An Interactive Data-Driven HPC System for Forecasting Weather, Wildland Fire, and Smoke

Jan Mandel
Mathematical and Statistical Sciences
University of Colorado Denver
Denver, CO, USA
jan.mandel@ucdenver.edu

Martin Vejmelka
Machine Learning Research
CEAI, Inc.
Prague, Czech Republic
vejmelkam@gmail.com

Adam K. Kochanski
Atmospheric Sciences
University of Utah
Salt Lake City, UT, USA
adam.kochanski@utah.edu

Angel Farguell
Mathematical and Statistical Sciences
University of Colorado Denver
Denver, CO, USA
angel.farguell@gmail.com

James D. Haley
Mathematical and Statistical Sciences
University of Colorado Denver
Denver, CO, USA
james.haley@ucdenver.edu

Derek V. Mallia
Atmospheric Sciences
University of Utah
Salt Lake City, UT, USA
derek.mallia@utah.edu

Kyle Hilburn
Cooperative Institute for Research in the Atmosphere
Colorado State University
Fort Collins, CO, USA
kyle.hilburn@colostate.edu

***Abstract*—We present an interactive HPC framework for coupled fire and weather simulations. The system is suitable for urgent simulations and forecast of wildfire propagation and smoke. It does not require expert knowledge to set up and run the forecasts.**

The core of the system is a coupled weather, wildland fire, fuel moisture, and smoke model, running in an interactive workflow and data management system. The system automates job setup, data acquisition, preprocessing, and simulation on an HPC cluster. It provides animated visualization of the results on a dedicated mapping portal in the cloud, and as GIS files or Google Earth KML files. The system also serves as an extensible framework for further research, including data assimilation and applications of machine learning to initialize the simulations from satellite data.

***Index Terms*—WRF-SFIRE, coupled atmosphere-fire model, MODIS, VIIRS, satellite data, fire arrival time, data assimilation, machine learning**

I. Introduction

The United States has entered a new era of increasing wildfire frequency and intensity, which has culminated in a number of devastating wildfire seasons over the past decade. The landscape has become more fire-prone due to climate change and urban development, which has resulted in steeply rising fire-suppression costs. On the other hand, fire is a part of the natural environment, and fire prevention practices can often lead to excessive fuel accumulation and catastrophic fires that are difficult to manage. There is a significant need for management decisions that are based on a multifaceted analysis of risks and benefits associated with wildfires and prescribed burns. As a result, new advanced decision support tools that integrate satellite/aerial remote sensing with a coupled fire, weather, fuel and smoke modeling framework are needed.

Configuring a real weather-fire simulation requires significant expertise. The forecast system described here is a distillation of such expertise into a system that is relatively easy to use. The authors found this forecast system to be the most convenient way for setting up fire simulations for research applications, instead of manually configuring model simulations.

This communication is primarily concerned with the HPC work flow and with the aspects of the models and data which affect the ability to provide a forecast. See [1]–[7] for more information on the modeling and data assimilation methodologies employed.

The rest of this paper is organized as follows. In Sections II and III, we provide a brief overview of the physics models employed. Section IV contains a description of the overall HPC workflow. Section V deals with data acquisition. The next two sections outline two breaking extensions of the system and demonstrate its applicability for advancing research purposes: ignition from satellite fire detection data by machine learning in Section VI, and assimilation of the satellite data in Section VII. Case studies showing the capability of the system to deliver forecasts are presented in Section VIII, and Section IX is the conclusion.

Partially supported by NSF grants DMS-1216481 and ICER-1664175, and NASA grants NNX13AH59G and 80NSSC19K1091. Computing resources were provided by the University of Utah Center for High Performance Computing (CHPC), NCAR Cheyenne (doi:10.5065/D6RX99HX) at NCAR CISL, sponsored by the NSF, and the Center for Computational Mathematics at the University of Colorado Denver.

978-1-7281-5995-9/19/$31.00 ©2019 IEEE
DOI 10.1109/UrgentHPC49580.2019.00010

II. Coupled Fire-Atmosphere Model

The core of the system is WRF-SFIRE, which couples a high-resolution multi-scale atmospheric model, NCAR's Weather Research and Forecasting (WRF) model [8], with a spread fire model (SFIRE). WRF-SFIRE was developed at the University of Colorado Denver and NCAR [2]. WRF-SFIRE became part of the WRF release in 2011 as WRF-Fire [3], [9], which was recently selected by NCAR as the foundation of the operational Colorado Fire Prediction System (CO-FPS), and is currently under further development at NCAR [10], [11]. WRF-SFIRE was further developed by coupling with smoke [1], a fuel moisture model [4], [5], and spin-up of the atmosphere model for perimeter ignition and data assimilation [7], [12]. An older iteration of WRF-SFIRE also serves as the foundation of the Israel national wildfire simulation and danger system MATASH [13]. See [14] for a survey of wildland fire models.

WRF-SFIRE has grown out of [15] by replacing the atmosphere model with WRF and its HPC infrastructure, and by using a level set method to trace fire progression. The fire rate of spread is parameterized based on the local wind speed, topography and fuel characteristics at the fireline, and also coupled with a prognostic dead fuel moisture model. Smoke propagation is modeled by WRF tracers, which are transported by the wind field. The tracers are emitted at a rate proportional to fuel consumption and inserted into the lowest layer of the WRF grid. Through the integration of a fuel moisture module that is driven by surface atmospheric conditions, this model can resolve the diurnal and spatial variability of fuel moisture, and its impact on the fire behavior, plume rises, and downwind smoke dispersion. As a result, the predicted fire behavior and plume dynamics are linked to the atmosphere through winds and fuel moisture, which are in turn locally affected by the fire. WRF-SFIRE is designed to simulate the landscape-scale physics of the coupled fire-atmosphere phenomena and focuses on the importance of rapidly changing meteorological conditions at the fire line by accounting for local feedbacks between the fire, fuel, terrain, smoke [16], and atmospheric boundary layer evolution. This modeling framework is driven by operational numerical weather prediction products to generate downscaled weather forecasts at a resolution of several hundred meters, which enables capturing fire-atmosphere interactions as well small-scale meteorological phenomena such as thunderstorms, micro-bursts, and topographical flows. WRF-SFIRE is a community, open source software,[1] which benefits from being built upon the community WRF code and the expertise of an international team of atmospheric modelers contributing to WRF.

III. Fuel Moisture Model

One of the unique aspects of WRF-SFIRE is its integration with the fuel moisture model. The fuel moisture code uses air temperature, relative humidity, and precipitation to estimate the equilibrium fuel moisture contents, and then runs a time-lag differential equation to trace the time evolution of the moisture content in four fuel time-lag classes (1 h, 10 h, 100 h, and 1000 h). Spatial maps of fuel moisture in these classes correspond to the fuel elements that have diameters that range from less than 1.81 mm (1 h) to 20.3 cm (1000 h). The differential equation is run at each node of the atmospheric mesh (hundreds of meters resolution) on the surface, so the computational cost compared to the cost of the atmospheric model is negligible. The fuel moisture contents in all classes are then interpolated to the finer fire mesh, and combined according to the proportion of the fuel moisture classes for each fuel type. The integrated contribution from these classes provide the overall fuel moisture at the fire model resolution (tens of meters), which is used in the computation of the fire progression.

In addition to the fuel moisture model integrated in WRF-SFIRE, a standalone implementation of the fuel moisture model with assimilation of dead fuel moisture observations from remote automated weather stations (RAWS) was developed [5]. The system uses a trend surface model (TSM), which accounts for the effects of topography and meteorological conditions on the spatial variability of fuel moisture. At each grid point, the TSM provides a pseudo-observation, which is assimilated via Kalman filtering into the fuel moisture model state and parameters. The standalone fuel moisture model with RAWS data assimilation is used for fuel moisture nowcasting. The output from this system (updated hourly) is processed through the visualization pipeline (Section IV-C below), as well as used to initialize the fuel moisture model and its parameters in WRF-SFIRE.

IV. Real-Time HPC Workflow

A flowchart of the entire system is in Fig. 1. The system consists of three main components: graphical front end `wrfxctrl`[2], HPC simulation and management system `wrfxpy`[3], written in Python, and web-based visualization `wrfxweb`[4] in JavaScript. There are no services running, except for the inevitable web servers. Jobs are independent shell scripts calling Python and running as background processes.

This design and the initial implementation were developed by Martin Vejmelka. They resulted in a transparent and extensible architecture, with a reasonable learning curve for new developers.

A. Initialization

The fire simulations are initialized and their execution monitored from the graphical front end control system `wrfxctrl`. The front end allows a user to click on a map to ignite a simulated fire anywhere in the Continental United States (CONUS), define ignition time, forecast length as well as to select a simulation profile with a predefined horizontal resolution, domain sizes, and type of meteorological forcing

[1] https://github.com/openwfm/wrf-fire

[2] https://github.com/openwfm/wrfxctrl

[3] https://github.com/openwfm/wrfxpy

[4] https://github.com/openwfm/wrfxweb

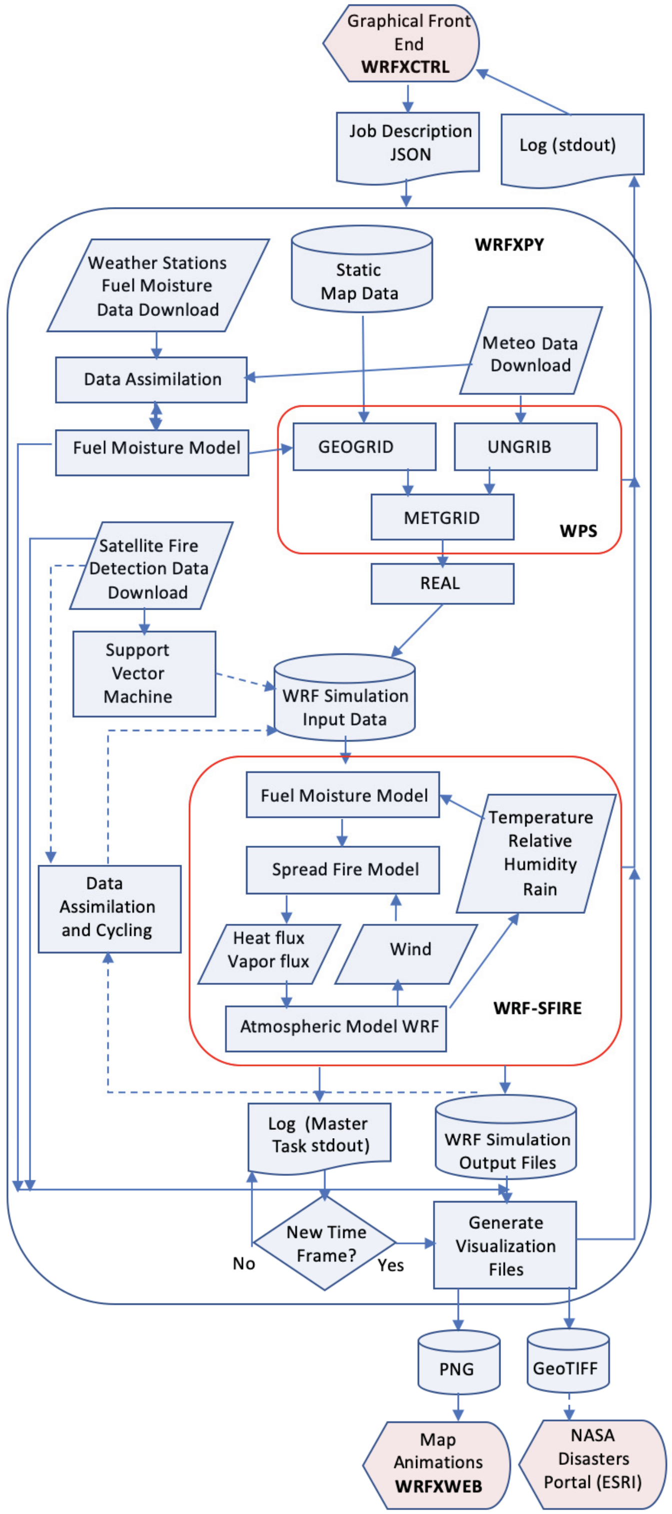

Fig. 1. Overall system flowchart. Dashed connectors are research pathways which are functional, but their use is currently not automated.

TABLE I
SIMULATION PROFILES

Parameter	Profile			
	1	*2*	*3*	*4*
Number of grid points in the innermost domain ($X \times Y$)	166 x 166	97 x 97	70 x 70	61 x 61
Resolution of the innermost atmospheric mesh	555 m	555 m	333 m	333 m
Fire mesh resolution (m)	28 m	28 m	33 m	33 m
Meteorological forcing	NAM218	NAM218	NAM227	HRRR
Max. forecast length	84 h	84 h	60 h	36 h

used to initialize the weather model. Examples of typically used simulation profiles are presented in Table I.

Then, the job description is created and encoded as a JSON file with all needed initialization parameters, data sources, HPC resources requested, and a selection of visualization products. The front-end starts the simulation as a background script `forecast.sh` in the `wrfxpy` system with the JSON job description file as command line argument. For experiments, the JSON input file can be modified and the forecast script run by hand. The front end monitors the progress of jobs graphically as well as by a scrolling the text log. The graphical front end is implemented as a single-user web server, running as a local application on a front end node of HPC cluster. Since the functionality of the front end controls is implemented by inquiring files, it can be restarted and continued without any adversary effects.

After a job is started by running `wrfxpy/forecast.sh`, it completes job parameters with defaults, creates the job directory, and creates a JSON file with the initial job state and various control text files, called namelists and tables in WRF, which contain description of the simulation domains and simulation parameters.

B. Preprocessing

The WRF Preprocessing System (WPS) [17] consists of three programs, GEOGRID, UNGRIB, and METGRID, and it is followed by REAL, which is distributed with WRF rather than WPS. In the preprocessing step, the system first starts two or more processes in the background, and waits for their completion:

1) GEOGRID defines the simulation domains and interpolates static data to the grids. GEOGRID is run on standard static data, which come with WRF, plus GeoTIFF files with fine-scate topography and fuel information (Section V-A below). WRF-SFIRE contains an extended version of GEOGRID with GeoTIFF support [18] for this purpose. If the job specifies the use of the output of the fuel moisture model with RAWS data assimilation, its state and parameters become one more dataset to be processed by GEOGRID.

2) Download atmospheric data (Section V-B below) for the simulation period and process them by the UNGRIB utility. UNGRIB processes meteorological data, which comes in GRIB format, to an intermediate format. There may be more than one such process; e.g., the CFSR product comes as two collections of files, each containing different 3D atmospheric fields, and the two collections of files are downloaded and processed in parallel.

The outputs of GEOGRID and UNGRIB are then processed by METGRID, which interpolates the meteorological data horizontally to the model grid, and then by REAL to create vertical levels and finalize the WRF input files.

The prescribed fire arrival time obtained from ignition by machine learning (Section VI below) or data assimilation (Section VII below) can be written into the WRF input files at this point.

C. Running the Simulation on HPC Cluster

Once the WRF input files are ready, the forecast script submits an MPI job to the queuing system on the cluster and waits until the log files appear, and then waits for a line in the log that indicates that WRF wrote a frame to the output files. If the frame is not found in the output files, which may happen on a distributed system, where a head node may not be presented with a timely image of the filesystem, the script will wait and keep trying. The system also inquires the cluster queuing system for the status of the MPI job and updates the job JSON status file, otherwise, if the job does not complete successfully, the system might wait for its output forever.

When the new time frame is found in the WRF output files, the system processes the frame and produces geolocated PNG files for each specified output product, then transfers the PNG files by ssh to the visualization server (`wrfxweb`, Section IV-D below) along with metadata as a JSON file. When the transfer is completed, the system executes a command remotely on the visualization server to add the metadata for the frame to its global JSON catalog of simulations.

Aside from the default products, such as the fire area, fire heat flux, wind vector field, and smoke, the post-processing system supports also custom products, such as integrated particulate matter ($PM_{2.5}$), smoke at a specified height, or winds at a specified pressure height, often used for aviation support.

If specified in the job description, the visualization system can create from each time frame Geographical Information Systems (GIS) output files in the standard GeoTIFF format. The conversion is designed not to lose information, i.e., each fire mesh node becomes a GeoFIFF pixel, without any interpolation. This is achieved by specifying in the GeoTIFF the same geographic coordinate system and projection that WRF uses (a special Lambert Conformal Conic projection centered on the simulation grid, based on spherical Earth with the radius 6370 km). In addition, the coordinate system needs to be adjusted for the fact that WRF mesh nodes are in the centers of the mesh cells, while the coordinates of GeoTIFF pixels refer to their corners. Wind vectors are written as GeoTIFF files with two bands comprised of the two directional components. This approach allows the map portal to choose a more appropriate display for a given scaling.

In addition, when processing a time interval, satellite data acquired and used in the system is also postprocessed into geolocated PNG files and and written as GeoTIFF files using the same fire mesh limits. The system takes into account that their geolocation uses WGS-84 ellipsoid without projection.

After the visualization files are created and transmitted, the system waits for the next time frame, or a line in the log indicating that the job has completed.

D. Web Visualization Portal

The web visualization portal `wrfxweb` is written in JavaScript. It uses the global JSON catalog of simulations, transmitted to it by `wrfxpy`, to build its menus and display animations of simulations. Each animation runs in a loop, which gets longer as additional time frames arrive. The visualization server is able to treat satellite data interpolated to the fire mesh the same way as fire simulations. The web server also makes available the download of a Google Earth file, with the same structure and overlays as the animation.

An instance of the visualization server `wrfxweb` runs publicly at http://demo.openwfm.org. It was used to obtain the screenshots in Fig. 6 below.

V. Data Acquisition

A. Static Data

The atmospheric model WRF runs at scales of hundreds of m to several km. In addition to standard surface data in numerical weather prediction (NWP), such as elevation and land use, WRF-SFIRE requires fine-scale elevation maps and fuel information for the fire propagation. Fuel information is available from LANDFIRE[5] as the number of one of 13 fuel categories [19], at the 30m resolution. This information is processed into a single GeoTIFF file covering CONUS. Elevation data from LANDFIRE are likewise processed into a single GeoTIFF file with 30m resolution for the whole CONUS.

B. Atmospheric Data

The atmospheric model for WRF-SFIRE uses initial and boundary conditions provided by a coarser-scale, gridded, 3D numerical weather prediction product, which needs to span the duration of the simulation. `wrfxpy` currently supports a number of weather forecast models such as the High-Resolution Rapid Refresh model (HRRR) [20], the North American Mesoscale model (NAM) [21], the North American Regional Reanalysis (NARR) [22] and the Climate Forecasting System Reanalysis (CFSR) [23]. The CFSR is a global reanalysis product at approximately 38 km resolution, and is available from 1979 to now, at 6 h intervals. The NAM and NARR models have coverage of North America, while CFSR

[5] https://www.landfire.gov

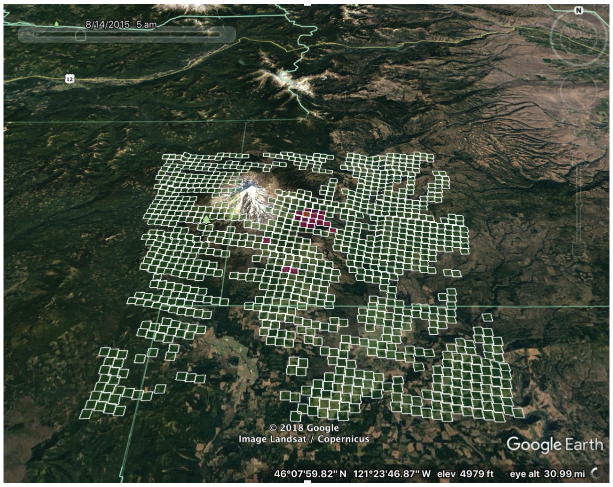
8/14/2015 5 am
© 2018 Google
Image Landsat / Copernicus
Google Earth
46°07'59.82" N 121°23'46.87" W elev 4979 ft eye alt 30.99 mi

Fig. 2. Example of Level 2 MODIS data, one granule restricted to a fire simulation domain. Red pixels are fire detection, green pixels are no fire detections, and transparent pixels are missing data. 2015 Cougar Creek fire, WA.

offers a global coverage. Two NAM versions are supported: NAM218 (executed 4 times a day and providing 12 km resolution forecast), and NAM227 (runs on the same schedule, but at 5 km horizontal resolution). The NAM218 is available up to 84h into the future, while the NAM227 is limited to 60h. The HRRR offers only a 36h forecast and its coverage is limited to the continental US, but it, unlike the other reanalysis and forecasting products, is executed in-line with an extensive data assimilation system ingesting radar data in order to help initialize convective processes. The HRRR archive starts at the end of 2014. The NARR reanalysis product has a grid-spacing of 32 km and provides grided historical meteorological data from 1979 to near present.

The fuel moisture nowcasting model is running off the Real Time Mesoscale Analysis (RTMA) [24], providing hourly surface meteorological fields, as well as elevation, on the 2.5 km resolution CONUS grid.

C. Satellite Fire Detection Data

Satellite data come as raw instrument readings (Level 0 and Level 1), which are processed into Level 2 products. Level 2 data consists of logically rectangular arrays of pixels along the satellite flight path, called granules, accompanied by a separate geolocation file for each granule, which contains the geographical coordinates of each pixel. The geolocation files are large and shared by all Level 2 products. Level 2 data are recommended for science use [25], while Level 3 data, which consists of fire detection pixels only, are typically used in operational applications to mark the fire locations. Level 3 products are compact, since only the fire detection pixels and their coordinates are transmitted, but they do not make a distinction between a pixel with no data available, e.g., because of clouds or obscuration by the terrain, and an observation of clear ground without a fire detected.

The satellite fire detection products acquired by our system are Level 2 Active Fires satellite data (Fig. 2) from polar-orbiting satellites, Terra and Aqua with the Moderate Resolution Imaging Spectroradiometer (MODIS) [25] and Suomi NPP with Visible Infrared Imaging Radiometer Suite (VIIRS) [26]. This satellite fire detection data consists of a categorical mask, where every pixel is classified to be either unknown, non-fire, or fire, and may be further distinguished by a confidence level.

Metadata of all available Active Fires granules intersecting the space-time bounding box of the fire mesh of WRF-SFIRE and the simulation interval are collected using Python CMR API.[6] The metadata contains links to NASA Land, Atmosphere Near real-time Capability for EOS (LANCE)[7] for current data and the NASA Land Processes Distributed Active Archive Center (LP DAAC)[8] for archive data. In the LANCE database, only data in the previous 7 days is stored and it is made available within 3 hours of the satellite observation. In contrast, LP DAAC database contains all past available granules after additional cleaning and calibration, with the latency within 8-40 hours of the satellite overpass. Sometimes a granule is found in both resources; in that case, the LP DAAC database is selected because of better data quality compared to the LANCE database. Finally, geolocation and Active Fires products for each granule are matched and downloaded using their url link from the metadata.

Best-case resolutions for satellite products are roughly 375 m [27] and 1 km [25] for the VIIRS and MODIS products, respectively. Because these satellites scan the terrain below them in a “back and forth” manner perpendicular to the flight path, resolution decreases as the scan angle increases from the nadir view. The new geostationary GEOS satellites provide observations several times each hour, but at a lower resolution of approximately 2 km and much less in high latitudes.

All atmospheric and satellite data files and the outputs of UNGRIB are cached, so repeated runs are faster when they require files already in the cache.

VI. Automatic Ignition by Machine Learning

We now present a procedure for automated initialization of a coupled fire-atmosphere simulation from satellite data when the fire has already grown. We want to retrieve the fire arrival time for all nodes on the fire simulation grid, and then we can start the coupled model in a consistent state using a spin-up (Section VII-B below). Existing approaches to estimation of the fire arrival time from satellite data include direct mapping of the first time in a year when fire was detected [28], spatial interpolation [29], and kriging [30].

We cast the estimation fire arrival time as a separation problem, which can be solved by a support vector machine (SVM, [31]). The fire arrival time can be interpreted as a surface in time-space, with two horizontal dimensions (longitude and latitude) and time as the vertical dimension.

[6] https://pypi.org/project/Python-cmr

[7] https://earthdata.nasa.gov/earth-observation-data/near-real-time

[8] https://lpdaac.usgs.gov

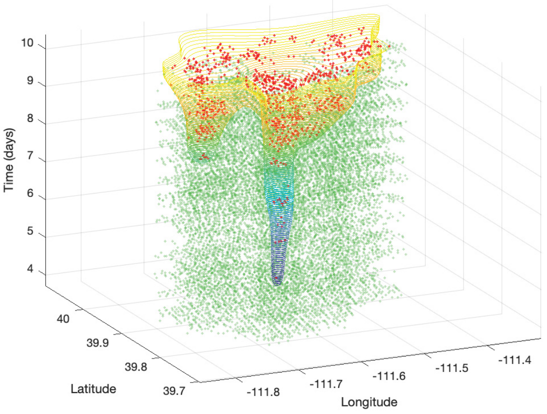

Fig. 3. Estimation of the fire arrival time for the 2018 Pole Creek fire from satellite Active Fires detections. Upper and lower bounds from satellite fire detections as red and green points respectively, and fire arrival time as the separating surface from SVM as contour lines plot.

Then, the presence of a fire can be interpreted as burning and not burning points in time-space, and the fire arrival time becomes a separating surface between the burning and non-burning time-space points (Fig. 3).

We apply the SVM with training data consisting of pairs (u_i, v_i), where $v_i = (t_i, x_i, y_i) \in \mathbb{R}^3$ (time, longitude, latitude) are points in space-time, and $u_i = 1$ if there is fire detected at v_i and $u_i = -1$ if not. The SVM then proceeds by fitting a function f of the form

$$f(v) = b + \sum_{j=1}^{N} w_j \ K(v, v_j), \ \ f(v_i) \approx u_i,$$

with kernel K and suitable weights w_j, and predicts that an arbitrary time-space point v is on fire if $f(v) > 0$ and v is not on fire if $f(v) < 0$. We use the standard Gaussian kernel $K(v, v') = e^{-\gamma ||v-v'||^2}$. The fire arrival time $T(x, y)$ at location (x, y) is then found from the separating surface $f(v) = 0$ by solving for T from

$$f(v) = 0, \quad v = (T, x, y). \tag{1}$$

SVM has no notion of looking for the graph of a function. Thus, the data needs to be preprocessed so that artificial fire and no-fire detections are added after each real fire and before each real no-fire detection respectively (once on fire, always on fire after, and clear ground, always clear before) and no-fire detections after a fire detection are blocked (once on fire, never clear ground). After that, the function f is found using the `svm.SVC` method from the Python package `scikit-learn` [32]. The hyperparameters, the kernel width and a penalty constant, are optimized by a grid search cross-validation. Finally, $T(x, y)$ at each node (x, y) of the fire grid is found as the minimal solution T of (1) by line search and inverse cubic spline interpolation.

The procedure is done in near-real time on a single core. Download of all fire detection data the MODIS sensor on on Terra and Aqua satellites and VIIRS on the Suomi-NPP satellite in the a typical 100 km by 100 km simulation domain and 7-day simulation window takes currently about 25 min. Then, the preprocessing and interpolation to the fire model mesh takes about 6 seconds. After that, currently an automatic 5-fold cross validation is performed to tune the hyperparameters and is the main bottleneck, taking 1 to 2 h depending on the complexity of the data. The crossvalidation executes the SVM many times, evaluating possible values of the hyperparameters by a 2D grid search. This bottleneck could be overcome exploring a method called the Relevance Vector Machine [33] (RVM), which performs similarly to SVM but without the necessity of tuning a pair of hyperparameters. Once we have the best values for the hyperparameters, the training is done in 70 s and the classification of the 3D mesh in 3 min. Finally, the postprocessing part where we define the final fire arrival time is done in 2 min. Therefore, currently the process takes between 1-3 h.

VII. Assimilation of Satellite Fire Detection Data

Many methods exist to update a fire spread model from data, see, e.g., the survey [34]. Reinitializing simulation directly from satellite fire detection pixels was proposed in [35] for a fire-spread model and in [36] for a coupled model. But since satellite data is much lower resolution than the fire model and burdened with errors and missing data, we use data assimilation to improve the model output only in a statistical sense.

Monte-Carlo data assimilation methods, such as ensemble and particle filters, run a large number of fire simulations, e.g., [2], [37]–[41]. However, the computational cost of running an ensemble of high-resolution coupled fire-atmosphere simulations is large, and when only the fire state is updated, the fire and the atmosphere states become inconsistent. Therefore, we are currently using a Maximum Aposteriori Probability (MAP) estimator from Bayesian statistics, e.g., [42], which operates only on a single simulation, and we keep the atmosphere and the fire states in sync by a spin-up, originally developed for perimeter ignition [12], [13].

A. *Maximum Aposteriori Probability Estimate*

With fire arrival time denoted by T^f, the MAP is formulated as the minimization problem

$$-\sum_{G} \sum_{(x,y)\in G} c_G(x,y) f_{G,x,y}(T - t_{G,x,y}) \\ + \frac{\alpha}{2} \|T - T^f\|_A^2 \to \min_{T}.$$

Here, the function $f_{G,x,y}(T - t_G, x, y)$ is the log likelihood of the conditional probability of a satellite observation in location (x, y) at the time t_G, which is the time associated with the satellite granule G containing the pattern of detections of fire, of no-fire condition, and and of missing data in the area of interest. The term $c_G(x, y)$ is the confidence level.

The data likelihood function can be derived from validation studies of the VIIRS Active Fires detection algorithm [27] as well as from the MODIS geolocation documentation [43]. This

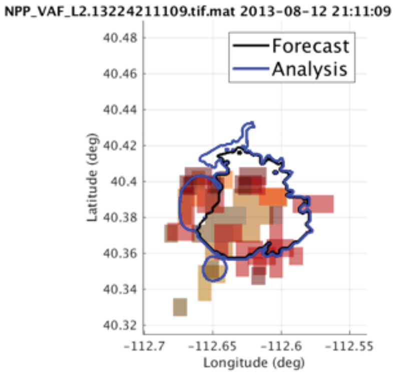

Fig. 4. The effect of data assimilation on the model forecast for the Patch Springs Fire of 2013. Note how the analysis has moved the fire perimeter to encompass the large number of fire detections on the western section of the fire but has not done so for the comparatively sparser sets of detection in the southwest.

data likelihood function gives the conditional probability of the satellite fire detections given the state of the model. More details on this likelihood function are given in [6]. The penalty term in the MAP reflects a prior belief about the state of the fire, to favor smooth changes in the fire arrival time and prevent overfitting to the satellite data. The operator A in the squared norm $\|T-T^f\|_A^2$ is a fractional power of the Laplace operator, implemented by Fast Fourier Transform. Further details of the derivation of this term are given in [12].

B. Atmosphere Model Spin-up

Because the WRF-SFIRE system couples a fire spread model with an atmosphere model and data assimilation of satellite fire detections changes the model's fire arrival time but does not change the state of the atmosphere, inconsistencies between the state of the fire and the state of the atmosphere develop, effectively breaking the coupling between the two models as the simulation is advanced forward in time. In order to overcome these inconsistencies, a technique for using an artificial fire history called the spin-up has been created [4]. In spin-up, the fire model is suspended and heat fluxes are computed from a prescribed fire history, i.e., a given fire arrival time.

After the fire arrival time has been modified by data assimilation (Fig. 4), the fire and the atmosphere model are brought back into a consistent coupled state again over a spin-up period (Fig. 5). For this purpose, we create an artificial fire history to blend the forecast and the analysis fire arrival time as follows. The coupled model produces a forecast fire arrival time T^f. Data assimilation of satellite fire information collected over a time interval T^r (restart time) to T^p (perimeter time) is performed in order to generate an analysis fire arrival time T^a. Then, the spin-up fire arrival time T^s is computed as a weighted average of the analysis T^a and the forecast T^f for the time period over which the satellite fire detections were collected. The heat fluxes from the spin-up are inserted into the atmosphere model, bringing the state of the atmosphere back into a consistent state with the fire model. Finally, the coupled model is advanced again from the state of the fire and the atmosphere at the end of the spin-up period, T^p.

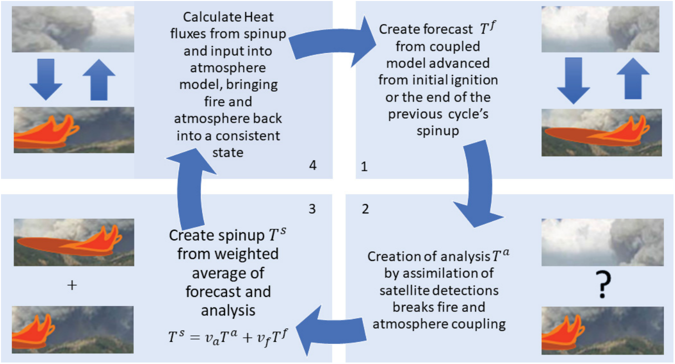
Calculate Heat fluxes from spinup and input into atmosphere model, bringing fire and atmosphere back into a consistent state
4
Create forecast T^f from coupled model advanced from initial ignition or the end of the previous cycle's spinup
1
3
Create spinup T^s from weighted average of forecast and analysis
$T^s = v_a T^a + v_f T^f$
+
2
Creation of analysis T^a by assimilation of satellite detections breaks fire and atmosphere coupling
?

Fig. 5. Flow chart of the data assimilation process, progressing in a clockwise manner from the upper right. 1) The WRF-SFIRE coupled model is advanced from an initial ignition or from a previous cycle's spinup, creating the forecast T^f. The fire and atmosphere are in a consistent state. 2) Assimilation of satellite data changes the state of the fire, giving the analysis T^a, but does not change the state of the fire. Fire and atmosphere are no longer in a consistent state. 3) The spin-up T^s, an artificial fire history, is made from a weighted average of the forecast and analysis. 4) Heat fluxes from the spin-up are computed and input into the atmosphere model, bring the fire and atmosphere back into a state of consistency. The model can now be advanced into the next cycle.

For each location (x, y) in the fire simulation mesh, the spin-up T^s is computed as a weighted average of the forecast T^f and the analysis T^a according to the weighting equations

$$w_f = \max(T^f - T^r, 0) \qquad w_a = \max(T^p - T^a, 0)$$
$$v_f = 1 - \frac{w_f}{w_f + w_a} \qquad v_a = 1 - \frac{w_a}{w_f + w_a}.$$

We note that $w_f + w_a$ is always a positive quantity and that that $v_f + v_a = 1$. Finally, the spin-up is given by

$$T^s = v_f T^f + v_a T^a.$$

VIII. Case Study

The Pole Creek Fire was ignited by lightning on September 6, 2018 in Juan County, Utah. This wildfire burned over 98,000 acres, making it one of the largest wildfires in Utah state history. On September 12, strong winds and low relative humidity caused an explosive fire growth resulting in the evacuations of several communities adjacent to active burn region. By September 17, the Pole Creek Fire merged with the Bald Mountain fire and was classified in the Incident Command System as an incident Type 1, which defines the wildfire as a major incident as it was very large, complex, and requiring multi-agency efforts and national resources for successful suppression and control. On October 6, the fire was finally reported by fire management officials as being fully contained.

Due to the fire's close proximity to major population centers (Provo and Salt Lake City), a number of WRF-SFIRE simulations were carried out to forecast the fire growth, smoke transport, and predict potential impacts on air quality. The first set of forecast simulations were generated on September

14 at 0000 UTC, with the forecasts going out 48 hours from the initialization time. Each simulation was re-initialized every 12 hours (0000 and 1200 UTC) until the fire's eventual containment in October. The model domains for this forecast simulation were centered over the fire location, and utilized 2-way nesting between 3 WRF domains at 5, 1.66, and 0.55 km grid spacing. All WRF domains had 97x97 horizontal grid points, each with 41 vertical levels. The fire growth was computed on fire mesh that had a horizontal grid spacing that was 1/20 the size of the innermost WRF domain (1940×1940 grid points). In order to ensure that the Courant-Friedrichs-Lewy (CFL) condition for numerical stability was met, a time step of 3.33 seconds was set for the innermost WRF domain.

On average, a 48-hour simulation was completed in approximately 3 hours and 5 minutes, with each simulation utilizing 150 GB of storage space.

GeoMAC infrared fire perimeters were used to initialize the fire position and locate regions with active burning, when available. The fuel moisture model was initialized with fuel moisture estimates from the National Fuel Moisture Database, which was obtained from the Wildland Fire Assessment System[9].

The atmospheric component of the WRF-SFIRE modeling system was initialized using boundary conditions derived from the NAM model [21], which is described in further detail in section VB. For a 48-hour forecast, 2 days worth of NAM forecast boundary conditions were needed, which used 3.6 GB of storage space. The total time to preprocess boundary conditions and fire input data took an average of 11 minutes. Downloading the NAM model data[10] took approximately 3 minutes, on average.

Visualized output from our WRF-SFIRE forecast simulations were publicly displayed on http://demo.openwfm.org, with the first frame being available 16 minutes after the simulation was initiated. An example of the visualized output can be seen in Fig. 6, which shows the fire location and area, downwind smoke dispersion, and wind vectors at forecast time +0400 hours.

Smoke forecasts from WRF-SFIRE were validated against air quality observations from Salt Lake City and Provo. Air quality was particularly bad at these cities during the morning of September 15th, when drainage flows from nighttime cooling advected smoke from the mountains into the basins below, where Salt Lake City and Provo are located (see the middle panel in Fig. 6). During this time, $PM_{2.5}$ concentrations significantly increased, especially at the QNP air quality site, which was located 7 km north of Provo (Fig. 7). Observations saw an enhancement of $PM_{2.5}$ concentrations that exceeded 80 μg m^{-3} at 1800 UTC, while the model predicted a similar enhancement at the same time, albeit the increase was more gradual from 1200 to 1800 UTC. By 2000 UTC, $PM_{2.5}$ concentrations dropped off sharply, which was likely the result of flow reversal due to daytime heating.

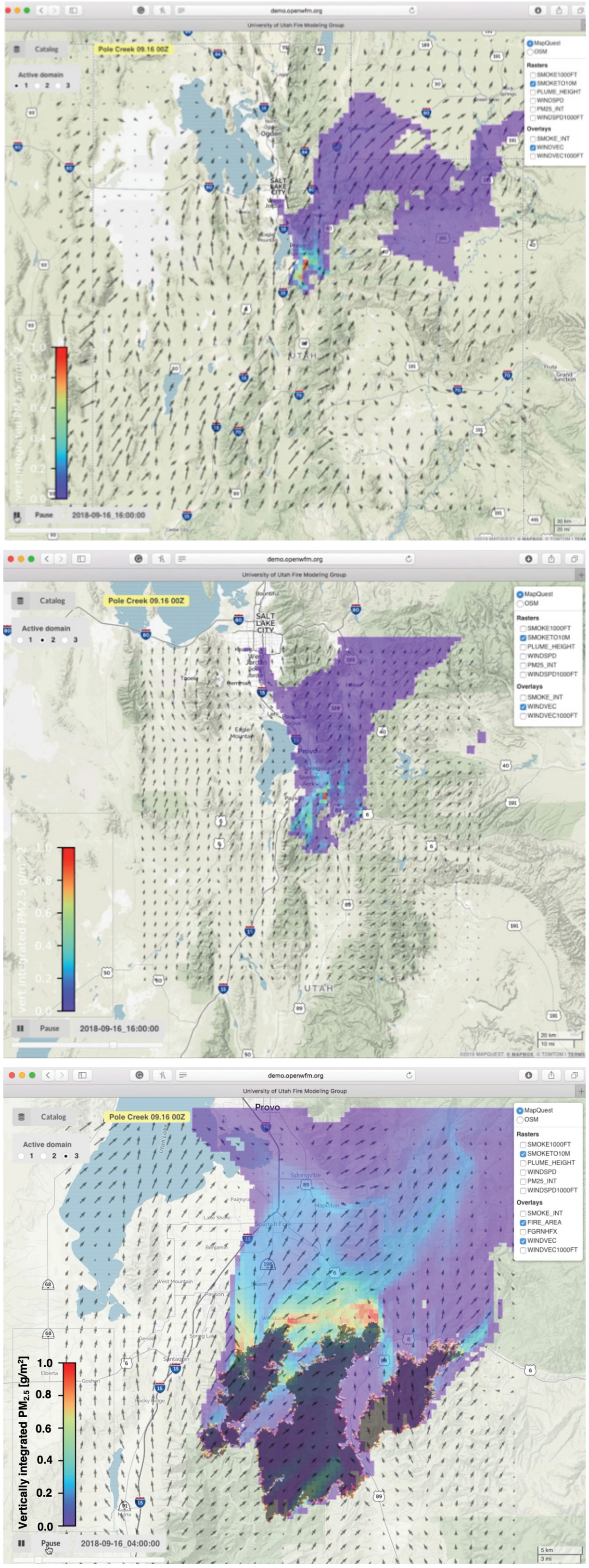

Fig. 6. wrfxweb visualization of WRF-SFIRE simulation of the Pole Creek Fire on September 16th at 0400 UTC (forecast hour = +0400 hours). Color filled contours represent column integrated smoke, dark filled contours represents the predicted fire area, while near-surface winds are plotted as wind vectors. Top panel: 5 km resolution domain 1, middle panel 1.66 km domain 2, bottom panel: 0.55 km domain 3.

[9] http://www.wfas.net

[10] https://nomads.ncep.noaa.gov

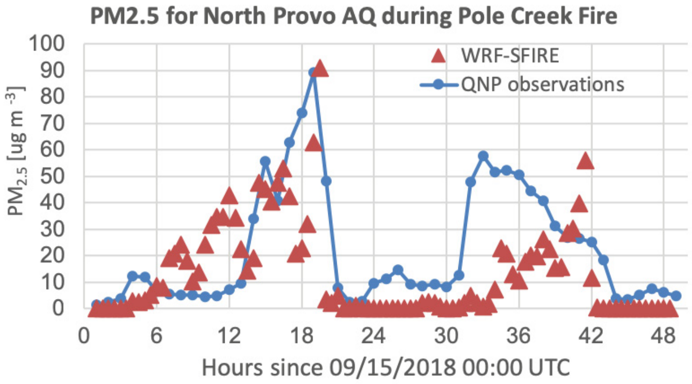

Fig. 7. Observed and model predicted $PM_{2.5}$ concentrations at the North Provo air quality site for a WRF-SFIRE simulation initialized on September 15th at 0000 UTC.

TABLE II
CASE STUDY TIMINGS

Time	Profile from Table I			
	1	*2*	*3*	*4*
All preprocessing	15 min	11 min	38 min	1 h 10 min
Data download	8 min	8 min	32 min	1 h 2 min
HPC simulation	4 h 19 min	3 h 5 min	5 h 39 min	6 h 10 min

Long-range smoke transport was also well captured by the forecast simulations as seen in Fig. 8. GOES-16 satellite data observed a smoke plume that originated from the Pole Creek Fire and drifted to the northeast towards the Uinta mountains and southwest Wyoming on September 16th at 0000 UTC. a smoke plume, with a similar orientation and width was also predicted by WRF-SFIRE, 24-hours into the forecast simulation (initialized on September 15th at 0000 UTC).

The timings of various configurations shown in Table 1, are presented in Table 2. Generally depending on the domain sizes, resolution and the meteorological data, 48 h forecasts required 3-4 hours to complete. The runs with higher resolution of the external meteorological data were expected to provide some computational benefit because of higher starting resolution of the outermost domain. However, as indicated in Table 1, the overall timing of the runs initialized with 5 km and 3 km data came out to take longer than the run initialized with 12 km forcing. The main reason, besides slightly higher resolution of the innermost domains requiring smaller time steps, was due to the download time, which contributed significantly the to total execution time. For instance, download of the 5 km data took 32 min while 12 km forcing were acquired in just 8 minutes. An even bigger difference was observed for the run initialized with the 3 km data, in which case the download time exceeded 1 h.

The system can be configured to interact with schedulers and queues on common HPC systems. The computations reported here were done on 7 Intel Xeon E5-2680v4 2.4GHz nodes with 28 cores each for total 196 cores, in a dedicated queue under the control of the SLURM scheduler on the Kingspeak cluster[11] at the University of Utah. The nodes form a partition owned by the project group in a condo model; they can be used by other users when idle but they are preempted by jobs from the project. The front end node used to run the system and to submit the jobs was a shared interactive dual Intel Xeon E5-2670v2 2.5GHz node. The visualization server is a virtual machine in Digital Ocean cloud with 2 dedicated Intel Xeon E5-2650Lv3 1.8GHz cores, 2GB of memory, and 243GB SSD.

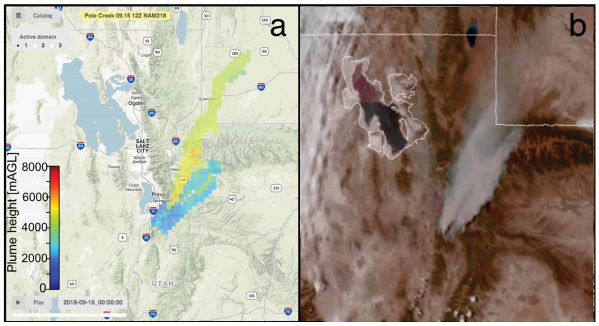

Fig. 8. (a) Model-derived smoke plume top heights [mAGL] on September 16th at 0000 UTC (forecast hour = 24). (b) GOES-16 visible satellite data centered over northern Utah for the same time.

IX. CONCLUSION AND FURTHER WORK

The Pole Creek case study presented here highlights the ability of the model to make accurate, real-time smoke forecasts in an operational setting. Methods for operational use of satellite data were demonstrated and will be moved inside the framework in future work.

Incorporating fire detections from the VIIRS sensor on the NOAA-20 satellite and the ABI sensor on geostationary GOES satellites is in progress. These new sources will require a different API and retrieval from Amazon S3, respectively. Also, GOES fire detection data has a different error profile [44], which may call for some modifications of the methods.

REFERENCES

[1] A. K. Kochanski, M. A. Jenkins, K. Yedinak, J. Mandel, J. Beezley, and B. Lamb, "Toward an integrated system for fire, smoke, and air quality simulations," *International Journal of Wildland Fire*, vol. 25, pp. 534–546, 2016.

[2] J. Mandel, J. D. Beezley, J. L. Coen, and M. Kim, "Data assimilation for wildland fires: Ensemble Kalman filters in coupled atmosphere-surface models," *IEEE Control Systems Magazine*, vol. 29, no. 3, pp. 47–65, June 2009.

[3] J. Mandel, J. D. Beezley, and A. K. Kochanski, "Coupled atmosphere-wildland fire modeling with WRF 3.3 and SFIRE 2011," *Geoscientific Model Development*, vol. 4, pp. 591–610, 2011.

[4] J. Mandel *et al.*, "Recent advances and applications of WRF-SFIRE," *Natural Hazards and Earth System Science*, vol. 14, no. 10, pp. 2829–2845, 2014.

[5] M. Vejmelka, A. K. Kochanski, and J. Mandel, "Data assimilation of dead fuel moisture observations from remote automatic weather stations," *International Journal of Wildland Fire*, vol. 25, pp. 558–568, 2016.

[11] https://www.chpc.utah.edu/documentation/guides/kingspeak.php

[6] J. Haley, A. Farguell Caus, J. Mandel, A. K. Kochanski, and S. Schranz, "Data likelihood of active fires satellite detection and applications to ignition estimation and data assimilation," in *Advances in Forest Fire Research*, D. X. Viegas, Ed. University of Coimbra Press, 2018, in press. Available at https://arxiv.org/abs/1808.03318, July 2018.

[7] J. Mandel, A. Fournier, M. A. Jenkins, A. K. Kochanski, S. Schranz, and M. Vejmelka, "Assimilation of satellite active fires detection into a coupled weather-fire model," in *Proceedings for the 5th International Fire Behavior and Fuels Conference April 11-15, 2016, Portland, Oregon, USA*. Missoula, Montana, USA: International Association of Wildland Fire, 2016, pp. 17–22, https://www.iawfonline.org/conference-proceedings, retrieved October 1, 2019.

[8] W. C. Skamarock *et al.*, "A description of the Advanced Research WRF version 3," NCAR Technical Note 475, 2008.

[9] J. L. Coen, M. Cameron, J. Michalakes, E. G. Patton, P. J. Riggan, and K. Yedinak, "WRF-Fire: Coupled weather-wildland fire modeling with the Weather Research and Forecasting model," *Journal of Applied Meteorology and Climatology*, vol. 52, pp. 16–38, 2013.

[10] P. A. Jiménez, D. Muñoz-Esparza, and B. Kosović, "A high resolution coupled fire-atmosphere forecasting system to minimize the impacts of wildland fires: Applications to the Chimney Tops II wildland event," *Atmosphere*, vol. 9, p. 197, 2018.

[11] D. Muñoz-Esparza, B. Kosović, P. A. Jiménez, and J. L. Coen, "An accurate fire-spread algorithm in the Weather Research and Forecasting model using the level-set method," *Journal of Advances in Modeling Earth Systems*, vol. 10, pp. 908–926, 2018.

[12] J. Mandel, A. K. Kochanski, M. Vejmelka, and J. D. Beezley, "Data assimilation of satellite fire detection in coupled atmosphere-fire simulations by WRF-SFIRE," in *Advances in Forest Fire Research*, D. X. Viegas, Ed. Coimbra University Press, 2014, pp. 716–724.

[13] J. Mandel *et al.*, "New features in WRF-SFIRE and the wildfire forecasting and danger system in Israel," *Natural Hazards and Earth System Sciences Discussions*, vol. 2, no. 2, pp. 1759–1797, 2014.

[14] A. L. Sullivan, "A review of wildland fire spread modelling, 1990-present, 1: Physical and quasi-physical models, 2: Empirical and quasi-empirical models, 3: Mathematical analogues and simulation models," *International Journal of Wildland Fire*, vol. 18, pp. 1: 347–368, 2: 369–386, 3: 387–403, 2009.

[15] T. L. Clark, M. A. Jenkins, J. Coen, and D. Packham, "A coupled atmospheric-fire model: Convective feedback on fire line dynamics," *J. Appl. Meteor*, vol. 35, pp. 875–901, 1996.

[16] A. Kochanski, D. V. Mallia, M. G. Fearon, T. Brown, J. Mandel, and J. K. Vaughan, "Do we need weather prediction models to account for local weather modifications by wildland fires?" in *Advances in Forest Fire Research*, D. X. Viegas, Ed. University of Coimbra Press, 2018, accepted.

[17] W. Wang *et al.*, "ARW version 3 modeling system user's guide," Mesoscale & Miscroscale Meteorology Division, National Center for Atmospheric Research, April 2011.

[18] J. D. Beezley, M. Martin, P. Rosen, J. Mandel, and A. K. Kochanski, "Data management and analysis with WRF and SFIRE," in *Geoscience and Remote Sensing Symposium (IGARSS) 2012*. IEEE, 2012, pp. 5274–5277.

[19] H. E. Anderson, "Aids to determining fuel models for estimating fire behavior," USDA Forest Service General Technical Report INT-122, 1982, https://www.fs.fed.us/rm/pubs_int/int_gtr122.pdf, retrieved October 1, 2019.

[20] S. G. Benjamin *et al.*, "A North American hourly assimilation and model forecast cycle: The rapid refresh," *Monthly Weather Review*, vol. 144, no. 4, pp. 1669–1694, 2016.

[21] NCEP, "National Centers for Environmental Prediction/National Weather Service/NOAA/U.S. Department of Commerce, NCEP North American Regional Reanalysis (NARR)," Research Data Archive at the National Center for Atmospheric Research, Computational and Information Systems Laboratory. http://rda.ucar.edu/datasets/ds608.0, accessed August 13, 2017, 2005, updated monthly.

[22] F. Mesinger *et al.*, "North American regional reanalysis," *Bulletin of the American Meteorological Society*, vol. 87, pp. 343–360, 2006.

[23] S. Saha *et al.*, "The NCEP climate forecast system reanalysis," *Bulletin of the American Meteorological Society*, vol. 91, no. 8, pp. 1015–1057, AUG 2010.

[24] M. S. F. V. De Pondeca *et al.*, "The real-time mesoscale analysis at NOAA's National Centers for Environmental Prediction: current status and development," *Weather and Forecasting*, vol. 26, no. 5, pp. 593–612, 2011.

[25] L. Giglio, W. Schroeder, and C. O. Justice, "The collection 6 MODIS active fire detection algorithm and fire products," *Remote Sensing of Environment*, vol. 178, pp. 31–41, 2016.

[26] W. Schroeder and L. Giglio, "Visible Infrared Imaging Radiometer Suite (VIIRS) 375 m Active Fire Detection and Characterization Algorithm Theoretical Basis Document 1.0," 2016, available at https://viirsland.gsfc.nasa.gov/PDF/VIIRS_activefire_375m_ATBD.pdf, retrieved August 28, 2018.

[27] W. Schroeder, P. Oliva, L. Giglio, and I. A. Csiszar, "The New VIIRS 375 m active fire detection data product: Algorithm description and initial assessment," *Remote Sensing of Environment*, vol. 143, pp. 85–96, 2014.

[28] S. Urbanski, B. Nordgren, C. Albury, B. Schwert, B. Quayle, D. Peterson, and W. M. Hao, "A VIIRS algorithm for near real-time mapping of wildfire burned area in the western United States," *Remote Sensing of Environment*, 2018, submitted.

[29] S. A. Parks, "Mapping day-of-burning with coarse-resolution satellite fire-detection data," *International Journal of Wildland Fire*, vol. 23, no. 2, pp. 215–223, 2014.

[30] S. Veraverbeke, F. Sedano, S. J. Hook, J. T. Randerson, Y. Jin, and B. M. Rogers, "Mapping the daily progression of large wildland fires using MODIS active fire data," *International Journal of Wildland Fire*, vol. 23, pp. 655–667, 2014.

[31] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995.

[32] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, Sep. 2001.

[34] M. Gollner *et al.*, "Towards data-driven operational wildfire spread modeling: A report of the NSF-funded WIFIRE workshop," Available at http://drum.lib.umd.edu/bitstream/handle/1903/17162/WIFIRE_Wks_Report_FINAL.pdf?sequence=1, accessed April 2017, 2015.

[35] A. Cardil, S. Monedero, J. Ramarez, and C. A. Silva, "Assessing and reinitializing wildland fire simulations through satellite active fire data," *Journal of Environmental Management*, vol. 231, pp. 996 – 1003, 2019.

[36] J. L. Coen and W. Schroeder, "Use of spatially refined satellite remote sensing fire detection data to initialize and evaluate coupled weather-wildfire growth model simulations," *Geophysical Research Letters*, vol. 40, pp. 1–6, 2013.

[37] M. Denham, K. Wendt, G. Bianchini, A. Cortés, and T. Margalef, "Dynamic data-driven genetic algorithm for forest fire spread prediction," *Journal of Computational Science*, vol. 3, no. 5, pp. 398–404, 2012.

[38] F. Gu and X. Hu, "Analysis and quantification of data assimilation based on sequential Monte Carlo methods for wildfire spread simulation," *International Journal of Modeling, Simulation, and Scientific Computing (IJMSSC)*, vol. 1, pp. 445–468, 2010.

[39] C. Brun, T. Artes, A. Cencerrado, T. Margalef, and A. Cortés, "A high performance computing framework for continental-scale forest fire spread prediction," *Procedia Computer Science*, vol. 108, pp. 1712–1721, 2017, international Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.

[40] T. Srivas, R. A. de Callafon, D. Crawl, and I. Altintas, "Data assimilation of wildfires with fuel adjustment factors in FARSITE using ensemble Kalman filtering," *Procedia Computer Science*, vol. 108, pp. 1572–1581, 2017, international Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.

[41] M. C. Rochoux, C. Emery, S. Ricci, B. Cuenot, and A. Trouve, "Towards predictive simulation of wildfire spread at regional scale using ensemble-based data assimilation to correct the fire front position," *Fire Safety Science*, vol. 11, pp. 1443–1456, 2014.

[42] A. M. Stuart, "Inverse problems: A Bayesian perspective," *Acta Numer.*, vol. 19, pp. 451–559, 2010.

[43] M. Nishihama, "MODIS level 1A Earth Location: Algorithm theoretical basis document version 3.0," 1997, available at https://modis.gsfc.nasa.gov/data/atbd/atbd_mod28_v3.pdf, retrieved April 1, 2018.

[44] J. V. Hall, R. Zhang, W. Schroeder, C. Huang, and L. Giglio, "Validation of GOES-16 ABI and MSG SEVIRI active fire products," *International Journal of Applied Earth Observation and Geoinformation*, vol. 83, p. 101928, 2019.