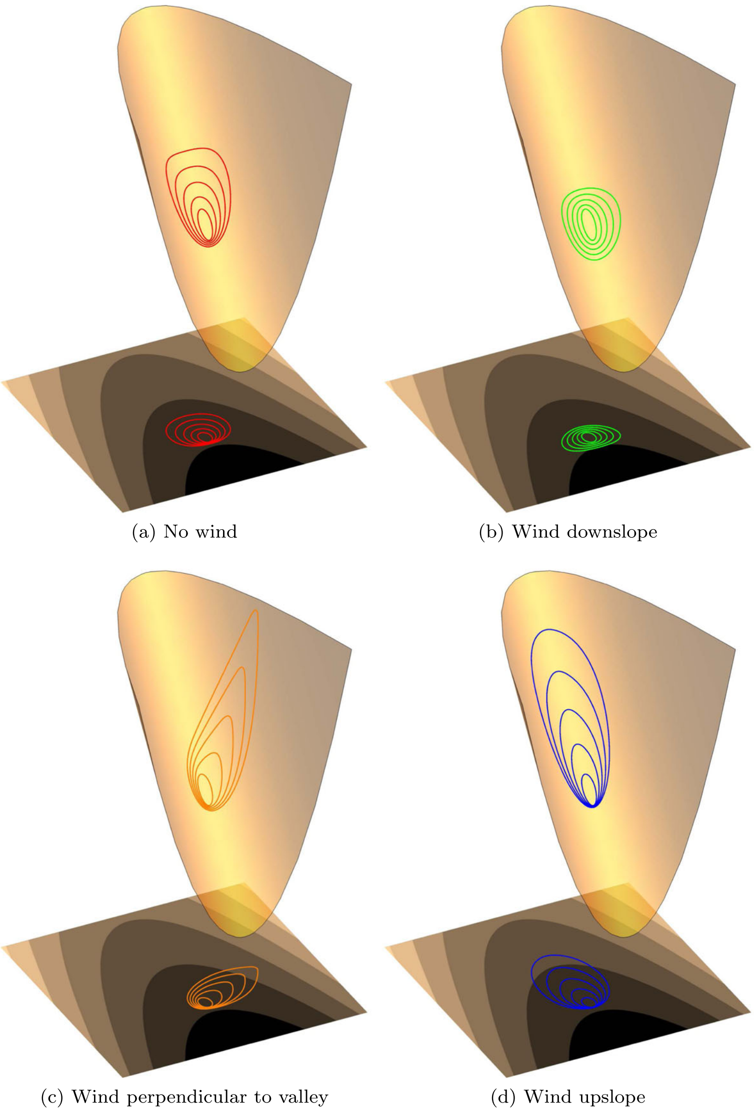
Fig. 12 A curve driven only by a unit external force (green), accompanied by the geodesic curvature (red) and a unit external force and the normal curvature (blue)

steps is constant. The red curve is driven by the external force and the geodesic curvature, $f = 1,\ \delta_{\mathcal{F}} = 1,\ \delta_g = 1,\ \delta_n = 0,\ \lambda_s = 0,\ \lambda_w = 0$. The curve expands although the geodesic curvature slows it down, the most in the middle of the valley. The blue curve is driven by the external force and the normal curvature, $f = 1,\ \delta_{\mathcal{F}} = 1,\ \delta_g = 0,\ \delta_n = 1,\ \lambda_s = 0,\ \lambda_w = 0$. The curve expands as in the first example. Moreover, the normal curvature causes the local acceleration in the middle of the valley.

The above experiments do not simulate the fire spread, since the slope influence is neglected. In the experiment presented in Fig. 13a, we set $f = 1,\ \delta_{\mathcal{F}} = 1,\ \lambda_s = 1,\ \lambda_w = 0,\ \delta_g = 1,\ \delta_n = 1$, i.e., we consider the fuel influence, the terrain slope, and the geodesic and normal curvatures. On the other hand, if we employ the wind at speed 1 mmin^{-1}, $\lambda_w = 1$, we see that the fire spread can be changed significantly. The fire spread can change the direction, e.g., in the case of the wind blowing down the valley (Fig. 13b) or perpendicular to the valley (Fig. 13c) or it can accelerate if the wind blows up the valley (Fig. 13d).

6.3 A real fire reconstruction and the model parameters optimization

In this section, we present the reconstruction of a real grassland fire. Although the fire is small (max. fire perimeter is 79.39 m), it is suitable for finding the model parameters λ_w, λ_s, and δ_g. Since the terrain is nearly an inclined plane, we do not consider the influence of the normal curvature, $\delta_n = 0$. This fire was set as a training for the Fire and Rescue Corps of the Slovak Republic, Ground Forest Fighting Module. They also provided us the video from a quadcopter. The quadcopter video was



(a) No wind

(b) Wind downslope

(c) Wind perpendicular to valley

(d) Wind upslope

Fig. 13 A curve evolution considering the fuel and the slope influence, geodesic and normal curvature, and various wind directions at a constant wind speed 1 mmin^{-1}

processed at the Department of Geodesy, Slovak University of Technology, using a photogrammetry software to obtain a vertical projection of the fire spread on a real terrain to the horizontal plane in a 5-s time interval. Then, the fire perimeters were segmented manually with the uniform segment length 0.15m in a CAD software.

The inputs to our model were as follows:

- the initial condition was given by the segmented fire perimeter 55 s after the ignition,

- the homogeneous rate of spread was 1.18 mmin^{-1} (measured at place),
- the wind velocity and direction was (North = 0 deg, East = 90 deg), see columns 6 and 7 of the Table 6 (measured at place),
- the terrain slope was given by the digital terrain model.

The desired curve segment length was set to $h_d = 0.15$ m, which corresponds to the manual segmentation spatial step and the time step was chosen as $\tau = \frac{h_d^2}{22.5} = 10^{-3}$ min, i.e., it is proportional to h_d^2.

The mean Hausdorff distance (MHD) [17, 39] was used as the criterion in the inverse modeling [30], i.e., we look for the optimal model parameters which give the minimal MHD between the numerically computed and the manually segmented fire perimeters. First, we look for the optimal parameters of the model, constant over every 5-s time interval where we have provided the manually segmented curves (see Table 6), and then we find the optimal model parameters, constant in the whole time range.

Table 6 The review of the model parameters λ_w, λ_s, δ_g (columns 2–5) and the wind direction estimated by the inverse modeling, and the measured wind direction and velocity (columns 6–7)

Time	Inverse modeling values				Measured values		MHD
	λ_w	λ_s	δ_g	Wind dir.	Wind dir.	Wind velocity	
[s]	[−]	[−]	[−]	[deg]	[deg]	$[\mathrm{mmin}^{-1}]$	[m]
55–60	0.021	0.25	0.70	356.5	0	180	0.084
60–65	0.011	0.25	0.50	352.0	0	180	0.106
65–70	0.023	0.25	1.90	94.5	0	120	0.194
70–75	0.029	0.10	0.90	37.0	0	120	0.221
75–80	0.025	0.30	1.95	5.0	0	240	0.297
80–85	0.018	0.50	2.00	5.0	0	360	0.351
85–90	0.019	0.15	2.00	351.5	0	360	0.321
90–95	0.014	0.10	1.15	1.0	0	360	0.293
95–100	0.012	0.50	0.60	357.5	0	360	0.345
100-105	0.014	0.45	0.75	0.5	0	360	0.327
105–110	0.017	0.10	1.05	359.5	0	360	0.340
110–115	0.014	0.35	1.05	358.5	0	360	0.334
115–120	0.010	0.10	0.50	300.0	0	180	0.317
120–125	0.022	0.50	1.00	354.0	0	180	0.321
125–130	0.021	0.10	0.55	340.0	0	180	0.351
130–135	0.025	0.10	1.20	342.0	0	180	0.424
Average	0.018	0.26	1.11	–	–	–	0.289

The last column presents the mean Hausdorff distance (MHD) between the computed and the manually segmented curves at the end of the time intervals indicated in the first column. The last row presents the averaged values of some estimated and computed quantities

The MHD between two point sets $A = \left\{a_1, ..., a_{n_A}\right\}$ and $B = \left\{b_1, ..., b_{n_B}\right\}$ is defined by the following formula

$$\mathrm{MHD}\,(A, B) = \max\left(\mathrm{mhd}\,(A, B)\,, \mathrm{mhd}\,(B, A)\right), \tag{55}$$

where the so-called mean directed Hausdorff distance $\mathrm{mhd}\,(A, B)$ is given by

$$\mathrm{mhd}\,(A, B) = \frac{1}{n_A}\sum_{i=1}^{n_A}\min_{\widehat{b}\in B}\left\|a_i - \widehat{b}\right\|, \tag{56}$$

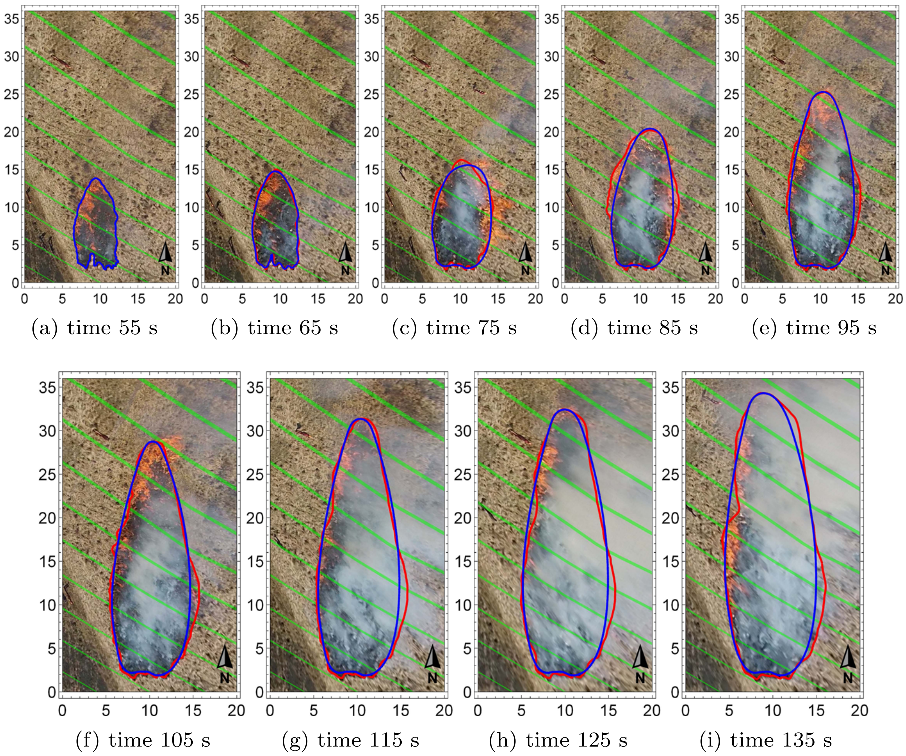
where n_A is the number of points of the point set A, a_i is the i-th point from the point set A, and $\widehat{b}$ is the linear segment between two neighboring points from the point set B.

For finding the optimal model parameter values, we assume $\lambda_w \in \langle 0.005, 0.03\rangle$, $\lambda_s \in \langle 0.1, 0.5\rangle$, and $\delta_g \in \langle 0.5, 2.0\rangle$. In every optimization step, we simply compute MHD for every combination of parameter values in the abovementioned ranges with respect to the discrete step 0.001 for λ_w and 0.05 for λ_s and δ_g.

Although the wind direction, used as an input, was measured at place as constantly north (0 deg), the quadcopter video shows a variable wind direction. Thus, we find also the wind direction with 0.5 deg step for wind directions from 0 deg to 360 deg.

The overview of the estimated model parameters, wind directions, and computed MHD is shown in Table 6. However, in our model, we assume constant model parameters while the fuel and weather conditions remain the same. Therefore, we subsequently estimate the constant values of the model parameters λ_s, δ_g, and λ_w. First, we take the average value $\lambda_s = 0.26$ from the first estimation (see Table 6) and look for the optimal values of the two other parameters λ_w and δ_g with respect to MHD in every 5-s interval. Interestingly, this new optimal model parameter estimation leads to the average $\overline{\mathrm{MHD}} = 0.285$m, which is slightly less than the averaged MHD after the first optimization step. Then, we take the average of δ_g over all time steps to be 1.084 and consider the same constant $\lambda_s = 0.26$ in order to estimate just the optimal values of λ_w. In this last optimization step, we obtained the averaged $\overline{\mathrm{MHD}} = 0.281$m, and we consider the time averaged $\lambda_w = 0.0185$ as the last optimal parameter value. We consider constant values $\lambda_s = 0.26$, $\delta_g = 1.084$, and $\lambda_w = 0.0185$ as the optimal model parameters resulting from the inverse modelling and the further computations can be done with such parameter values in similar conditions.

We measure the accuracy of our fire spread reconstruction by computing the ratio MHD/L in every 5-s time interval, where L is the length of the segmented curve. This measure is in range 0.2–0.7% in every time moment. We assume that the slight differences between the segmented and the numerically computed curves (see Fig. 14) could be caused by the fuel heterogeneity (e.g., variable moisture, which is not included in our model yet) and more dynamic changes in wind direction than the measured and computed in 5-s intervals. The fuel heterogeneity can be seen, e.g.,



(a) time 55 s (b) time 65 s (c) time 75 s (d) time 85 s (e) time 95 s

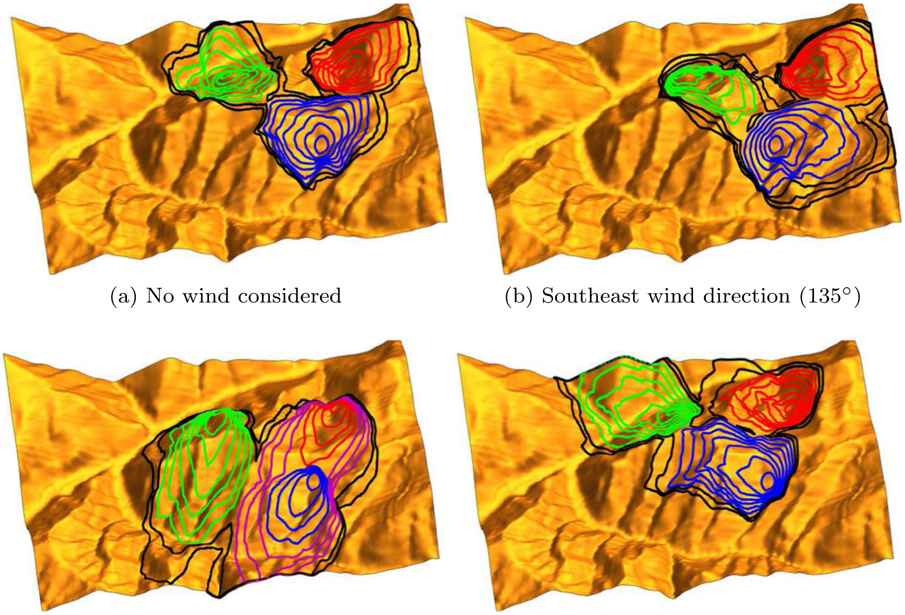
(f) time 105 s (g) time 115 s (h) time 125 s (i) time 135 s

Fig. 14 A real fire reconstruction. We compare the manually segmented curve (red) and the curve computed numerically (blue). The initial condition is given by the segmented curve in time 55 s. The contour lines of the digital terrain model (green) are thicker with increasing elevation

in interval 130–135 s (see Fig. 14i) where the segmented fire perimeter is locally spreading faster. More dynamic changes in the wind direction are obvious, e.g., in the interval 80–85 s (see Fig. 14d) where the segmented fire perimeter is wider than the computed one. Also, the segmentation error is non-negligible due to the low visibility of fire perimeter through the smoke, especially in the last time intervals (see Fig. 14h, i).

6.4 Simulation on a real topography

In this subsection, we demonstrate the flexibility of our surface fire spread model on the real variable topography. The area of Staré Hory in the central Slovakia was chosen since a repeated fire occurred here in the last two decades. The area of 11 km^2 is given by the digital terrain model in 10-m resolution. The wind velocity is set to 8 mmin^{-1}. These simulations, considering the model parameters in the range of the estimated values from the inverse modeling in the previous subsection, have the parameters set as follows: $\delta_{\mathcal{F}} = 1$, $\lambda_s = 0.25$, $\lambda_w = 0.02$,



(a) No wind considered

(b) Southeast wind direction (135°)

(c) South-Southwest wind direction (202.5°)

(d) West-Northwest wind direction (292.5°)

Fig. 15 A large-scale simulation in the area of Staré Hory in the central Slovakia, considering a homogenous fuel, the wind speed 8 m.min^{-1} and various wind directions. The initial condition is given by 3 circles in the vertical projection. Their merging is signalized by the color change (e.g., violet is merged blue and red curve)

$\delta_g = 1,\ \delta_n = 1,\ \omega = 20$, $h_d = 2$ m and $\tau = 0.05$ min. The number of time steps is 9600.

First, we present the simulations with a homogeneous ROS, $f = 1$, and different wind directions (see Fig. 15). The initial condition is given by three circles (red, green, and blue) in the vertical projection and their merging is signalized by a color change.

In the next simulation (see Fig. 16), we consider a southeast wind (135 deg) and a heterogenous fuel, i.e., a variable ROS given by the ROS map in resolution 0.83 m. The ROS map, made from the forestry typological map, consists of four colors and the ROS were assigned from [33]:

- black (roads, rivers, settlements) — non burnable
- dark grey (broad-leaved forest) — low rate of spread 0.46 mmin^{-1}
- light grey (mixed forest) — medium rate of spread 0.76 mmin^{-1}
- white (coniferous forest) - highest rate of spread 1 mmin^{-1} .

This simulation of 8-h fire spread took 42.96 s of computational time with 9600 time steps, including the detection of topological changes. The number of grid points varied from 500 points in the first time step to 3668 points in the last time step and 2921.28 points per time step at an average.

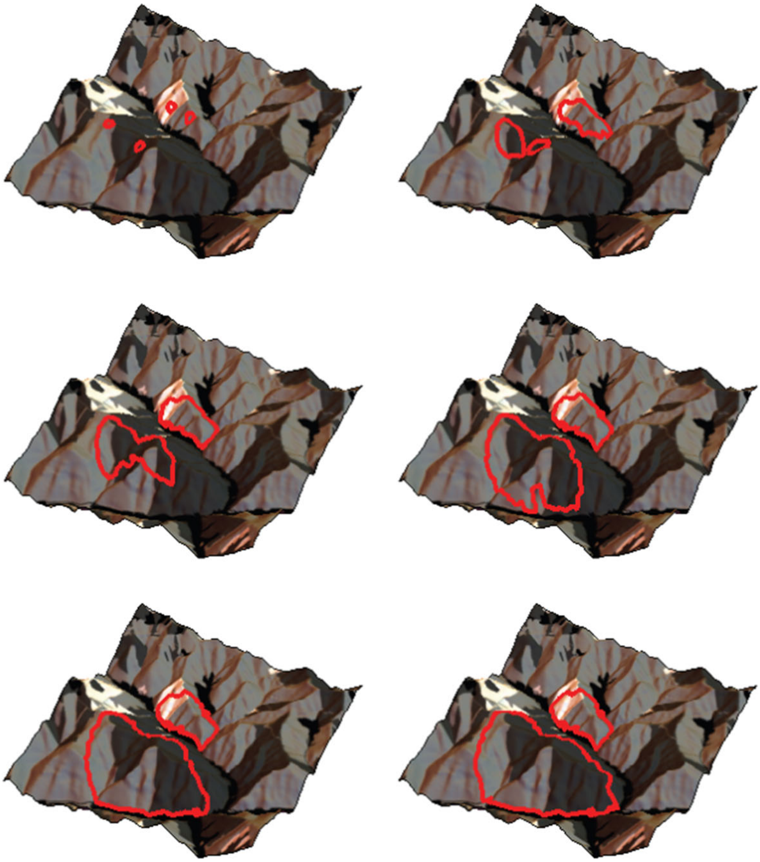


Fig. 16 Large-scale simulation in the area of Staré Hory in central Slovakia, considering heterogenous fuel and northwest wind (315°) at speed 8 $\mathrm{m\,min}^{-1}$. Initial condition is given by 4 circles in the horizontal projection

7 Conclusions

In this paper, we introduced a new fast and stable forest fire propagation model based on the Lagrangian approach. We described the surface curve, representing the fire perimeter and its projection into a 2-D planar curve. Next, we prescribed the mathematical model for the fire perimeter spread over a heterogeneous fuel on a variable terrain, influenced by the wind direction and velocity. Moreover, we considered the influence of the surface fire perimeter shape itself through the geodesic and normal curvatures. The discrete formulation of our model was based on the semi-implicit approach for the curvature term, and the inflow-implicit/outflow-explicit and upwind scheme approaches for the advection term, which allowed reasonable computational time step choice. We used the tangential velocity to asymptotically uniformly redistribute the grid points along the curve. In addition, we maintained the desired segment length thanks to our strategy for adding and removing points and the used tangential velocity. This is useful for our $O(n)$ approach for topological changes treatment of splitting and merging of curves. Our numerical model turned out to be second-order accurate, using the coupling $\tau \approx {h_d}^2$, as was shown in EOC experiments on various surfaces. The grassland fire experiment was used to estimate the model parameters. The numerical results revealed the ability of our model to reconstruct the real fire perimeter with a negligible error ($< 1\%$). Finally, we showed a large-scale simulation with topological changes in a mountainous area, considering a heterogeneous rate of spread influenced by the wind. Such comprehensive simulation of an 8-h fire spread took 42.96 s, and thus one can possibly expect the inverse modeling of selected parameters in real time.

Funding information This work was supported by the grants VEGA 1/0608/15 and APVV-15-0522.

References

1. Alexandrian, D.: Vesta - large scale fire simulator. http://www.fireparadox.org/large_scale_fire_simulator.php. Accessed on 25 Oct 2016
2. Balažovjech, M., Mikula, K.: A higher order scheme for a tangentially stabilized plane curve shortening flow with a driving force. SIAM J. Sci. Comput. **33**(5), 2277–2294 (2011)
3. Balažovjech, M., Mikula, K., Petrášová, M., Urbán, J.: Lagrangean method with topological changes for numerical modelling of forest fire propagation. In: Proceedings of ALGORITMY, pp. 42–52 (2012)
4. Barrett, J.W., Garcke, H., Nürnberg, R.: Numerical approximation of gradient flows for closed curves in $\mathbb{R}^d$. IMA J. Numer. Anal. **30**(1), 4–60 (2009)
5. Benninghoff, H., Garcke, H.: Efficient image segmentation and restoration using parametric curve evolution with junctions and topology changes. SIAM J. Imag. Sci. **7**(3), 1451–1483 (2014)
6. Benninghoff, H., Garcke, H.: Image segmentation using parametric contours with free endpoints. IEEE Trans. Image Process. **25**(4), 1639–1648 (2016)
7. Benninghoff, H., Garcke, H.: Segmentation and restoration of images on surfaces by parametric active contours with topology changes. J. Math. Imag. Vis. **55**(1), 105–124 (2016)
8. Benninghoff, H., Garcke, H.: Segmentation of three-dimensional images with parametric active surfaces and topology changes. J. Sci. Comput., 1–35 (2017)
9. Bose, C., Bryce, R., Dueck, G.: Untangling the Prometheus nightmare. In: Proc. 18th IMACS World Congress MODSIM09 and International Congress on Modelling and Simulation, pp. 13–17. Cairns (2009)
10. Butler, B., Anderson, W., Catchpole, E.: Influence of slope on fire spread rate. In: The Fire Environment–Innovations, Management, and Policy; Conference Proceedings, pp. 75–82 (2007)
11. Dziuk, G.: Discrete anisotropic curve shortening flow. SIAM J. Numer. Anal. **36**(6), 1808–1830 (1999)
12. Finney, M.A. et al.: FARSITE Fire Area Simulator–model development and evaluation, vol. 3. US Department of Agriculture, Forest Service, Rocky Mountain Research Station Ogden, UT (1998)
13. Hou, T.Y., Klapper, I., Si, H.: Removing the stiffness of curvature in computing 3-d filaments. J. Comput. Phys. **143**(2), 628–664 (1998)
14. Hou, T.Y., Lowengrub, J.S., Shelley, M.J.: Removing the stiffness from interfacial flows with surface tension. J. Comput. Phys. **114**(2), 312–338 (1994)
15. Subirana, J.S., Zornoza, J.M.J., Hernández-Pajares, M.: Ellipsoidal and cartesian coordinates conversion. http://www.navipedia.net/index.php/Ellipsoidal_and_Cartesian_Coordinates_Conversion (s). Accessed 10 Jun 2016
16. Krasnow, K., Schoennagel, T., Veblen, T.T.: Forest fuel mapping and evaluation of LANDFIRE fuel maps in Boulder County, Colorado, USA. For. Ecol. Manage. **257**(7), 1603–1612 (2009)
17. Krivá, Z., Mikula, K., Peyriéras, N., Rizzi, B., Sarti, A., Stašová, O.: 3d early embryogenesis image filtering by nonlinear partial differential equations. Med. Image Anal. **14**(4), 510–526 (2010)

18. Lopes, A., Cruz, M., Viegas, D.: Firestation: An integrated software system for the numerical simulation of fire spread on complex topography. Environ. Modell. Software **17**(3), 269–285 (2002)
19. McDermott, R., McGrattan, K., Hostikka, S.: Fire dynamics simulator (version 5) technical reference guide NIST. Spec. Publ. **1018**(5) (2008)
20. Mell, W.E. et al.: The wildland–urban interface fire problem–current approaches and research needs. Int. J. Wildland Fire **19**(2), 238–251 (2010)
21. Mikula, K., Ohlberger, M.: A new level set method for motion in normal direction based on a semi-implicit forward-backward diffusion approach. SIAM J. Sci. Comput. **32**(3), 1527–1544 (2010)
22. Mikula, K., Ohlberger, M., Urbán, J.: Inflow-implicit/outflow-explicit finite volume methods for solving advection equations. Appl. Numer. Math. **85**, 16–37 (2014)
23. Mikula, K., Ševčovič, D.: Evolution of plane curves driven by a nonlinear function of curvature and anisotropy. SIAM J. Appl. Math. **61**(5), 1473–1501 (2001)
24. Mikula, K., Ševčovič, D.: Computational and qualitative aspects of evolution of curves driven by curvature and external force. Comput. Vis. Sci. **6**(4), 211–225 (2004)
25. Mikula, K., Ševčovič, D.: A direct method for solving an anisotropic mean curvature flow of plane curves with an external force. Math. Methods Appl. Sci. **27**(13), 1545–1565 (2004)
26. Mikula, K., Ševčovič, D.: Evolution of curves on a surface driven by the geodesic curvature and external force. Appl. Anal. **85**(4), 345–362 (2006)
27. Mikula, K., Urbán, J.: New fast and stable lagrangean method for image segmentation. In: 2012 5th International Congress on Image and Signal Processing (CISP), pp. 688–696. IEEE (2012)
28. Mikula, K., Urbán, J.: A new tangentially stabilized 3D curve evolution algorithm and its application in virtual colonoscopy. Adv. Comput. Math. **40**(4), 819–837 (2014)
29. Monoši, M., Majlingová, A., Kapusniak, J.: Lesné požiare Žilinská univerzita v Žiline (2015)
30. Nakamura, G., Potthast, R.: Inverse Modeling, pp. 2053–2563. IOP Publishing (2015). https://doi.org/10.1088/978-0-7503-1218-9
31. Osher, S., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Applied Mathematical Sciences. Springer, New York (2002)
32. Pauš, P., Beneš, M.: Algorithm for topological changes of parametrically described curves. In: Proceedings of ALGORITMY, pp. 176–184 (2009)
33. Prichard, S.J. et al.: Fuel characteristic classification system version 3.0: Technical documentation. Tech. rep., U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station (2013)
34. Scott, J.H., Burgan, R.E.: Standard fire behavior fuel models: a comprehensive set for use with rothermel's surface fire spread model. The Bark Beetles, Fuels, and Fire Bibliography, p. 66 (2005)
35. Sethian, J.A.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press (1999)
36. Sullivan, A.: A review of wildland fire spread modelling, 1990-present 3: Mathematical analogues and simulation models. arXiv:0706.4130 (2007)
37. Vakalis, D. et al.: A GIS based operational system for wildland fire crisis management i. Mathematical modelling and simulation. Appl. Math. Model. **28**(4), 389–410 (2004)
38. Viegas, D. et al.: Slope and wind effects on fire spread. In: IVth International Forest Fire Conference. Coimbra (Portugal). FFR & Wildland Fire Safety. Millpress, Rotterdam (2002)
39. Zhang, J.W., Han, G.Q., Wo, Y.: Image registration based on generalized and mean hausdorff distances. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005, vol. 8, pp. 5117–5121. IEEE (2005)

Affiliations

Martin Ambroz[1] · Martin Balažovjech[1] · Matej Medl'a[1] · Karol Mikula[1,2]

Martin Balažovjech
balazovjech@math.sk

Matej Medl'a
medla@math.sk

Karol Mikula
mikula@math.sk

[1] Department of Mathematics, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, 810 05 Bratislava, Slovak Republic

[2] Algoritmy:SK s.r.o., Šulekova 6, 81106 Bratislava, Slovak Republic