Nat. Hazards Earth Syst. Sci., 14, 1491–1503, 2014
www.nat-hazards-earth-syst-sci.net/14/1491/2014/
doi:10.5194/nhess-14-1491-2014

Natural Hazards and Earth System Sciences
Open Access

Forecasting wind-driven wildfires using an inverse modelling approach

O. Rios[1], **W. Jahn**[2], **and G. Rein**[3]

[1]Centre for Studies on Technological Risk (CERTEC), Universitat Politècnica de Catalunya, Av. Diagonal, 647, 08028
Barcelona, Spain
[2]Departamento de Ingeniería Mecánica y Metalúrgica, Pontificia Universidad Católica de Chile, Santiago, Chile
[3]Department of Mechanical Engineering, Imperial College London, SW72AZ, London, UK

Correspondence to: G. Rein (g.rein@imperial.ac.uk)

Received: 30 September 2013 – Published in Nat. Hazards Earth Syst. Sci. Discuss.: 4 December 2013
Revised: 10 April 2014 – Accepted: 15 April 2014 – Published: 13 June 2014

Abstract. A technology able to rapidly forecast wildfire dynamics would lead to a paradigm shift in the response to emergencies, providing the Fire Service with essential information about the ongoing fire. This paper presents and explores a novel methodology to forecast wildfire dynamics in wind-driven conditions, using real-time data assimilation and inverse modelling. The forecasting algorithm combines Rothermel's rate of spread theory with a perimeter expansion model based on Huygens principle and solves the optimisation problem with a tangent linear approach and forward automatic differentiation. Its potential is investigated using synthetic data and evaluated in different wildfire scenarios. The results show the capacity of the method to quickly predict the location of the fire front with a positive lead time (ahead of the event) in the order of 10 min for a spatial scale of 100 m. The greatest strengths of our method are lightness, speed and flexibility. We specifically tailor the forecast to be efficient and computationally cheap so it can be used in mobile systems for field deployment and operativeness. Thus, we put emphasis on producing a positive lead time and the means to maximise it.

1 Towards an operative forecasting tool

Current computational wildfire dynamics simulators are not fast enough to provide valid predictions on time (Sullivan, 2009) and require input parameters that are difficult to acquire and sense during an emergency situation. A potential solution to develop an operational forecasting tool is to assimilate real-time sensor data (Cowlard et al., 2010), which has been shown to greatly accelerate fire simulations without loss of accuracy (Mandel et al., 2008; Jahn et al., 2011; Rochoux et al., 2013). The cornerstone to reach such a tool is to find a computational algorithm that combines a fire model with sensor data that reliably delivers an accurate forecast with a positive lead time (i.e. time before the event, in the order of 10 min for a spacial scale of 100 m), and enables emergency responders to make better tactical decisions. At the same time, it has to be versatile enough to be adapted in different fire situations (range of fuels, complex topography, weather conditions). Ideally, it should also be able to incorporate the effect of fire fighting actions (e.g. water lines, fire breaks, back fires) and weather forecasts. More importantly, it should not require high computational resources (i.e. high-performance computing or supercomputers) so that it can also be deployed flexibly in portable devices by fire responders.

1.1 Data assimilation and inverse modelling

Inverse modelling, which is the core of data assimilation techniques, consists of studying measurements from sensors to gain information about the physical phenomena using a variety of mathematical models and algorithms. This new information is then used to forecast the future evolution of the phenomena. Instead of just writing the outputs of a model, inverse modelling exploits the sensor outputs and aims to unveil the governing parameters and the boundary conditions of the problem.

Published by Copernicus Publications on behalf of the European Geosciences Union.

The inverse method is particularly appropriate for wildfire modelling due to the large amount of unknowns. The fuel properties, location, area covered by foliage, moisture content, meteorological conditions and topography are necessary parameters to initialise a fire model but all of them can hardly ever be measured. By contrast, the inverse approach can find these parameter by using a range of sensor measurements of the ongoing fire.

Despite its promising capacity for coping with complex problems with a large number of unknown parameters, only few authors have tried to apply data assimilation or forecasting techniques to fire science. Among these, Jahn et al. (2011, 2012) successfully pioneered the approach to forecast fires in enclosures using both simple and complex models (two-zone model and computational fluid dynamics). In the field of wildfire, Mandel et al. (2009) explored this technique to predict the time–temperature curve of a sensor placed in the way of an advancing wildfire. They examined a reaction–diffusion equation and a semi-empirical fire line propagation model coupled with an Eulerian level-set-based equation. Despite this progress, their implementation was found to be unstable due to the generation of spurious fires which cause non-physical results.

Rochoux et al. (2013) pioneered the successful application of data assimilation to predict the location and spread of the wildfire front using infra-red sensors. Data were assimilated with a Kalman filter to balance computational and sensor errors. Rochoux et al. (2013) assimilates perimeter locations at different times and uses the fuel depth as the only input. The propagating model uses two components: the rate of spread (RoS) is represented by a product between the fuel depth (δ) and a constant (Γ) to be quantified as part of the forecasting problem ($\mathrm{RoS} = \Gamma \cdot \delta$). Their model uses a level-set-based equation to cast the fire perimeter. They tested the model in a controlled small-scale experiment assimilating one fire front and delivering a 30 s forecast. Most recently, Rochoux et al. (2014) have presented a work that solves for more than one parameter and uses parallel computing. Also using a level-set model and sensor data, Lautenberger (2013) explored stochastic optimisation of the wildfire problem. Coen et al. (2013) used satellite data to initialise a weather–wildfire growth model at the kilometre scale.

The greatest strengths of our method presented here are lightness, speed and flexibility. We specifically tailor the forecast to be efficient and computationally cheap so it can be used in mobile systems for field deployment. Thus, we put emphasis on producing a positive lead time and the means to maximise it, while at the same time solving for multiple parameters. These are not the objectives of other papers in the literature. For example, Rochoux et al. (2013), the truly first paper in the literature that effectively forecast wildfire behaviour, integrates measurement errors with model errors to increase accuracy (standard procedure in weather forecasting), but it comes at the price of higher computational expense. Rochoux et al. (2013) and Mandel et al. (2008) use one single parameter at a time and do not emphasize lead times. Moreover, they seem tailored more towards supercomputing platforms than to mobile systems for field deployment.

Another highlight of our method is the incorporation of automatic differentiation into the inverse model, which is accurate and fast, further decreasing the computational expense of a forecast.

1.2 Forecasting algorithm

We formulate the inverse problem based on the premise that some invariant exists by following the contributions of Jahn et al. (2012) on forecasting fire dynamics in enclosures. We define the invariants as the set of governing parameters that are mutually independent and constant for a significant amount of time. Invariant is a concept already in the literature (Jahn et al., 2012). Therefore, our implementation relies on the assumption that some physical attributes of the system remain constant at least during some time. Those attributes can be uniform, a scalar or a vector field with spatial dependency. From the point of view of our methodology, invariants are a central concept to forecasting systems that do not focus on the initial conditions only. For example, weather forecasting (i.e. Coen et al., 2013) solves an inverse problem to find the initial conditions, and then runs the forward model for predictions. In our work, we solve the inverse model of selected key parameters inside the governing equations, the invariants, not the initial conditions. It is an essential property of the invariants that they remain constant during the lead time of the forecast. When any invariant changes significantly (e.g. due to divergence of the assumptions or external conditions) its effect is to limit the lead time. Examples of such quantity are initial fuel's moisture content or fuel depth. However, the invariants are not restricted to physical variables but can represent mathematical attributes of the system as well. For instance, if the wind speed changes but its effect on the RoS remains constant (boundary layer regime is maintained) the most important invariant will be its effect on the rate of spread rather than the wind speed itself.

After assimilating data during a period of time (assimilation window) the invariants are estimated and used to forecast the perimeter evolution. This forecast is then accurate until any of the invariants change significantly, which would be detected with the help of the continuous data feed from sensors. The sensor errors in the assimilated data are considered to be smaller than the model accuracy and therefore their influence is not directly considered here. This is a complementary approach to that of Rochoux et al. (2013) who balance data errors with model errors.

Regarding sensor data feeds, in the present work we consider fire front positions hypothetically supplied by airborne observations, or ground crews.

However, additional data such as flame height or spreading rate (recently measured by infra-red images and stereo vision; Rossi et al., 2013) could be considered in future developments.

2 Building up the forward model

The initial step when posing an inverse modelling problem is to determine the forward model and its invariants. The forward model is the set of equations that relates the invariants to the observables (variables that can be measured with sensors). Its importance in the is twofold: the forward model is first used iteratively to quantify the invariants and then run again to deliver a forecast valid until the invariants change or the next assimilation process is started.

To create our forward model, we combined Rothermel (1972) and Richards (1990) models. The Rothermel model estimates the RoS of any point in the fire front whereas the Richard model uses these RoS to generate the elliptical firelets that expand the fire front and compute its location at any time.

2.1 Rothermel's model

Rothermel's model is based on an energy balance equation where the heat sources and sinks are identified to estimate the RoS of a surface fire. The original model uses several empirical correlations from wind-tunnel experiments for fires spreading at quasi-steady state. This means that any acceleration of the fire is not considered. The shape of the fire front is assumed to have no influence on the RoS.

Rothermel's equation can be recast with three invariants (I_x), defined as follows:

$$\mathrm{RoS} = I_\mathrm{mf}(1 + I_\mathrm{u} \cdot I_\mathrm{w}). \tag{1}$$

I_mf captures the effect of all the fuel properties; ovendry fuel loading (w_o), surface-area-to-volume ratio (σ), moisture content (M_f), moisture of extinction (M_x) and fuel depth (δ):

$$I_\mathrm{mf} = \mathcal{F}(\sigma, w_\mathrm{o}, \delta, M_\mathrm{f}, M_\mathrm{x}). \tag{2}$$

The wind speed is directly equal to I_u:

$$I_\mathrm{u} = U. \tag{3}$$

The effect of the wind speed on the fire spread also depends on fuel properties such as layout, bulk density, surface-area to volume ratio and fuel depth. Its effect is embedded in I_w as

$$I_\mathrm{w} = \mathcal{K}(\sigma, w_\mathrm{o}, \delta) \cdot U^{B-1}, \tag{4}$$

where B is an empirical coefficient.

2.2 Huygens principle

Although Rothermel's model can estimate the RoS of any point, it is a mean value for the head fire (Rothermel, 1972) and does not inform about different directions of spread. Therefore, it is not sufficient in predicting the fire front shape and location. In parallel to RoS estimation, some other model must be used to represent the fire perimeter expansion. We used Huygens' principle – originally postulated to explain light wavefront propagation – with elliptical expansion, as proposed by Richards (1993). Applying it to wildfire, this principle considers every point in the fire perimeter at time t as a new ignition source that spreads during a time dt following an elliptical template shape – known as a *firelet*. The corresponding fire front line at time $t + \mathrm{d}t$ is the outer curve that envelopes the firelets centred on the rear focus as shown in Fig. 1.

The details of the Huygens firelet model can be found in Richards (1990, 1993), but an overview of the main concepts and equations is provided here.

Considering the initial ignition point situated at $\{X_0, Y_0\}$ and using a parametrisation variable $s \in [0 - 2\pi]$, the $\{(x_i(t), y_i(t)\}$ coordinates of fire front vertices can be analytically calculated by integrating a set of partial differential equations:

$$x(s,\hat{t}) = X_0 + \int_0^{\hat{t}} \left(\frac{a^2(t)\cos\theta(t)\cos(K) + b^2(t)\sin\theta(t)\sin(K)}{\sqrt{a^2(t)\cos^2(K) + b^2(t)\sin^2(K)}} \cdot c(t)\sin\theta(t) \right) \mathrm{d}t \tag{5}$$

$$y(s,\hat{t}) = Y_0 + \int_0^{\hat{t}} \left(\frac{a^2(t)\sin\theta(t)\cos(K) + b^2(t)\cos\theta(t)\sin(K)}{\sqrt{a^2(t)\cos^2(K) + b^2(t)\sin^2(K)}} \cdot c(t)\cos\theta(t) \right) \mathrm{d}t, \tag{6}$$

where

$$K = \theta(t) + s, \tag{7}$$

where θ is the wind direction and b and c are related to the backwards and to the forward propagation velocities that can vary spatially and are calculated by imposing Rothermel's rate of spread for the head fire from the new ignition point:

$$b(s,t) + c(s,t) \equiv \mathrm{RoS}(s,t). \tag{8}$$

The lateral front velocity a, however, is directly related to the eccentricity of the firelet. It was originally estimated using an experimental correlation found by Anderson (1983) that relates the ratio between the major and the minor firelet's axis, and thus, the ratio between b and a (independent of the time step Δt used). Its value depends on the wind speed (U) in accordance with the equation

$$\frac{a(s,t)}{b(s,t)} = 0.936e^{0.2566U} + 0.461e^{-0.1548U} - 0.397 \equiv \mathrm{LB}. \tag{9}$$

Note that the power coefficients in this empirical equation have units of [$\mathrm{s\,m^{-1}}$], LB is called length-to-breadth ratio and accounts for the eccentricity of the elliptical firelets. The

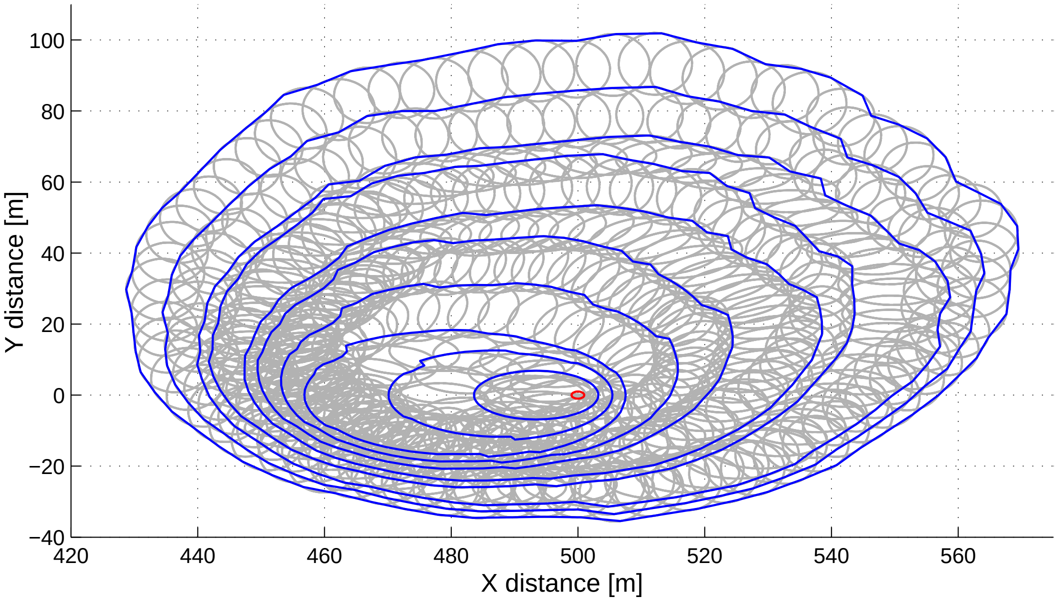

Figure 1. Example of Huygens expansion with elliptical firelets (grey lines) from an ignition point (red dot in the centre). Ten fire fronts (blue lines) are plotted during the spread of the fire over an heterogeneous fuel depth (0.6 ± 0.3 m) and under changing wind ($5 \pm 2\,\mathrm{m\,s^{-1}}$) speed and direction at every time step (1 min each). This Rothermel–Huygens model is also used in FARSITE (Finney, 1998).

constant 0.397 is a modification of Anderson's original formula to ensure that the fire expands circularly (LB = 1) under no-wind conditions ($U = 0$).

Once the LB, a, b, c velocities can be calculated using Eqs. (8) and (9) and the elliptical geometry properties:

$$a = \mathrm{RoS}\frac{1+1/\mathrm{HB}}{2\mathrm{LB}} \tag{10}$$

$$b = \mathrm{RoS}\frac{1+1/\mathrm{HB}}{2} \tag{11}$$

$$c = b - \frac{\mathrm{RoS}}{\mathrm{HB}}, \tag{12}$$

where

$$\mathrm{HB} = \frac{\mathrm{LB}+\sqrt{\mathrm{LB}^2-1}}{\mathrm{LB}-\sqrt{LB^2-1}}.$$

If the invariant $I_\mathrm{u} = U$, introduced in Rothermel's model, is reused in Huygens' firelets expansion, only one additional invariant is required to account for the principal direction of spreading determined by the wind direction:

$$I_\theta = \theta. \tag{13}$$

The forward model is then a function of four invariants:

$$\mathcal{M}(I_\mathrm{u}, I_\mathrm{w}, I_\mathrm{mf}, I_\theta, T) = \begin{cases} \mathrm{RoS} = \mathcal{R}(I_\mathrm{u}, I_\mathrm{w}, I_\mathrm{mf}) \\ \{x, y\} = \mathcal{H}(\mathrm{RoS}, I_\mathrm{u}, I_\theta, T), \end{cases} \tag{14}$$

where T is the time when the latest sensor data arrive, $\mathcal{R}$ represents Rothermel's model with cast invariants (Eq. 1) and $\mathcal{H}$ the firelet expansion (from Eqs. 5 and 6).

Depending on the available sensor data, the invariants can be turned into input data for the problem. For example, if reliable wind speed data arrive, there is no need to solve for it but instead it is directly used as input in the forward model.

2.3 Cost function

The invariants are calculated by minimising a cost function J that measures the difference between the model output and the sensor observations. The cost function proposed is the Euclidean norm summed over the different assimilation times:

$$\mathcal{J}(\boldsymbol{p}) = \sum_{t=t_i}^{t_f} \sqrt{\left[\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i(\boldsymbol{p})\right]^\mathrm{T} \mathbf{W}_i \left[\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i(\boldsymbol{p})\right]}, \tag{15}$$

where $\{\boldsymbol{x}_i\} \in \mathbb{R}^2$ are the N-coordinate set of the observed fire front position in a given time step i and $\hat{\boldsymbol{x}}_i(p) = \mathcal{M}_\mathrm{x}(\boldsymbol{p})$ are the corresponding model output positions for a set of invariants ($\boldsymbol{p}$). $\mathbf{W}_i$ is a weigh function that could be used to give more importance to particular sets of sensor data. However, in the present work no weighting function is used ($\mathbf{W}_i = \mathbb{I}$) but the framework is set to allow introductions of non-uniform weights in future work (for example to give more importance to aerial images than to in situ observations).

Equation (15) can be simplified if the $\boldsymbol{x}$–$\boldsymbol{y}$ coordinates are concatenated as one-row vector $\overline{\boldsymbol{y}}_i$ and $\tilde{\boldsymbol{y}}_i = \mathcal{M}_i(\boldsymbol{p})$:

$$\mathcal{J}(\boldsymbol{p}) = \sum_{t=t_i}^{t_f} \sqrt{\left[\overline{\boldsymbol{y}}_i - \tilde{\boldsymbol{y}}_i(\boldsymbol{p})\right]^{\mathrm{T}} \left[\overline{\boldsymbol{y}}_i - \tilde{\boldsymbol{y}}_i(\boldsymbol{p})\right]}. \tag{16}$$

Although the square root gives the correct Euclidean norm, it does not affect the minimisation and therefore was removed for the computational implementation. Each observed front ($\overline{\boldsymbol{y}}_i$) is angularly discretised between rays emanating from the origin of coordinates. The model output ($\tilde{\boldsymbol{y}}_i$) is also angularly discretised at each optimisation step, so a Lagrangian framework is used and updated for the evaluation of the cost function. No refinement is added regarding the front convexity, although this could be explored in further versions of the work to handle more complex front shapes.

2.4 Optimisation

There are two main approaches to minimise Eq. (16): gradient-free or gradient-based (Nocedal and Wright, 1999). The first group are stochastic algorithms that evaluate the cost function $\mathcal{J}(\boldsymbol{p})$ at many points to find the absolute minimum, whereas the second group use an initial guess ($\boldsymbol{p}^{\mathrm{b}}$) and follow the gradient direction towards the closest minimum. Although gradient-free algorithms can sweep a broader search space to find the absolute minimum, they have to evaluate the cost function multiple times which is computationally expensive if the forward model is slow. On the other hand, when the cost function is continuous and the possible range of values of the invariants ($\boldsymbol{p}$) is known as it is in our problem), the gradient-based algorithms are more suitable and efficient. Gradient-based algorithms can converge to a local minima instead of a global one. However, the extended sensitivity analysis performed on our problem showed that the system is benign in the sense that all the functions involved behave smoothly.

If the forward model $\mathcal{J}(\boldsymbol{p})$ is linear then the cost function is quadratic and can be minimised by solving a system of linear equations (as will be shown in the following sections). For forward models that are not linear – as is the case – the tangent linear model (TLM) is used for local linearisation (Griewank, 2000).

2.5 Tangent linear model

The TLM consists in linearising the forward model $\mathcal{M}(\boldsymbol{p})$ in the vicinity of an initial guess $\boldsymbol{p}^{\mathrm{b}}$. This linearisation can be done if the model is weakly nonlinear, as in this case. The viability of the TLM relies on the initial guess and the fact that the procedure is iterated until convergence. To calculate the TLM we use first-order Taylor series expansion around $\boldsymbol{p}^{\mathrm{b}}$.

The gradient of the linearised function is then

$$\nabla_{\boldsymbol{p}}\mathcal{J}(\boldsymbol{p}) = 2\sum_{t=t_i}^{t_f} \left[\left(\nabla_{\boldsymbol{p}}\mathcal{M}_i(\boldsymbol{p}^{\mathrm{b}})(\boldsymbol{p}-\boldsymbol{p}^{\mathrm{b}})\right)\right]^{\mathrm{T}} \tag{17}$$
$$\left[\overline{\boldsymbol{y}}_i - \left(\mathcal{M}_i(\boldsymbol{p}^{\mathrm{b}}) + \nabla_{\boldsymbol{p}}\mathcal{M}_i(\boldsymbol{p}^{\mathrm{b}})(\boldsymbol{p}-\boldsymbol{p}^{\mathrm{b}})\right)\right].$$

Applying the first-order condition for minimisation and introducing the following notation:

$$\mathbf{M}_i = \mathcal{M}_i(\boldsymbol{p}^{\mathrm{b}})$$
$$\mathbf{H}_i = \nabla_{\boldsymbol{p}}\mathcal{M}_i(\boldsymbol{p}^{\mathrm{b}})$$
$$\overline{\boldsymbol{p}}_i = (\boldsymbol{p}-\boldsymbol{p}^{\mathrm{b}})$$

gives

$$\sum_{t=t_i}^{t_f} \mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\overline{\boldsymbol{p}} = \sum_{t=t_i}^{t_f} \mathbf{H}_i^{\mathrm{T}}(\overline{\boldsymbol{y}}_i - \mathbf{M}_i), \tag{18}$$

which is a linear system that can be easily solved by using a QR factorisation with column pivoting (Nocedal and Wright, 1999).

2.6 Automatic differentiation

Calculating the Jacobian multiplication term $\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i$ in Eq. (18) requires partially differentiating the model with respect to the different invariants. This has to be done $p \times 2n \times m$ times, where p is the number of invariants used, $2n$ the coordinates of the fire front and m is the number of times that data arrives during the assimilation window..

The simplest way to numerically evaluate the Jacobian is by finite centred differences:

$$\mathbf{H}_{k,i}^{j} = \frac{\partial\mathcal{M}_i^j(\boldsymbol{p}^{\mathrm{b}})}{\partial p_k} \simeq \frac{\mathcal{M}_i^j\left(\boldsymbol{p}^{\mathrm{b}}+\epsilon_k\right) - \mathcal{M}_i(\boldsymbol{p}^{\mathrm{b}})}{||\epsilon_k||},$$

where $\epsilon_k \in \mathbb{R}^p = \{0, 0, \ldots \epsilon, \ldots 0\}$ is a small perturbation of magnitude ϵ in the position k.

But this approach has two downsides: the forward model has to be evaluated twice each time, and ϵ should be reduced as much as possible which introduces numerical truncation errors (Griewank, 2000). For these reasons, we discarded finite differences and chose an automatic differentiation approach.

Automatic differentiation allows to directly calculate the Jacobian matrix $\mathbf{H}_i$ (normally called *Tangent Linear* or *Forward*) or $\mathbf{H}_i^{\mathrm{T}}$ (called *Adjoint*). It consists of iteratively applying the chain rule of differential calculus to the programming code of the forward model and so obtain directly the code for all the partial derivatives.

Automatic differentiation is also suitable to differentiate numerical integral evaluations – as in our case – since all the statements can be split down to elemental mathematical operations.

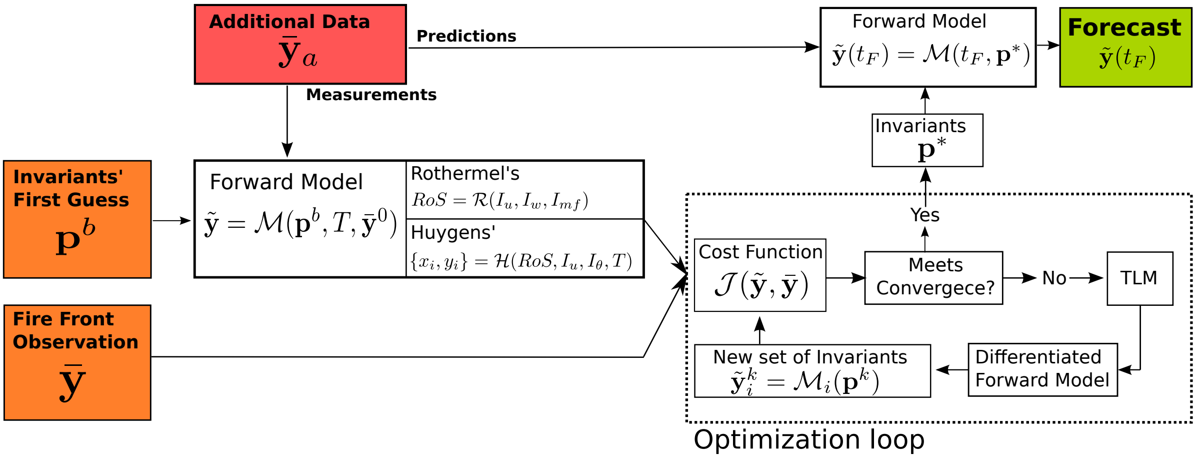
Additional Data
$\bar{\mathbf{y}}_a$
Predictions
Measurements
Forward Model
$\tilde{\mathbf{y}}(t_F) = \mathcal{M}(t_F, \mathbf{p}^*)$
Forecast
$\tilde{\mathbf{y}}(t_F)$
Invariants
$\mathbf{p}^*$
Invariants' First Guess
$\mathbf{p}^b$
Forward Model
$\tilde{\mathbf{y}} = \mathcal{M}(\mathbf{p}^b, T, \bar{\mathbf{y}}^0)$
Rothermel's
$RoS = \mathcal{R}(I_u, I_w, I_{mf})$
Huygens'
$\{x_i, y_i\} = \mathcal{H}(RoS, I_u, I_\theta, T)$
Yes
Cost Function
$\mathcal{J}(\tilde{\mathbf{y}}, \bar{\mathbf{y}})$
Meets Convergece?
No
TLM
Fire Front Observation
$\bar{\mathbf{y}}$
New set of Invariants
$\tilde{\mathbf{y}}_i^k = \mathcal{M}_i(\mathbf{p}^k)$
Differentiated Forward Model
Optimization loop

Figure 2. Program structure flow diagram. Orange boxes are the required inputs, green box is the output and red box shows additional inputs.

The tangent linear differentiation is preferable when the number of observables $2n$ (i.e. model outputs) is much larger than the number of invariants p. By contrast, calculating the adjoint differentiation is more convenient and efficient when there is a large number of invariants. Therefore, in the present work, we use tangent linear differentiation.

In future work, if the number of invariants is increased, the Adjoint Automatic Differentiation should be explored to keep the computational efficiency high and maximise the lead time.

2.7 Structure of forecasting algorithm

Figure 2 summarises the principal parts of the assimilating and inverse modelling program.

First, fire front positions $\bar{\mathbf{y}}$ are assimilated during a specific period of time (called assimilation windows). Meanwhile, an educated guess estimates the first set of invariants $\boldsymbol{p}^{\mathrm{b}}$. This first guess is based on roughly estimated data. Its influence on the model is explored in Sect. 3.1. This invariant guess is input into the forward model together with the time T of the last sensor data arrival and one known fire front position (or the initial ignition point) $\mathcal{M}(\boldsymbol{p}^{\mathrm{b}}, T, \boldsymbol{y}^0)$. The consequent first prediction set of fronts $\tilde{\mathbf{y}}_i$ is compared with the assimilated data by means of the cost function $\mathcal{J}(\tilde{\mathbf{y}} - \bar{\mathbf{y}})$ (see Eq. 16). If the the cost function is not zero, the algorithm starts the optimisation loop.

The first statement in the loop is to calculate the Jacobian terms in Eq. (16). The output is a new set of values for invariants $\boldsymbol{p}^k$ that is input to the forward model to get a new estimated set of fire fronts. If the convergence criteria are met, then the best estimated invariants vector has been found ($\boldsymbol{p}^*$) and thus the forecast is delivered by running the forward model at until the forecast time t_F. Otherwise, the loop is iterated again.

The fact that a loop is needed to estimate the invariants reduces the inaccuracy added by applying a tangent linear approach to a nonlinear model since in every new iteration the model is linearised (i.e. the differentiated forward model is run) in a new state point ($\boldsymbol{p}^{k+1}$). In addition, if any of the new invariant values in the vector $\boldsymbol{p}^{k+1}$ exceeds the physical range, its value is set back to the initial guess to prevent non-physical results.

Note that every time that the differentiated forward model is run, the forward model is also evaluated. Thus the forward model is always evaluated at the same time as the differentiated model, speeding up the algorithm and enabling the use of complex forwards models that would be prohibitive with a finite differences approach.

Regarding convergence, two criteria can be requested. The first is to state a maximum allowable error for the predictions via the cost function. The second is to state a maximum allowable change between consecutive invariant vectors. While the first criterion ensures the predictions match the observations, the second criteria might not always do so. In the following sections, both criteria are explored and compared.

2.8 Synthetic data

In order to investigate the capabilities of the forecasting algorithm, we use it with synthetic data that works as a controlled experiment before challenging it with real data. The synthetic data were generated also by a Rothermel–Huygens firelet expansion model. Fuels properties provided by Scott and Burgan (2005) were used. The synthetic data are input to the forecasting algorithm in due time mimicking the sensor data acquisition in a real wildfire.

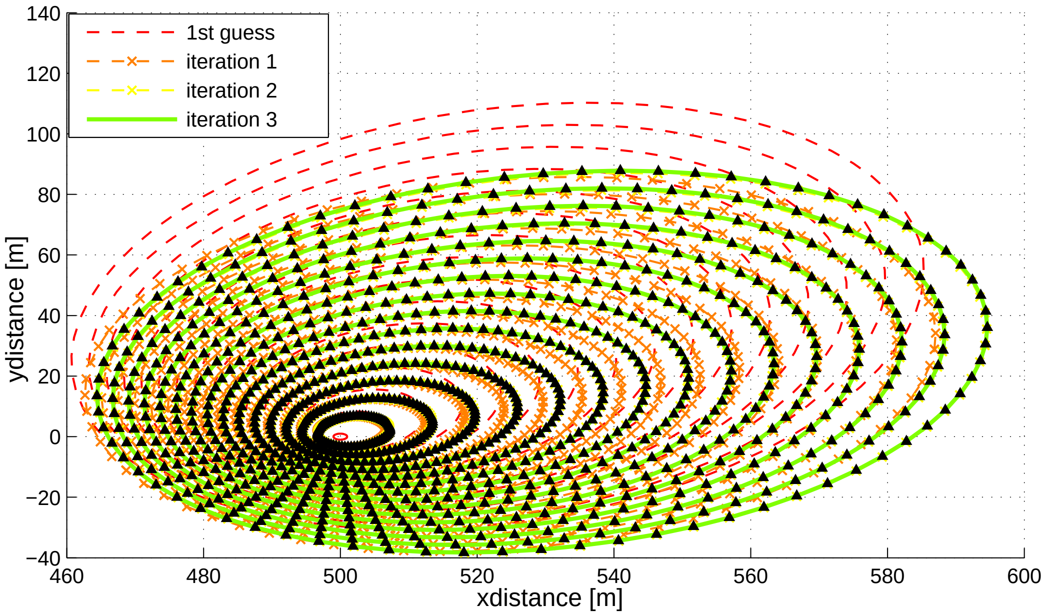

Figure 3. Guess, observation and iterations of fire fronts in an x–y plane (plan view of a wildfire). The black triangles are the 15 observed perimeters. The red dashed lines are the fire fronts generated with the first guess and the dashed lines are the following iterations. The last iteration is the green solid lines.

3 Results

The performance of the forecasting algorithm is investigated in different situations where synthetic data simulate the observations to assimilate. The tests are performed for different values of parameters like the assimilation window, assimilated data (fire fronts locations and feeding frequency) and initial guess. We look at several features like convergence of the invariants, minimisation of the cost function, effect of the initial guess, effect of the assimilating window width, the computing time and the leading times obtained.

The same methodology is also applied with alternative invariants to handle situations where some of the quantities assumed as constant are allowed to vary.

In all of the following tests, punctual ignition source is considered as the initial integration point for the fire front expansion. This ignition point source is depicted as a red spot in all the plots and is a required piece of information to run the forecasting algorithm. In a real wildfire situation, it could be identified as the first reported location of the fire. If the fire has spread out before the first bit of information arrives and it is no longer a point source, the first assimilated fire front can be also used as a virtual ignition perimeter by considering the whole fire front as a set of initial ignition sources.

3.1 Initial guess

The forecasting algorithm needs an initial guess of the invariant value where the first tangent linear approximation (TLM) is performed. This first educated guess can be directly generated within the range of validity of each invariant – without considering any hint from the actual wildfire – or by using Rothermel equivalent equations (Eqs. 2–4 and 13) and estimating the six physical underlying quantities δ, M_f, M_x, σ, W_0 and θ which can be roughly done by observing the fuel and wind.

The six initialising quantities were studied over the range of values found according to operational-based considerations. For instance, the fuel depth δ can be easily distinguished to be between 5 cm pine needle litter or 1 m for tall grass. Its offset of the initial guess is lower than 1.50 m. In contrast, some other variables such as moisture content (M_mf) or ovendry fuel loading (w_o), cannot be estimated with such easy and therefore the possible offset is much larger.

3.2 Quantifying the invariants

The first scenario investigated here assimilates 15 fire fronts during a window width of 15 min (i.e. data of the fire front position arrives once every minute). The invariants converge within 3 iterations (i.e. three runs of TLM). Figure 3 shows the observed data, the fronts generated with the initial guess of invariants and the respective fire fronts after each iteration until convergence is reached. The invariants and cost function convergence are shown in Fig. 4. The cost function shows

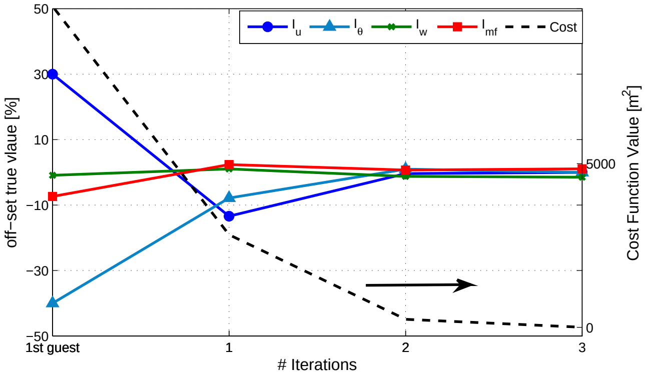

Figure 4. Convergence of cost function (dashed line, right axis) and individual convergence of each invariant to the true value (solid lines, left axis) as a percentage difference. Assimilation windows = 15 min (1 assimilation min^{-1}).

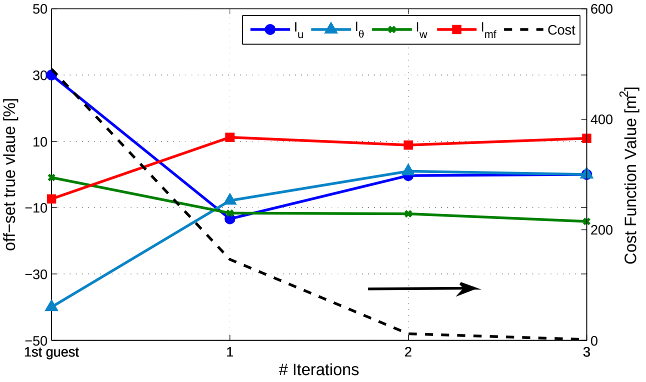

Figure 5. Convergence of cost function (dashed line, right axis) and individual convergence of each invariant to true value (solid line, left axis). Assimilation windows = 3 min (1 assimilation min^{-1}).

a rapid decrease towards zero. Its slope quantifies the convergence rate. At the first iteration the slope is steep which indicates that the algorithm quickly corrects the large discrepancies. As the cost decreases so does the slope, indicating that convergence is achieved. Fig. 4 also shows that all invariants converge to the true values with 2 % of difference.

3.3 Invariant Multiplicity

The window width (WW) is the length of time during which the forecasting algorithm is being fed data (i.e. fire front location in the case at hand). The time between consecutive fire front observations is called assimilation period (ΔT) and can be directly related to the assimilating frequency ($F = 1/\Delta T$).

The main effect of the number of assimilated fronts (WW/ΔT) is resolving the problem of invariant multiplicity (or interdependence). Multiplicity is when different values of two or more invariants lead to the same prediction of the fire perimeter. The value of the cost function tends to increase as the assimilation window increases and more fronts are assimilated. The error of the initial guess amplifies with the propagation (the previous fire front position is required to calculate the new one) and therefore the forecasting algorithm is more sensitive to the wrong identification of invariants. This is shown in Fig. 5 where instead of assimilating 15 min (and 15 fire fronts) – as in the converging example Fig. 4 – we assimilate front positions during 3 min (i.e. three front positions). The cost function rapidly drops to zero but in this case the value estimated for both I_{mf} and I_{w} differs from the true value by 10 %. The reason is that now the initial cost function has a lower absolute value since the propagation of an inaccurate estimation is truncated in time and therefore the effects of an incorrect assimilation are hidden. It is worth mentioning that despite the possibility of I_{mf} and I_{w} misconverging, RoS is always correctly estimated as it has no multiplicity in the forward model and only one value can fit the observations.

One way to deal with multiplicity is by defining only one invariant for the RoS. This approach, however, does not allow for the forecasting algorithm to be ameliorated if extra data become available (as will be done in Sect. 3.5) since no information about particular contributions is achieved. Thus, a more interesting way to diminish multiplicity is to recast the invariants and input extra data in a way that they become functionally independent. For instance, if the fuel–moisture invariant is multiplied by a measurable quantity (such as fuel depth or moisture content) that varies spatially or over time, then its value is no longer exchangeable with the wind factor. The same strategy could be used for the wind invariant if wind speed is known. This approach is successfully explored in the following sections.

The third way to deal with multiplicity is by assimilating additional quantities that are predicted by the forward model. It is worth pointing out the difference between inputting additional values and assimilating more data. The first consists of extra inputs to run the forward model and allows it to handle more complex situations. Examples of this could be information of moisture content, fuel properties or wind speed. More data assimilation, in contrast, requires more outputs from the forward model. Thus, in our case, only the positions of the fronts are assimilated but the forward model can be complemented so it delivers additional characteristics such as flame height or fire intensity. By assimilating this additional data the invariant multiplicity is reduced since each invariant is then part of different equations and they are no longer dependent.

3.4 Positive lead times

In order for it to be an operative tool, the forecasting algorithm must deliver the forecast ahead of the event, thus any forecast must meet the positive lead time requirement. The

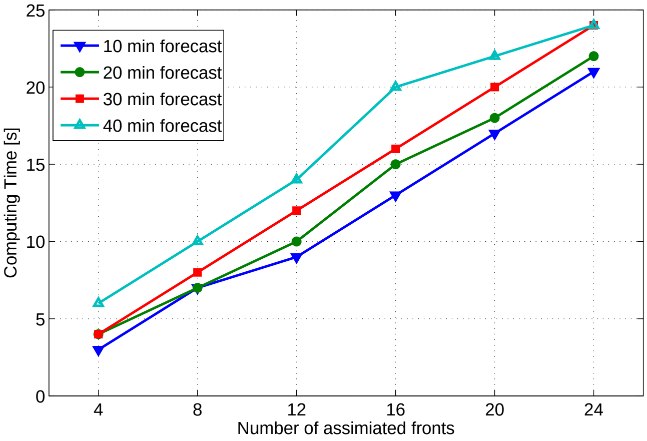

Figure 6. Computing time required for four different forecasting time lengths (10, 20, 30 and 40 min) versus the number of assimilated fire fronts.

lead time is defined as the amount of time between the delivery of the forecast and the successful predicted event. If the forecasting algorithm needs 25 s of computing time to deliver a 10 min forecast, then the lead time is 9.6 min. As shown in Fig. 6, the model is so fast (in the range from 2 to 25 s) that it delivers always a positive lead time in the order of dozens of minutes for the case of synthetic data.

The lead time principally depends on the number of assimilated fronts and the initial guess (i.e. iterations required for convergence). The forecasting time length t_F(either we ask for a 10 min or 40 min forecast) also plays a role when the forward model is computationally demanding. However, due to the synthetic data scenario used in the case at hand, its contribution is limited as shown in Fig. 6.

3.5 Different data contexts

The invariants can be adapted to different data situations. To show the versatility of our model two different cases with different available data are presented as example.

In the first case, wind speed and direction are provided and assumed to be uniform – same wind speed and direction for all the fire perimeter – although they can vary in time. By contrast, in the second case, the fuel depth δ is provided as as sensor data and is allowed to vary spatially. Wind speed and direction can be gathered from deployed units as well as from weather stations. Regarding the information about fuel, forest managers usually map forest areas in advance to list their spatially distributed characteristics. New techniques recently brought into the field such as the use of lidar – light detection and ranging (Mutlu et al., 2008), potentially increases the accuracy and availability of this information and opens the door for preparing operative measuring systems for the situations when these data are not known.

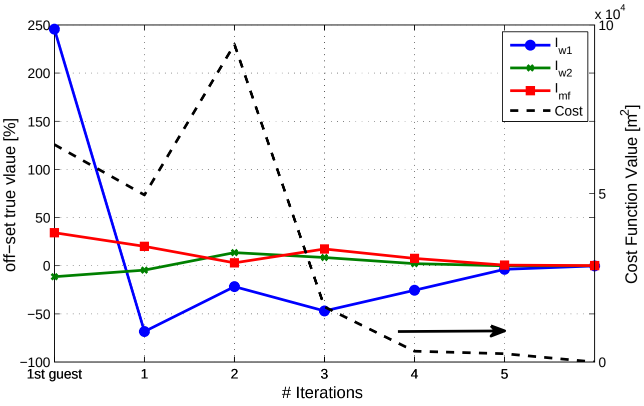

Figure 7. Convergence of the cost function and the invariants when wind speed and direction are used as an input. The peak in the third iteration of the cost function is due to the correcting algorithm that resets negative values.

3.5.1 Wind speed as sensor data

The first step is to recast the invariants related to wind speed and wind direction by reversing I_u and I_θ into known parameters. Then I_w is redefined using the wind factor from Rothermel:

$$\Phi_w = CU^B\left(\frac{\beta}{\beta_0}\right)^{-E} = \mathcal{P}(\sigma,\beta,w_o,\delta)\cdot U^B = I_{w_1}\cdot U^{I_{w_2}}. \quad (19)$$

Thus,

$$I_{w_1} = \mathcal{P}(\sigma,w_o,\delta) = C\left(\frac{\beta}{\beta_0}\right)^{-E} \quad (20)$$

$$I_{w_2} = \mathcal{F}(\sigma) = B, \quad (21)$$

where C and B are calculated with experimental correlations derived by Rothermel and β, β_0 are the nominal and the optimal packing ratio respectively.

The other invariant I_{mf} remains the same and, thus, the model is described by three invariants plus the simulation time T:

$$\mathcal{M}(I_{w_1},I_{w_2},I_{\mathrm{mf}},T) = \begin{cases}\mathrm{RoS} = \mathcal{R}(I_{w_1},I_{w_2},I_{\mathrm{mf}})\\ \{x,y\} = \mathcal{H}(\mathrm{RoS},I_u,I_\theta,T).\end{cases} \quad (22)$$

The reason why three invariants are needed despite the new sensor data is because the effect of the wind in the RoS and the firelets shape depends on fuel parameters such as the packing ratio or ovendry bulk density. However, the important difference is that now the wind changes in time but is known (it is not an invariant any more) and, therefore, the forecasting algorithm can deal with more complicated – less idealised – situations.

Despite this recast being, to some extent, more complicated than the previous one, it makes it possible to identify the invariants more accurately than the previous recast. Nevertheless, on average, more iterations are needed to reach the

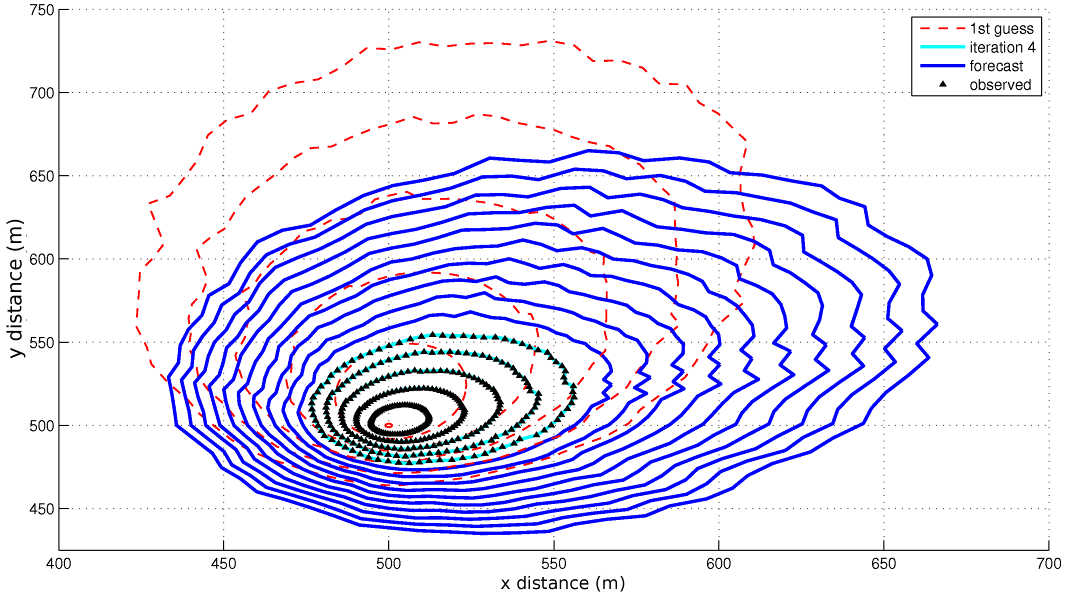

Figure 8. Five assimilated fire fronts with 1 min intervals (black solid lines). The first guess (red dashed line) is taken to be far from the true invariants vector to check the algorithm capability to converge. A 10 min forecast (blue solid lines) is also calculated using fuel depth as sensor data.

required convergence which slightly increases the computing time.

Besides considering observed values for wind speed and direction, the forecast algorithm can also consider meteorological predictions to deliver a more accurate forecast when these quantities vary. To illustrate this, five fire fronts are assimilated during 25 min (at a frequency of 1 fire front every 5 min). The invariants are perfectly identified with six iterations as shown in Fig. 7. Then, a forecast is launched for the next 25 min with a synthetic prediction of wind speed and direction.

3.5.2 Fuel depth as sensor data

We consider now the case where fuel depth is available and varies spatially but is constant. To cast the new invariants we use the information obtained with a sensitivity analysis performed on Rothermel's model. The analysis reveals that RoS is linearly related to fuel depth δ as a first approximation. Thus, the RoS can now be written as

$$\mathrm{RoS} = I_{\mathrm{mfw}} \cdot \delta(x, y), \tag{23}$$

where fuel depth $\delta(x, y)$ varies spatially.

The wind contribution is now included in $\mathrm{RoS} = I_{\mathrm{mfw}}$ and therefore we have to create a new parameter that accounts for the shape of the elliptical firelets (i.e. the eccentricity): I_{LB}, where LB stands for length-to-breadth ratio. This invariant also depends on wind speed and, thus, is not independent of I_{mfw}. This does not affect the capacity of our forecasting model since I_{LB} could be interpreted as a shaping factor and the way it is used in the forward model (only in the Huygens expansion part) prevents it from being mixed with I_{mfw}. As in the previous cases the wind direction invariant I_θ is required to close the invariant cast.

The effect of assimilating a space-dependent variable is that RoS now also depends on the location. This adds an extra non-linear behaviour to the model, since now when the fire front location changes, the RoS changes as well. Despite this higher complexity, our algorithm handles it in the optimisation loop and correctly matches the observations (Fig. 8) and identifies the invariants (Fig. 9).

3.6 Lead time

The lead time for the different implementations discussed above is investigated by assimilating different number of fire fronts and recording the computing time to deliver a 30 min forecast. The total assimilating time since it depends on the assimilation frequency (i.e. the number of assimilations per unit of time). Changing this frequency has a minor influence on the computing time since its contribution is linear in our forward model but might be important if more complex forward models (such as Computational Fluid Mechanics based, for example). The Rothermel variables that generate the synthetic data and the educated guess were kept constant for all the scenarios when they were not sensor data (such as wind speed, wind direction or fuel depth).

Figure 10 depicts the computing time versus the number of assimilated fronts. The invariant cast for the situation when wind speed and direction are known sensor data turns out to be the faster case. As expected, decreasing the number of invariants to be identified, speeds up the model since the

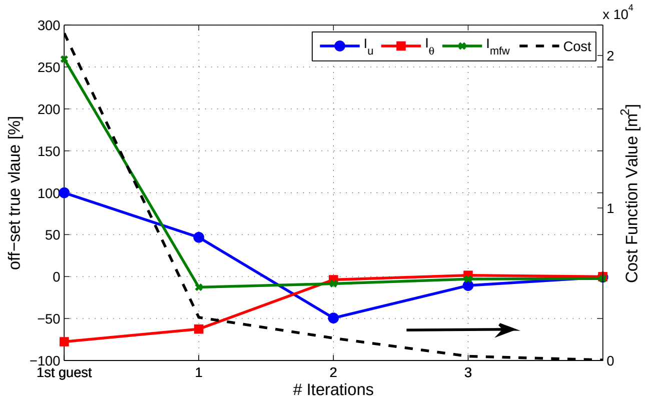

Figure 9. Cost function and invariant convergence when fuel depth is sensor data.

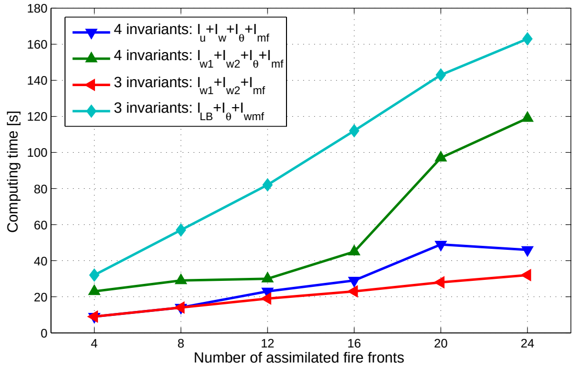

Figure 10. Computing time for all the implementations of the forecasting algorithm studied.

dimension of the matrices involved in the optimisation process decreases. The exception is when fuel information is data. The spatial dependency of the fuel depth and the fact that RoS has to be recalculated in every node raises the computing time, and thus this case is the slower one. The effect of feeding the algorithm with wind speed data becomes noticeable above 16 assimilated fronts when the complexity of the fire front shapes increases the number of iterations required to reach convergence.

Despite these significant differences, when eight fronts are assimilated the forecast is delivered in less than 1 min and even when 24 fronts are assimilated the lead time is well above 25 min for a 30 min forecast.

A laptop with dual processor core of 2.2 GHz is used as a computational tool since (as stated in the initial requirements) the forecasting algorithm must be suitable for desktop computers.

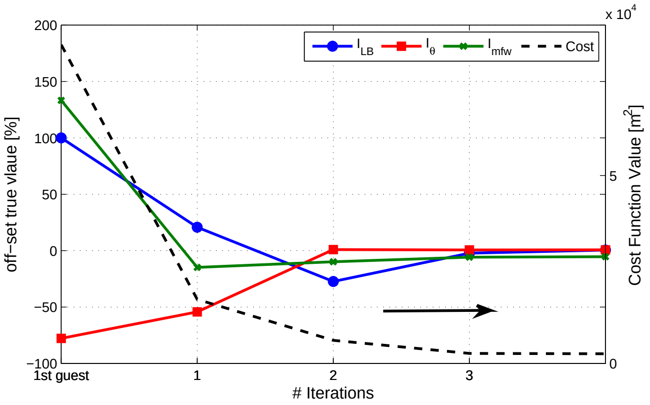

Figure 11. Convergence of cost function (black dashed line) and invariants (solid lines) when perturbed data are assimilated.

3.7 Effect of errors in the data

The fact that the synthetic data are generated with a Rothermel–Huygens model implies that there exists at least one true invariant vector that exactly generates the observed fronts. However, this is not the case in reality since the forward model used is only an approximation of the real fire dynamics. Thus, to test the forecasting algorithm in a situation where such a true vector no longer exists (thus, perfect convergence is then impossible), the synthetic data used in the fuel depth sensor data case (Sect. 3.5.2) have been randomly perturbed with an error uniformly distributed in the range of [0 ± 10] m. Apart from exploring the response of the forecasting algorithm in a case where the forward model cannot properly describe the fire locations, this test can be seen as a sensitivity check of the errors and accuracy involved in data acquisition.

As expected, the best optimisation does not match the observations perfectly and, thus, the cost function converges to a value of 2500 m^2 instead of zero (see Fig. 11). Despite this fact, the convergence of the invariants – towards true values – is still reached with an error lower than 5 %.

Figure 12 shows the observed fronts and the corresponding optimisation after four iterations. The sharp corners in the observed perimeters are due to the random distribution of the fuel depth and the added error. More tests performed while extending the error added to observation demonstrate that the algorithm manages to assimilate the observations with perturbations of up to ±20 m in magnitude. The invariants also converged in this case which demonstrates the potential of this forecasting algorithm even when inaccurate data are available.

4 Conclusions

A simple but powerful methodology to forecast wildfire dynamics based on data assimilation is implemented and

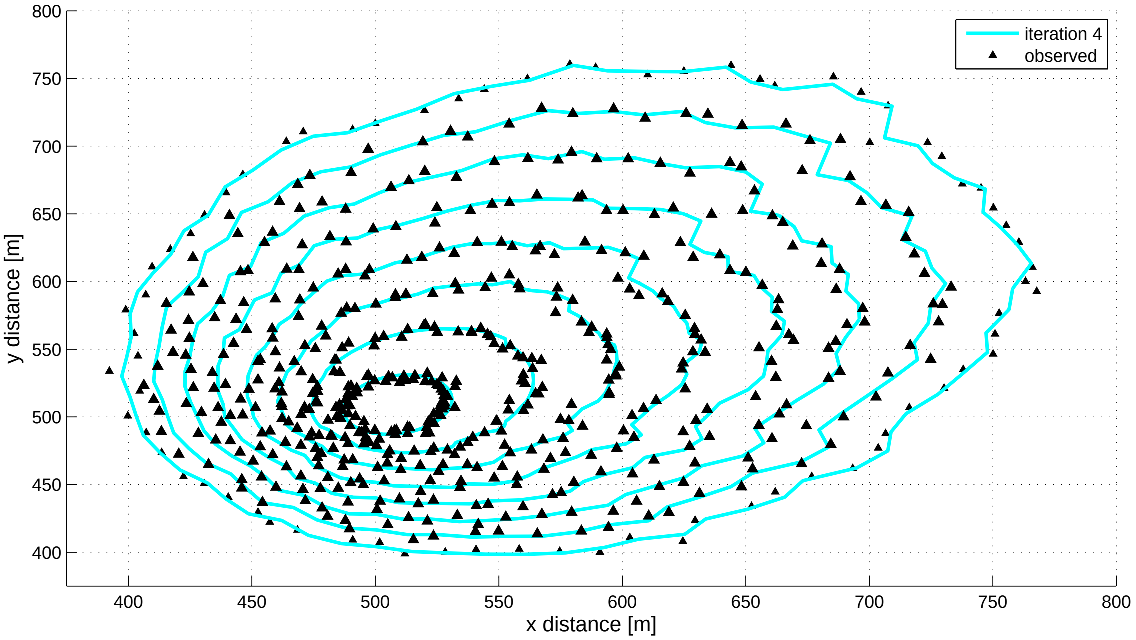

Figure 12. Perturbed fire fronts (black lines) correctly assimilated after four integrations. The invariant cast used is this of Sect. 3.5.2 where fuel depth is used as an input and three invariants are identified.

explored focusing on wind-driven wildfires. The algorithm framework is general enough to be valid to different sensor data or forward models. In this work the forward model is composed by use of Rothermel's spread theory and Huygens expansion and is challenged with synthetically generated front locations. The forecasting algorithm uses direct automatic differentiation and a tangent linearisation of the forward model to solve the optimisation problem. This strategy showed great efficiency finding the invariants within less than 10 iterations (runs of TLM), although special attention must be taken regarding multiplicity in the determination of the invariants. Multiplicity can be avoided by extending the forward model so that it predicts extra parameters (such as flame height or heat release rate) and assimilates them, or including extra information about the system to break the multiplicity. The latter was implemented and illustrated in two different scenarios. All the invariants were then correctly identified, even when the first guess greatly differed from the true value. All the implementations had a positive lead time (time ahead of the event). The most computationally expensive implementation is the one that uses fuel depth since the RoS varies in each node of the front.

Future work should study real sensor data (e.g. Coen et al., 2013) and look into improved (more accurate yet faster) models. To keep developing the methodology some identified limitations should be tackled as spotting fires – which do not follow the classical (i.e. Rothermel) fire spread – and the capacity of the forecasting algorithm to deal with uncertainties caused by the lack of reliable data, and deliver probabilistic values as outputs. To pursue this, we propose to increase the number of invariants to several dozen. Then the automatic direct differentiation should be switched to adjoint differentiation (adjoint modelling approach) to keep the low computational cost requirement.

A discussion on whether data assimilation should involve more complex models or not might be spurious at this point. This will be decided by the International wildfire community at large, specially the Fire Service. If the development of weather forecasting systems over the last few decades can serve as guidance somehow for wildfire forecasting systems, we note that they currently simulate weather patterns in a series of models of diverse complexity of which the grids range from fine and regional to global and coarse. So far, we can show that our wildfire forecast method is light, fast and flexible. It can be adapted to run on models of any complexity, and we show its strengths here using synthetic data and a model that, albeit rather simple, is the most widely used model by the international wildfire community.

Acknowledgements. Support for O. Rios from the Erasmus Mundus European Project and the International Master of Science in Fire Safety Engineering (IMFSE) is gratefully acknowledged. The authors also want to thank Elsa Pastor for comments that improved draft versions.

Edited by: D. Veynante
Reviewed by: A. Simeoni and two anonymous referees

References

Anderson, H. E.: Predicting wind-driven wild land fire size and shape, US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, 1983.

Coen, J. and Schroeder, W.: Use of spatially refined satellite remote sensing fire detection data to initialize and evaluate coupled weather-wildfire growth model simulations, Geophys. Res. Lett., 40, 5536–5541, 2013.

Cowlard, A., Jahn, W., Abecassis-Empis, C., Rein, G., and Torero, J. L.: Sensor assisted fire fighting, Fire Technol., 46, 719–741, 2010.

Finney, M.: FARSITE, fire area simulator – model development and evaluation, vol. 3, US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.

Griewank, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, no. 19 in Frontiers in Appl. Math., SIAM, Philadelphia, PA, 2000.

Jahn, W., Rein, G., and Torero, J. L.: Forecasting fire growth using an inverse zone modelling approach, Fire Safety J., 46, 81–88, 2011.

Jahn, W., Rein, G., and Torero, J. L.: Forecasting fire dynamics using inverse computational fluid dynamics and tangent linearisation, Adv. Eng. Softw., 47, 114–126, 2012.

Lautenberger, C.: Wildland fire modeling with an Eulerian level set method and automated calibration,. Fire Safety J., 62, 289–298,2013.

Mandel, J., Bennethum, L. S., Beezley, J. D., Coen, J. L., Douglas, C. C., Kim, M., and Vodacek, A.: A wildland fire model with data assimilation, Math. Comput. Simulat., 79, 584–606, 2008.

Mandel, J., Beezley, J. D., Coen, J. L., and Kim, M.: Data assimilation for wildland fires, IEEE Contr. Syst. Mag., 29, 47–65, 2009.

Mutlu, M., Popescu, S. C., Stripling, C., and Spencer, T.: Mapping surface fuel models using lidar and multispectral data fusion for fire behavior, Remote Sens. Environ., 112, 274–285, 2008.

Nocedal, J. and Wright, S. J.: Numerical Optimization, Springer Series in Operations Research and Financial Engineering, Springer, New York, 1999.

Richards, G. D.: An elliptical growth model of forest fire fronts and its numerical solution, Int. J. Numer. Meth. Eng., 30, 1163–1179, 1990.

Richards, G. D.: The properties of elliptical wildfire growth for time dependent fuel and meteorological conditions, Combust. Sci. Technol., 95, 357–383, 1993.

Rochoux, M. C., Delmotte, B., Cuenot, B., Ricci, S., and Trouvé, A.: Regional-scale simulations of wildland fire spread informed by real-time flame front observations, P. Combust. Inst., 34, 2641–2647, 2013.

Rochoux, M. C., Emery, C., Ricci, S., Cuenot, B., and Trouvé, A.: Towards predictive simulation of wildfire spread at regional scale, IAFSS 11th Symposium, 2014.

Rossi, L., Molinier, T., Akhloufi, M., Pieri, A., and Tison, Y.: Advanced stereovision system for fire spreading study, Fire Safety J., 60, 64–72, 2013.

Rothermel, R.: A mathematical model for predicting fire spread in wildland fuels, Intermountain Forest & Range Experiment Station, Forest Service, US Department of Agriculture, 1972.

Scott, J. and Burgan, R.: Standard fire behavior fuel models : a comprehensive set for use with Rothermel's surface fire spread model, USDA Forest Service, Rocky Mountain Research Station, General Technical Report RMRS-GTR-153, 72 pp., 2005.

Sullivan, A. L.: Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models, Int. J. Wildland Fire, 18, 387–403, 2009.