FIRESCIENCE.GOV
Research Supporting Sound Decisions

Project Title: Modeling support for FASMEE experimental design using WRF-SFIRE-CHEM

Final Report: JFSP project number: **16-4-05-3**

Principal investigator:

Adam Kochanski, PhD, Research Assistant Professor, Department of Atmospheric Sciences, University of Utah

Co-Principal investigators:

Jan Mandel, PhD, Professor, Department of Mathematical and Statistical Sciences, University of Colorado Denver

Aimé Fournier, PhD, Research Associate Professor, Department of Mathematical and Statistical Sciences, University of Colorado Denver

Mary Ann Jenkins, PhD, Research Associate Professor, Department of Atmospheric Sciences, University of Utah

This research was sponsored in part by the Joint Fire Science Program.

For further information go to http://www.firescience.gov

We would like to acknowledge high-performance computing support from Yellowstone (ark:/85065/d7wd3xhc) and Cheyenne (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation.

List of Tables

List of figures:

Abstract:

Recent advancements in fire-atmosphere numerical modeling have increased the number of physical processes integrated into these coupled models. This greater complexity allows for more comprehensive representation of the coupled interactions and feedbacks between the fire and the

atmosphere. However, as a consequence of these advancements, data requirements for model initialization and validation have increased as well. As coupled fire-atmosphere models utilize local flow properties to parameterize fire progression, emissions and plume rise, integrated in-situ measurements are needed for model validation and future development.

The proposed observational data to be collected during the Fire and Smoke Model Evaluation Experiment (FASMEE) are identified as critical for progressing and transitioning coupled fire-atmosphere models like WRF-SFIRE and WRF-SFIRE-CHEM into operational use. Historical meteorological data, representing typical weather conditions for the anticipated burn locations and times, have been processed to initialize and run WRF-SFIRE to create a set of simulations representing FASMEE's planned experimental burns. Based on analysis of these numerical simulations, this report provides recommendations on the experimental setup that include the ignition procedures, size and duration of the burns, and optimal sensor placement. Newly developed techniques to initialize WRF-SFIRE with weather conditions typical of FASMEE's burn locations, to ignite FASMEE's prescribed burns, and to ensure accurate travel times of subgrid scale fire spread independent of the numerical grid mesh and time step are described. The WRFx system, designed to provide a web-based platform to conveniently create wildfire simulations, is presented as a forecasting tool for use during the experimental phase of FASMEE.

1. Objectives

To summarize, the objectives of this study are to:

- support the FASMEE leadership team by identifying critical measurement needs from the standpoint of evaluation and future development of coupled fire-atmosphere models similar to WRF-SFIRE and WRF-SFIRE-CHEM
- assist in defining burn requirements (size, duration, ignition patterns)
- perform numerical simulations of the expected burns under typical burn conditions
- provide insight into measurement strategies optimal from the modeling standpoint

2. Background

Fire and smoke models are being increasingly relied upon for wildland fire decision making and planning. However, many models are used without adequate validation and evaluation due to the lack of suitable data. Accurate estimates of smoke emissions and dispersion from wildland fires are highly dependent on reliable characterizations of area burned, pre-burn biomass of vegetation, fuels, fuel consumption, heat release, plume dynamics, meteorology and smoke chemistry. Characterizing fire-atmosphere interactions, including wildland fire behavior and plume dynamics, is fundamental to improving estimates of smoke production and dispersion. The proposed observational data to be collected during the Fire and Smoke Model Evaluation Experiment (FASMEE) have been identified as critical for progressing and transitioning newer coupled fire-atmosphere models like WRF-SFIRE and WRF-SFIRE-CHEM into operational use.

Fire and smoke modeling exist across a broad range of model complexities. In general, relatively simple systems are currently used for operational applications (e.g. wildland fire spread and smoke prediction for tactical and strategic planning purposes) and rely on sets of assumptions and algorithms derived from observations and theory. For example, plume models can be as

basic as empirical statistical relationships based on smokestack observations (e.g. Briggs 1969; 1971; 1972) or as complex as full coupled fluid dynamics modeling with explicit treatment of fire-atmosphere dynamics: e.g., Wildland Fire Dynamics Simulator (Mell et al. 2007) and FIRETEC (Linn et al. 2002). Operational applications require that the models have forecasting capability (can be executed fast enough to provide usable forecasts) and that the data needed for their execution are readily available. For those reasons, simpler systems are still used in operational applications. More physically-based fluid dynamical models can explicitly resolve key processes important from the fire and smoke modeling standpoint that must be simplified and parameterized in the simple models. Improved linkages with complex modeling systems offer the ability to develop a new generation of moderately complex but physically-based models for operational application. These would be more accurate across a wider array of conditions than the operational models of today and calibrated using fully dynamic physically-based models. Additionally, progressive increases in computational capacity are expected to greatly enhance the feasibility of running coupled fire-atmosphere models, or hybrid systems (e.g. WRF-SFIRE; Mandel et al. 2011; 2014), for operational use in the next 5-10 years.

As the resolution of weather forecasting models increases, the fire-atmosphere interactions and smoke plumes that had to be treated as sub-grid-scale processes requiring a simplified treatment through external parameterizations become explicitly resolvable. Thanks to improved computational capabilities, operational applications of coupled-fire atmosphere models (e.g. Israeli Matash system based on WRF-SFIRE) become increasingly feasible ("Matash" fire prediction system 2016; Mandel et al. 2014). The rapid increase in the resolution of numerical weather prediction products in recent years opens new avenues for development in the near-to-moderate future of integrated systems (e.g., WRF-SFIRE-CHEM; Kochanski et. al 2015), capable of resolving, in a fully-coupled way, fire progression, plume rise, smoke dispersion and chemical transformations.

In order to advance current operational modeling capabilities, rigorous testing, evaluation and model improvements are needed. The performance of the currently used modeling tools as well as next generation models needs to be assessed. This allows for setting expectations for how well a model will perform under real-world applications, what the level of model uncertainties is, and what are the key sources of these uncertainties that need improvements. Such work can also provide the baseline for distilling from computationally intensive, physics-based models, more intermediate complexity models that are more computationally efficient in time-critical operational settings.

Currently, there are insufficient observational data available to facilitate this work, especially in the context of coupled fire-atmosphere models requiring integrated datasets providing information about the fuel, energy release, local micrometeorology, plume dynamics, and chemistry. Without such observational data, the accuracy and validity of even the most physics-based approaches cannot be fully assessed. Further, known science gaps, such as relationships between heat release, wind fields, plume dynamics and feedbacks to wildland fire behavior, cannot be addressed. This project represents an attempt to identify the most critical observational needs, to assist in planning of the actual experimental burns, and to help in optimization of the sensor locations in those burns.

3. Methods

This study utilizes a suite of numerical simulations, performed using the coupled fire-atmosphere model WRF-SFIRE, and designed to assist in planning the experimental phase of FASMEE. These are simulations of the experimental burns to be conducted in Fishlake, North Kaibab and Fort Stewart, as well as idealized (burner) runs executed for the purpose of cross-comparison against other models. As the exact weather conditions cannot be forecasted for the burns planned for years 2019-2021 (the exact burn dates are not even known), our experimental burn simulations are driven by weather conditions typical of the burn sites during the FASMEE burning seasons. The simulations are driven by time-varying, 3-dimensional historical reanalysis data representing the most typical weather conditions meeting burn requirements and accounting for the possible local diurnal variability in weather conditions. The most typical days are defined by a statistical analysis of historical weather observations from remote automated weather stations closest to the burn sites for days that meet the burn criteria. The statistical methodology used to select the most typical days is described in Section 3.1. These days serve as proxies in terms of weather conditions for the simulations of Fishlake, North Kaibab and Fort Stewart experimental burns. The model development activities needed to perform these simulations (as well as the idealized ones) are described in section 3.2 and the actual simulations are described in section 3.3. Section 3.4 presents a method for statistical optimization of sensor placement.

3.1. Statistical Analysis of Typical Burn Dates

3.1.1. Overview: Defining "typical" for multiple incommensurate values

Coupled weather-fire-chemistry simulations (section 3.3.2) provide important technological support for burn planning, operation and analysis. To be relevant, these simulations are initialized with weather conditions typical of the burn location. In the present analysis, "typical" is defined by a statistical analysis procedure applied only to weather-station data records, independent of any simulations. This is because the space of all possible simulations is unfeasibly large, expensive to navigate, and anyway actual data should be more representative than simulations.

To start, consider a weather state defined by a single quantity e.g., air temperature. Then, one might define a "typical" day as one that has temperature closest to the historical sample mean of the temperature from all days considered. The sample-mean formula must be modified if there are missing data, or if data must be omitted for failing to meet criteria for a burn operation, or in other cases explained below.

More generally, a weather state is defined as a vector of temperature and multiple additional quantities, namely relative humidity, and the speed, direction, and gust of the horizontal wind vector. Such data present a challenge, because the quantities have different units, value ranges, and statistical character, which make them incommensurate. First, the bounds on the value ranges are removed by transforming from relative humidity, speed, direction and gust to dew-point temperature, wind vector, and gust-logarithm. Even after this transformation of value ranges, the weather-state vector-entries still have different units, and their collection for all days considered varies more in some directions and less in others, so that measuring closeness by a simple root of sum of squared differences is not appropriate. Instead, we choose as typical days

the days with the least Mahalanobis distance from the sample mean. The Mahalanobis distance, computed using the sample covariance of the weather state vectors, puts more weight on the differences between weather state vectors along the directions where the collection of the weather state vectors varies less, on average. Another interpretation of this method is as a maximum likelihood: If the weather state vectors were a random sample from a multivariate Gaussian (a.k.a. normal) distribution, the days with smaller Mahalanobis distances from the sample mean would be the more likely ones. Section 4.1 and Appendix A provide the technical details.

3.2. Model Development

3.2.1. Prescribed ignition patterns

Ignition in prescribed burns is quite different from the point ignition in a wildfire. To support prescribed burns the model needs to accommodate arbitrary ignition lines, either continuous, as in walking with a drop torch, as well as consisting of a discrete succession of drops of incendiary devices, dispensed from all-terrain vehicles or aircrafts. The incendiary drops are at arbitrary locations and times, and the distances between drops can be smaller or larger than the fire simulation mesh size, which presents additional modeling challenges. WRF-SFIRE has the ability to simulate continuous drop-torch ignition at steady speeds along straight lines (developed for simulation of the FireFlux experiment; Kochanski et al., 2013). Generalization of this scheme to complex ignition patterns is unwieldy, so, in this project, we have developed a new ignition mechanism by building on earlier work on perimeter ignition.

Instead of modeling the fire spread until the time of the perimeter is reached, perimeter ignition in WRF-SFIRE prescribes the fire arrival time so that an appropriate atmospheric circulation due to the fire forcing can develop (Kochanski et al., 2016; Mandel et al., 2012). The new scheme allows the user to specify the latest fire arrival time at the nodes of the fire simulation mesh as an additional input array.[1] The fire model keeps running, but should the location already be on fire at the specified fire arrival time, the specified fire arrival time has no effect. We have also developed scripts that process large collections of point ignitions (e.g., by aerial incendiary devices), encoded in KML or other files, into fire arrival times at nearby nodes in the fire simulation mesh. The fire arrival array thus produced is inserted into WRF-SFIRE input.

In WRF-SFIRE, the state of the fire is represented by a so-called level set function, defined on the simulation domain. The fire is burning where the level set function is negative, and ignition at a location is implemented by making the level set function take a negative value at that location. However, the level set function is only defined by its values on the nodes of the fire simulation mesh, which are typically spaced several meters to tens of meters apart, while the incendiary device drops are at arbitrary locations. To achieve a sub-fire-mesh representation of the ignition process, the code takes the minimum of the level set function and cones (pointing down) with vertices at the individual ignition points. This approach allows the ignition locations

[1] https://github.com/openwfm/wrf-fire/commit/d13e8fb2491cd56d77323041afa46390a2e50b9a

and times not to coincide with the mesh points or the time steps in the model. The slope of the cones is given by the imposed spread rate in the early stages of the ignition simulation, before the fire reaches the mesh node and the model takes over.

3.2.2. Burner

A burner is an idealized simulation with a stationary and constant heat and emission source. An earlier version of the WRF-SFIRE had implemented a prescribed and ad hoc code for a constant heat source for heat extinction depth studies. However, to maintain compatibility with other parts of the code, we have developed a new burner simulation model. It is specified by providing an array of ignition times on the fire simulation mesh and activating a special switch that turns off the fuel decay with time, keeps the burning rate equal to the rate when the fire starts, and turns off the fire spread. The burner mode uses all other components of the model, such as fuel maps, fuel definitions, and emission factors, so it is not limited to ideal problems, and is automatically compatible with future enhancements of the model. The heat and emission fluxes are specified indirectly by setting fuel properties. The burner case developed for this project is now a part of the WRF-SFIRE distribution, including a more detailed technical description and a script that creates the array of ignition times and inserts it into the input stream.[2] (The burner feature may be of interest to WRF-SFIRE users. We have received an inquiry from a user desiring such functionality recently, and were able to simply direct the user to the burner example provided in the WRF-SFIRE distribution.)

3.2.3. Control and visualization system WRFx

The WRFx system was developed by Martin Vejmelka and partially funded by the NASA grant "Wildland fire behavior and risk forecasting." The WRFx system provides a platform to conveniently create and run wildland fire simulations. It consists of three parts:

1. wrfxpy[3] – The control and execution system. Based on user's specifications in a text file or provided interactively on the command line, the system will
    a. download the selected atmospheric product data for the initial and boundary conditions
    b. process the USGS topography data and LANFDFIRE fuel data for the simulation domain
    c. set up the WRF-SFIRE simulation
    d. execute and monitor the simulation
    e. postprocess the output into geolocated images, and stream the results to a visualization server as soon as partial output becomes available

[2] https://github.com/openwfm/wrf-fire/tree/master/wrfv2_fire/test/em_fire/burner

[3] https://github.com/openwfm/wrfxpy, documentation https://wrfxpy.readthedocs.io

The wrfxpy system currently supports the atmospheric data products HRRR, NAM218, NAM227, and NARR, and ignition anywhere in the contiguous U.S. at any time for which one of the atmospheric data products is available.

2. wrfxweb[4] - The visualization server to show map-based animation of the simulation results. It is currently installed and animations are available at http://demo.openwfm.org.
3. wrfxctrl[5] - The web-based front-end system, which allows users to select the ignition by clicking on a map, pick simulation parameters, and initialize and monitor the simulation. Because the web server controls the cluster, it runs only on a private network behind a firewall, for security reasons.

WRFx enhancements developed by the investigators under this project include:

1. Support for NAM227 high-resolution atmospheric data product.
2. Support of general ignition patterns. This is achieved by the new ability to set up the simulation, interrupt the system, make custom modifications, and then run the simulation and visualize the results. This feature was used in all the simulations reported here.
3. The ability to run simulations set up in wrfxpy on another supercomputer and stream the visualizations directly from that supercomputer. This feature was used to clone thesimulations for the sensitivity study of sensor placement in this project (see section 4.3).

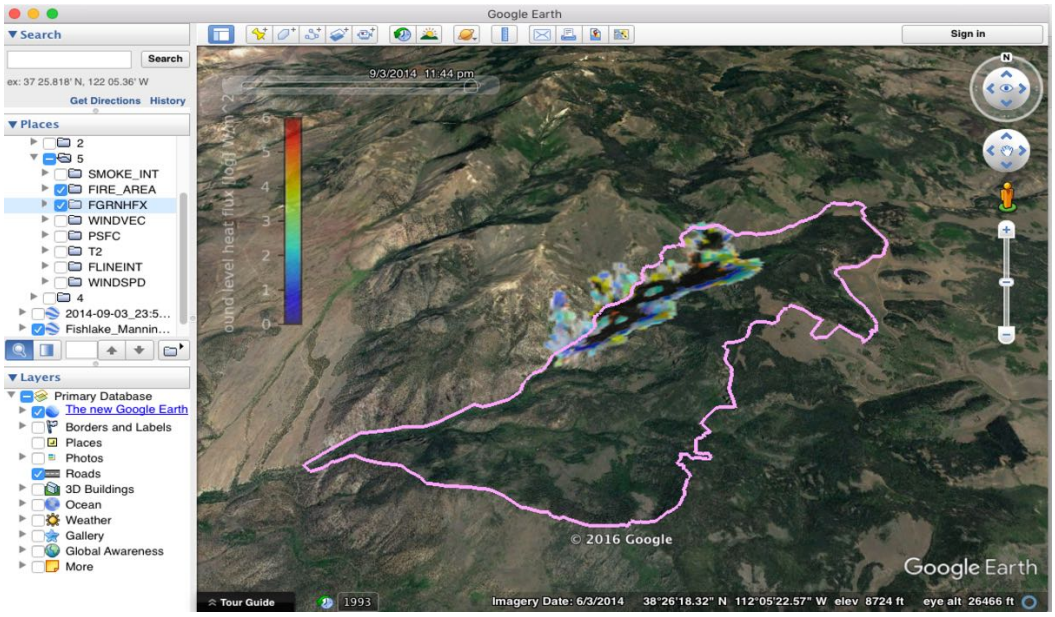


Figure 1. An example from an animation of the fire heat flux from a simulation of the Fishlake burn, produced by WRFx.

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

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

4. Animation of results as KML files running in Google Earth. A sample screenshot from such animation is presented in Figure 1.

3.3. Numerical Simulations

3.3.1. Idealized burner simulations

The newly added capabilities described in Section 3.2 enabled idealized burner-type simulations with fire approximated as a stationary heat source of defined geometry and strength. The main goal of these simulations was to provide numerical data needed for comparison against other models (WFDS, FIRETEC and MesoNH) and to help define minimum energy release measurement requirements for FASMEE experimental burns. The idealized burner simulations were performed using simplified atmospheric forcing in a form of static vertical profiles of wind speed, temperature and moisture, and the stationary surface heat source representing the thermal effect of the fire. The simulations consisted of four runs, executed with two different wind profiles and two lapse rates. These runs were used to assess how WRF-SFIRE performs as a smoke model compared to WFDS and MESO-NH under low and high wind speed conditions in moderately and very stable atmospheres. The detailed model configurations are summarized in Table 1.

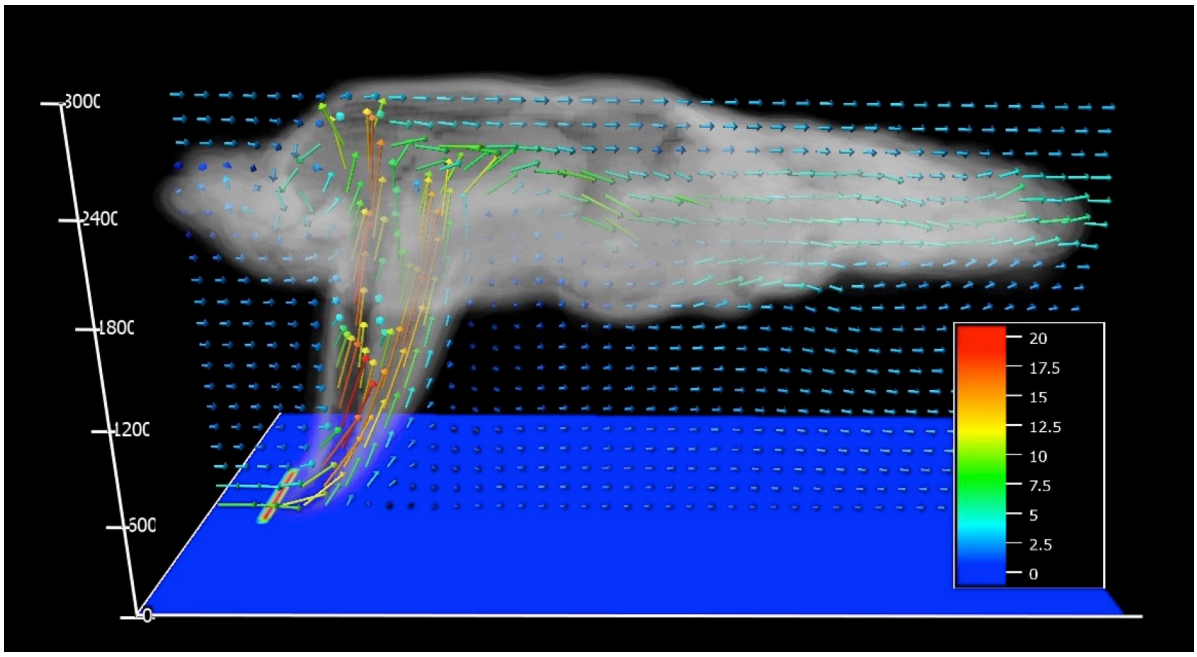


Figure 2: An example of idealized burner simulation (U1_LR-6_50), showing wind vectors in X-Z plane color coded according to the wind speed magnitude, and volume rendered smoke. The red line on the bottom plane shows the location of the burner.

Table 1. Model setup for the idealized burner runs (wind speed and stability sensitivity).

Name of the run:	U1_LR0_50	U1_LR-6_50	U5_LR0_50	U5_LR-6_50
Number of domains	1 (idealized)	1 (idealized)	1 (idealized)	1 (idealized)
Domain size	120×60×60	120×60×60	120×60×60	120×60×60
Horizontal	Δx=Δy=50m	Δx=Δy=50m	Δx=Δy=50m	Δx=Δy=50m
Vertical resolution	38m - 64m	38m - 64m	38m - 64m	38m - 64m
Fire mesh resolution	25m	25m	25m	25m
Height of the model	3000m	3000m	3000m	3000m
Boundary conditions	open	open	open	open
Surface temperature	293.15K	293.15K	293.15K	293.15K
Relative humidity	33%	33%	33%	33%
Lapse rate	LR=0K/km	LR=-6K/km	LR=0K/km	LR=-6K/km
Wind profile	$U(z) = 1\ (z/2)^{1/7}$ m/s	$U(z) = 1\ (z/2)^{1/7}$ m/s	$U(z) = 5\ (z/2)^{1/7}$ m/s	$U(z) = 5\ (z/2)^{1/7}$ m/s
Burner length	750m	750m	750m	750m
Burner thickness	25m	25m	25m	25m
Burner heat flux	1800kW/m^2	1800kW/m^2	1800kW/m^2	1800kW/m^2
Burner start delay	10s	10s	10s	10s
Simulation length	1200s	1200s	1200s	1200s
Output interval	10s	10s	10s	10s
Time step	1s	1s	1s	1s

Besides these four runs (performed at 50m horizontal resolution), six additional runs were performed to assess how horizontal resolution impacts the model's ability to resolve plume dynamics. These simulations were run with horizontal model grid sizes varying from 10m to 400m (see Table 2). All simulations were performed in a Large Eddy Simulation mode, using the 1.5 TKE sub-grid scale closure, open horizontal boundary conditions, a 500m deep top dumping layer and the vertical grid stretched from 38m near the surface to 64m at the model. Vertical wind and temperature profiles were identical to the U5_LR0_50 case (power law wind profile with U_0=5m/s and isothermal atmosphere with 0K/km lapse rate). An example of the burner simulation U1_LR-6_50 is shown in Figure 2.

Table 2. Model configurations for burner runs (resolution sensitivity).

Name of the run:	U5_LR0_10	U5_LR0_12	U5_LR0_25	U5_LR0_100	U5_LR0_200	U5_LR0_400
Number of domains	1 (idealized)	1 (idealized)	1 (idealized)	1 (idealized)	1 (idealized)	1 (idealized)
Domain size	600×300×60	480×240×60	240×120×60	60×30×60	60×30×60	60×30×60
Horizontal resolution	Δx=Δy=10m	Δx=Δy=12.5m	Δx=Δy=25m	Δx=Δy=100m	Δx=Δy=200m	Δx=Δy=400m
Vertical resolution	38m - 64m	38m - 64m	38m - 64m	38m - 64m	38m - 64m	38m - 64m
Fire mesh resolution	10m	12.5m	25m	25m	25m	25m
Model top	3000m	3000m	3000m	3000m	3000m	3000m
Boundary conditions	open	open	open	open	open	open
Surface temperature	293.15K	293.15K	293.15K	293.15K	293.15K	293.15K
Lapse rate	LR=0K/km	LR=0K/km	LR=0K/km	LR=0K/km	LR=0K/km	LR=0K/km
Wind profile	$U(z) = 1\ (z/2)^{1/7}$ m/s	$U(z) = 1\ (z/2)^{1/7}$ m/s	$U(z) = 5\ (z/2)^{1/7}$ m/s	$U(z) = 5\ (z/2)^{1/7}$ m/s	$U(z) = 5\ (z/2)^{1/7}$ m/s	$U(z) = 5\ (z/2)^{1/7}$ m/s
Burner length	750m	750m	750m	750m	750m	750m
Burner thickness	25m	25m	25m	25m	25m	25m
Burner heat flux	1800kW/m^2	1800kW/m^2	1800kW/m^2	1800kW/m^2	1800kW/m^2	1800kW/m^2
Burner start delay	10s	10s	10s	10s	10s	10s
Simulation length	1200s	1200s	1200s	1200s	1200s	1200s
Output interval	10s	10s	10s	10s	10s	10s
Time step	0.25s	0.25s	0.5s	1s	1s	2s

3.3.2. Simulations of planned experimental burns

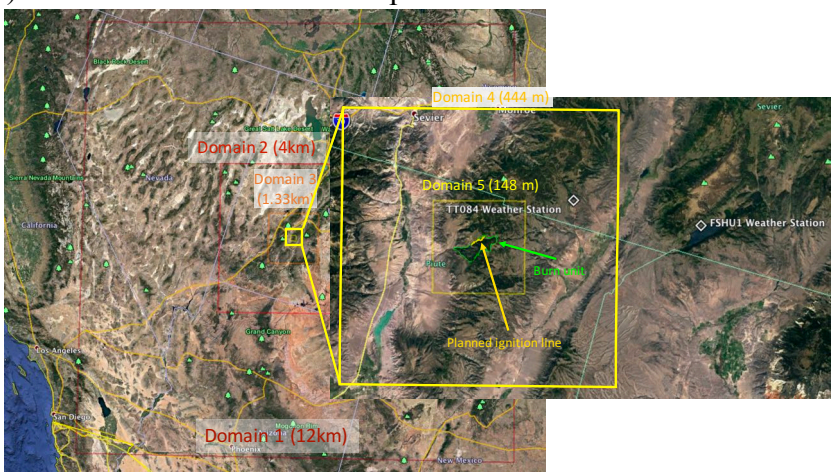
Numerical simulations for the planned experimental burns were performed for the typical days (meeting burn requirements for the specific sites) obtained from the analysis described in section 3.1, in particular Figure 7. The numerical experiments were for Fishlake, North Kaibab, and Fort Stewart burn sites and all used a multiscale setup of telescopic domains of resolutions gradually increasing from 12km to 148m. Each simulation was performed with 30m-resolution topography, land use and fuel maps from LANDFIRE, and driven by time-varying large-scale weather forcing obtained from Northern American Regional Reanalysis (NARR Mesinger et al 2006). Model configurations used for these runs are presented in Table 3.

Table 3. Parameters of the experimental burn simulations

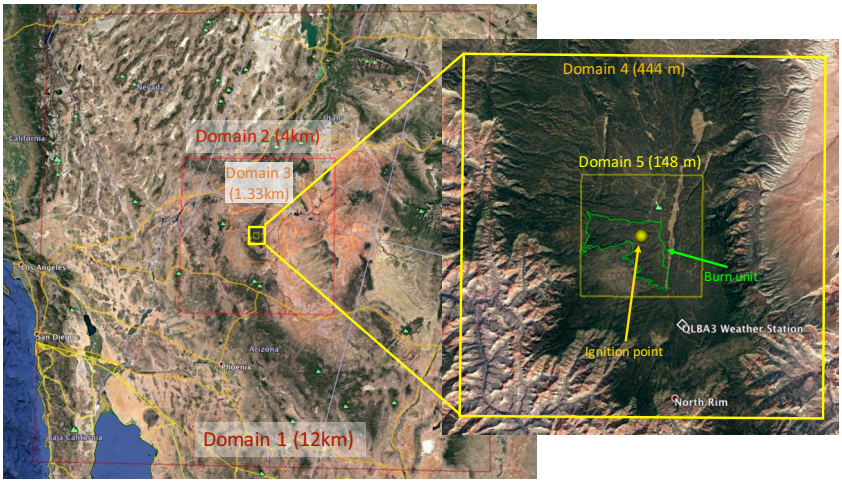
Burn Site:	Fishlake	North Kaibab	Fort Stewart	RxCADRE
Meteo forcing	NARR	NARR	NARR	NARR
Number of domains	5 (realistic)	5 (realistic)	5 (realistic)	7 (realistic)
Domain sizes(X×Y×Z)	97×97×41	97×97×41	97×97×41	97×97×41
Model top	13.2km	13.2km	13.2km	13.2km
Horizontal resolution	12km/4km/1.33km/ 444m/148m	12km,4km,1.33km, 444m,148m	12km,4km,1.33km, 444m,148m	12km,4km,1.33km, 444m,148m/19m/16m
Vertical resolution	5.3m – 2233m	5.3m – 2233m	5.3m – 2233m	5.3m – 2233m
Fire mesh resolution	29.6m	29.6m	29.6m	1.6m
Total number of runs	7	4	10	1
Ignition	'Helicopter ignition'	Point ignition	Straight Line/Point	3 ATV lines
Simulation start	09.03.2014 00:00 UTC 09.11.2016 00:00 UTC 09.22.2012 00:00 UTC 09.26.2015 00:00 UTC	09.05.2008 00:00 UTC 09.19.2015 00:00 UTC 09.01.2001 00:00 UTC 09.02.2011 00:00 UTC	04.22.2014 00:00 UTC 04.27.2009 00:00 UTC 02.27.2013 00:00 UTC 02.26.2008 00:00 UTC	11.11.2012 15:00 UTC
Ignition Time	15:00 UTC (9:00 local)	15:00 UTC (9:00 local)	15:00 UTC (11:00 local)	18:00 UTC (14:00 local)
Simulation Length	48h	48h	48h	5h
Fire Output Interval	5 min	5 min	5 min	10s
Time step (d05)	0.5s	0.5s	0.74s	0.022s
1hr fuel moisture	6.0%	6.0%	15.0%	12.2%
10hr fuel moisture	8.0%	8.0%	13.0%	17.3%
100hr fuel moisture	9.0%	9.0%	13.0%	26.0%
1000hr fuel moisture	12.0%	12.0%	18.0%	40.6%

Domain configurations for the simulations of prescribed burns, including burn plot boundaries, ignitions and locations of weather stations providing data for the typical day analysis are presented in Figure 3.

a) Fishlake WRF domain setup



b) North Kaibab WRF domain setup



c) Fort Stewart WRF domain setup

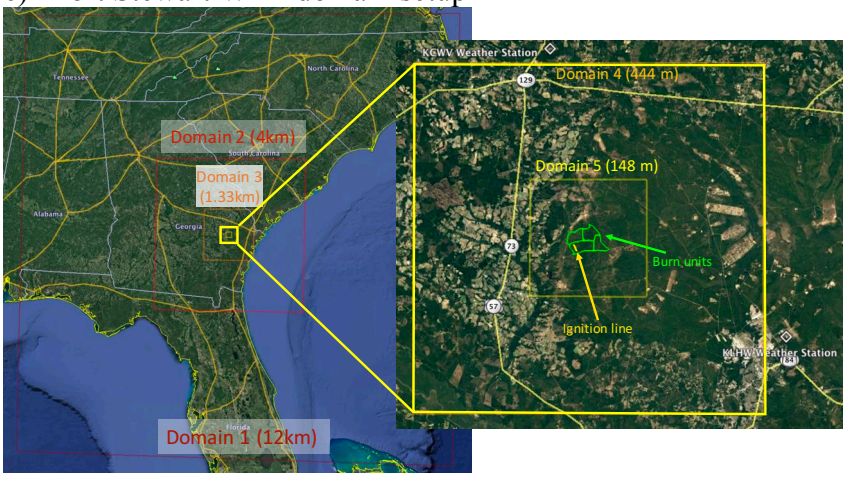


Figure 3. WRF domain setups for burn simulation: a) Fishlake, b) North Kaibab, c) Fort Stewart with indication of nearby meteorological stations used to define typical days.

3.3.3. Test simulation of the RxCADRE L2F burn

For simulations of fires started from arbitrary ignitions (for instance helicopter ignitions, or ATV ignitions) defined as a group of points of known latitude, longitude and ignition time, a new ignition capability was added to the model (see section 3.2.1 for details). The new ignition code was employed in a preliminary simulation of the RxCADRE L2F experiment which again used a multiscale setup providing gradual downscaling from 12km to 16.5m, using 7 telescopically nested domains (see Figure 4a).

a) b)

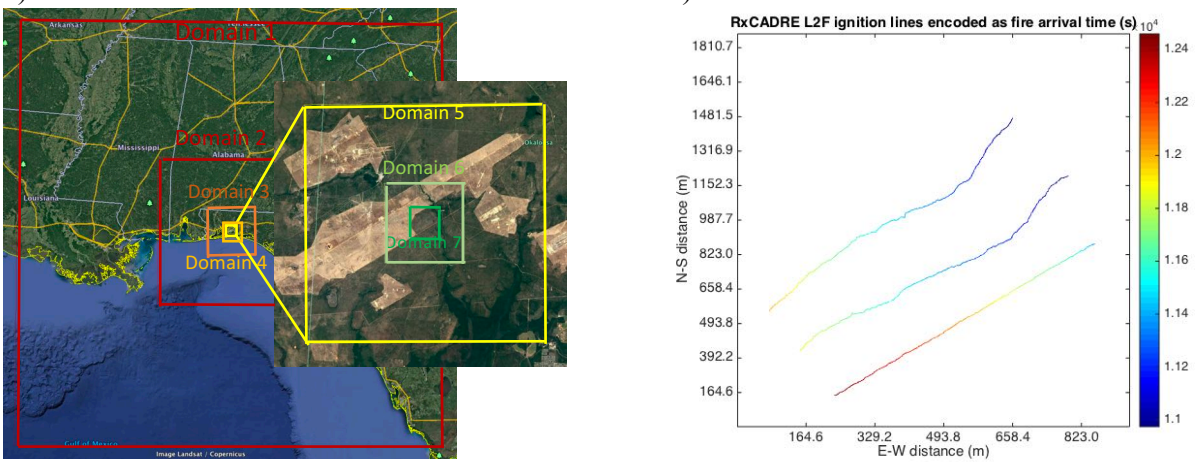


Figure 4. a) RxCADRE L2F domain setup, b) RxCADRE L2F ignition lines encoded as the fire arrival time in secoonds from the simulation start at 11.11.2012 15:00 UTC (10:00 local time).

The northern and middle ignition line was defined directly from the GPS records. The southern ignition line for which no GPS data are available, was placed along the southern plot boundary, while its timing was approximated based on WASP IR data (Hudak et al. 2016). All three ignition lines were encoded in one fire arrival matrix defining ignition times and locations (see Figure 4b). Model configuration details are presented in Table 3.

3.4. Statistical optimization of sensor placement

It is most useful to place (the limited number of) sensors where the observations or parameters of interest change or show the largest variance. To estimate the sensitivity of measurements to model parameters, global sensitivity analysis with repeated Latin Hypercube Sampling (McKay et al. 1979; McKay 1995; Saltelli et al. 2004) is used. Consider first a simple precursor method: run a number of simulations with the values of the parameters of interest chosen randomly and then compute the variance of the output over the simulations at various potential sensor locations. The method utilized here improves on such a conventional approach in two aspects: repeated Latin Hypercube Sampling (rLHS) allows for a drastic reduction in the number of simulations to be made by a systematic coverage of the parameter space, and the Sobol (2001) variance decomposition can distinguish the effect of individual parameters. This method was originally developed to evaluate computer simulations at the Los Alamos National Laboratory (McKay et al. 1979; McKay 1995) and it found use recently in guiding sensor placement in oil reservoir engineering (Chugunov et al. 2014). Appendix B provides the technical details.

4. Results and Discussion:

4.1. Analysis of Typical Days

As an example, the input data (meteorological states comprising temperature, relative humidity, wind speed, direction and gust) for the "typical day" analysis described in section 3.1 are displayed in Figure 5. Similar figures for other stations are presented in Appendix A. These figures are used as a "sanity check" i.e., a minimal quality-control of the data (indeed, some spurious values were discovered this way), as well as to visualize the burn requirements for state values, day-of-year and time-of-day, and the relative amounts of state values that do and do not meet the burn requirements. By visually summing over the abscissas of each plot, an experienced statistician can infer that the variability of the raw data is far from that of a multivariate Gaussian random process.

The transformed data and their statistics are presented in Figure 6. is a decreasing function of the area of the green ellipse, and each ellipse major axis tilts up or down as correlation is positive or negative, respectively. To an experienced statistician, it is apparent from the scatter of plot points that the transformed state is still not very impressively Gaussian, but certainly is much more Gaussian than the original (temperature, relative humidity, wind speed, direction and gust) description. Also, the ellipse tilts and major-minor axis ratios confirm the general impression of the scatter of data points that meet the burn requirements. Similar figures for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A. for remaining stations can be found in Appendix A.

The culmination of these analyses is set of norms (root-mean-squared values) $\|\boldsymbol{\delta}\|$ of the Mahalanobis (or standardized) deviation vectors $\boldsymbol{\delta}$, shown in Figure 7. By definition, the most "typical" day has the smallest $\|\boldsymbol{\delta}\|$, as indicated in each sub-plot title. These typical days, along with the days exhibiting the successively next-smallest $\|\boldsymbol{\delta}\|$ values, inform the simulation initializations described in section 3.3. Note that although the statistics defining $\|\boldsymbol{\delta}\|$ were conditioned on burn requirements of state-value ranges, the time requirements were not used to define $\|\boldsymbol{\delta}\|$; otherwise, there would have been too few acceptable data to yield robust statistics. Also, since the time requirements are independent of the values requirements, there may possibly be many smaller $\|\boldsymbol{\delta}\|$ outside the required time windows than inside. A summary of the typical days defined for the analyzed burn sites is presented in Table 4. Figures similar to Figure 7 for remaining stations can be found in Appendix A.

Table 4. Summary of most typical days identified for the burn sites.

Burn site	Fishlake		North Kaibab	Fort Stewart	
Weather station	FSHU1	TT084	QLBA3	KCWV	KLHW
Most typical days	2014-09-03	2012-09-05	2008-09-05	2014-04-22	2009-04-27
	2015-09-26	2012-09-04	2005-09-19	2013-02-24	2013-02-26

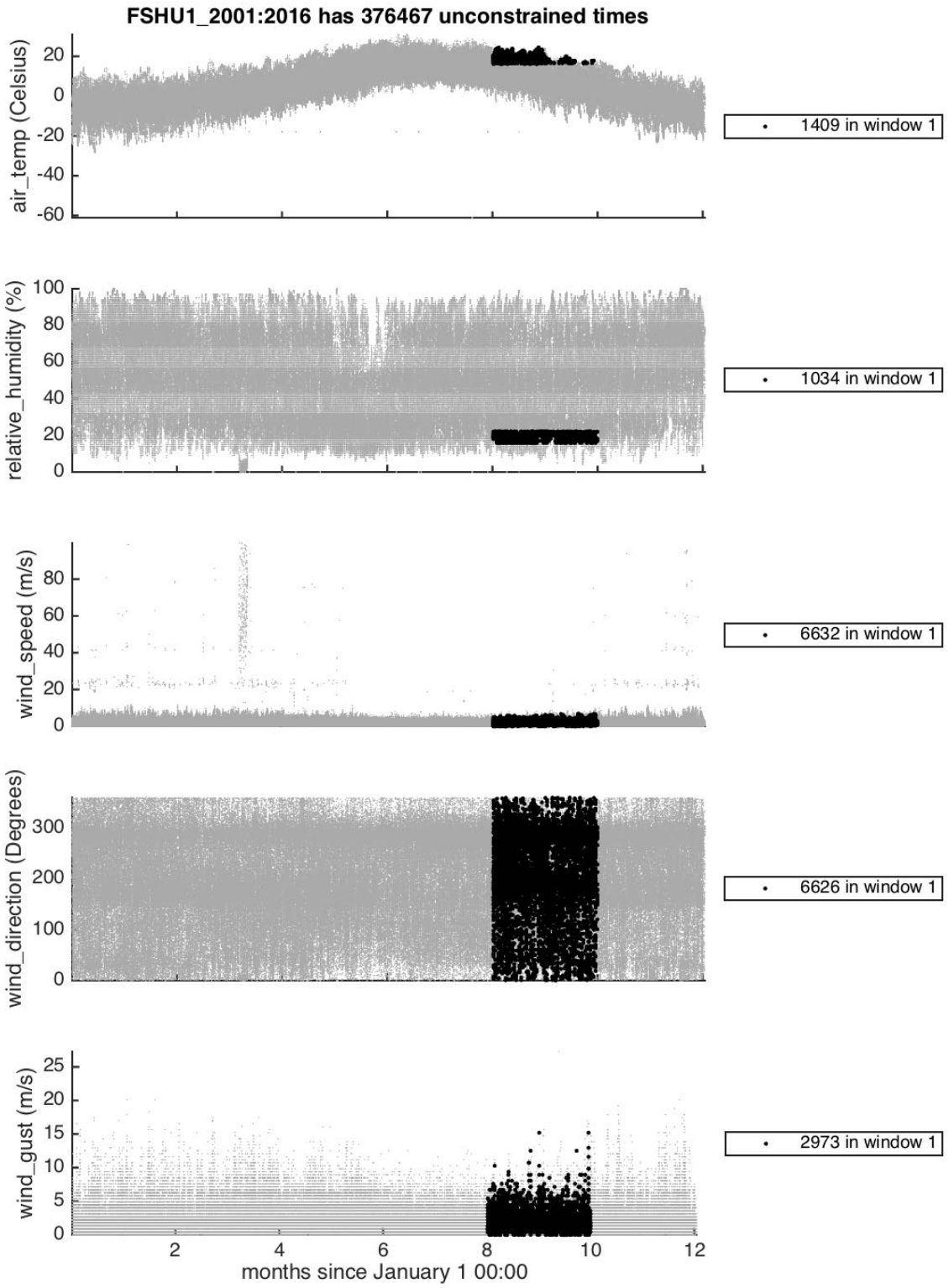


Figure 5. Station data (ordinate) vs month number (abscissa) for station FSHU1 (Fish Lake ranger station near Koosharem 10ENE, 38.55167°N, 111.72278°W, 8880 ft in Utah) in years 2001--2016. Points that do

and do not meet the burn-requirements (Table 6) are dark and light gray, respectively, and enumerated in the legends and titles, respectively.

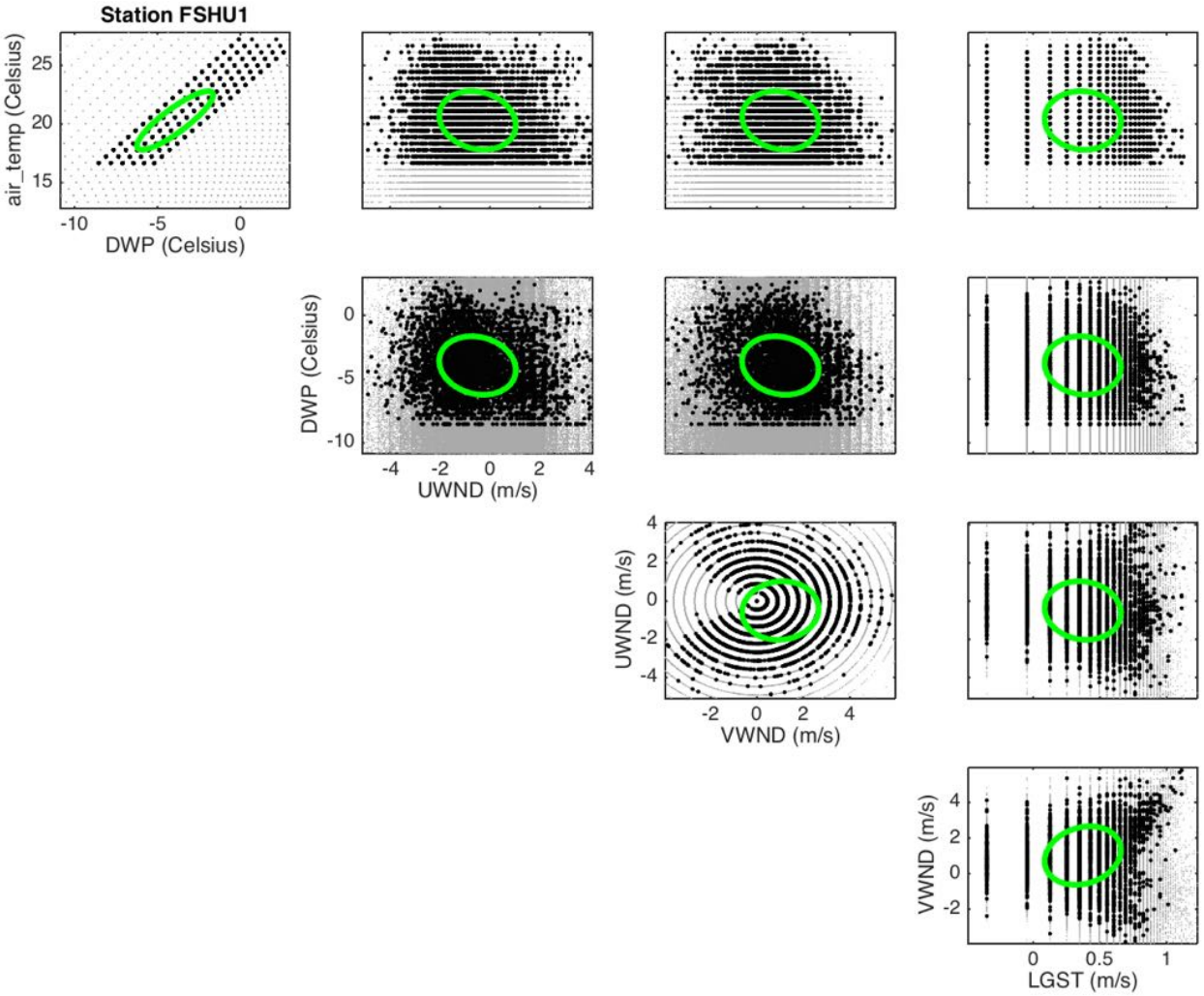


Figure 6. Scatter plots of FSHU1 state values x_j (ordinates, rows j = 1:4) vs x_k (abscissas, columns $k - 1 = j$:4). Black and gray indicates values that are and are not constrained by the burn requirements. The marginal covariance is depicted by the θ-parameterized green curve

$$(x_j, x_k) = (\langle x_j \rangle, \langle x_k \rangle) + ((1 - \rho_{jk}^2)/(1 - \rho_{jk} \sin 2\theta))^{1/2} (\sigma_j \cos \theta, \sigma_k \sin \theta)$$

that is an ellipse of area $\pi(1 - \rho_{jk}^2)^{1/2} \sigma_j \sigma_k$ inscribing the rectangle

$$[\langle x_k \rangle - \sigma_k, \langle x_k \rangle + \sigma_k] \times [\langle x_j \rangle - \sigma_j, \langle x_j \rangle + \sigma_j].$$

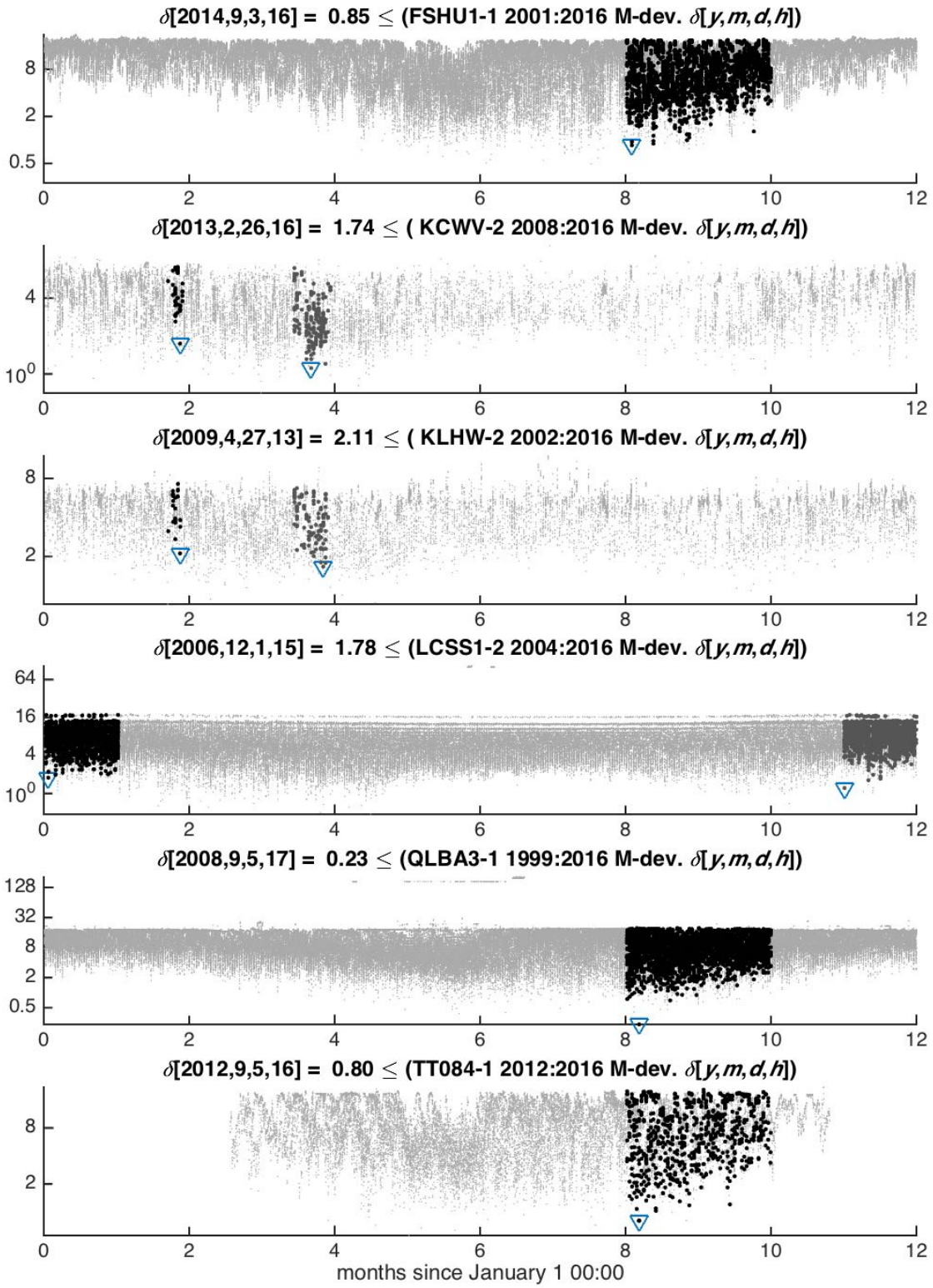


Figure 7. Time series of Mahalanobis (standardized) deviation norm $\|\delta\|$ for each station (6 rows). Points for states that do and do not meet burn requirements are dark and light gray, respectively. For each station, the most typical day in the burn window has the smallest $\|\delta\|$ and is listed in the title and indicated with a □ marker.

4.2. Results from Numerical Experiments

4.2.1. Idealized burner simulations

The idealized burner simulations performed during this study provide opportunities to compare smoke simulated by different models and to learn about the sensitivity of the simulated plume dynamics to the model parameters (for instance model resolution). The full cross comparison of all models tested during the first phase of FASMEE is presented in the final report from WFDS/FIRETEC modeling group led by Ruddy Mell. Here in Figure 8, as an illustration, we show plume vertical cross-sections for WRF-SFIRE and Meso-NH, simulated under two different lapse rates and two wind profiles at 1000s into the simulation. The top row shows simulations with 0C/km lapse rate, or isothermal atmosphere, and high winds (see case U5_LR0_50 in Table 1), the middle row shows simulations with negative lapse rate -6C/km and high winds (case U5_LR-6_50), the bottom row shows low the wind case with negative lapse rate -6C/km (case U1_LR-6_50). In the wind dominated cases the plume (top and middle rows) both models show a very similar plume structure. However, in the low wind speed case (more plume dominated), WRF-SFIRE shows a higher and more tilted plume than Meso-NH.

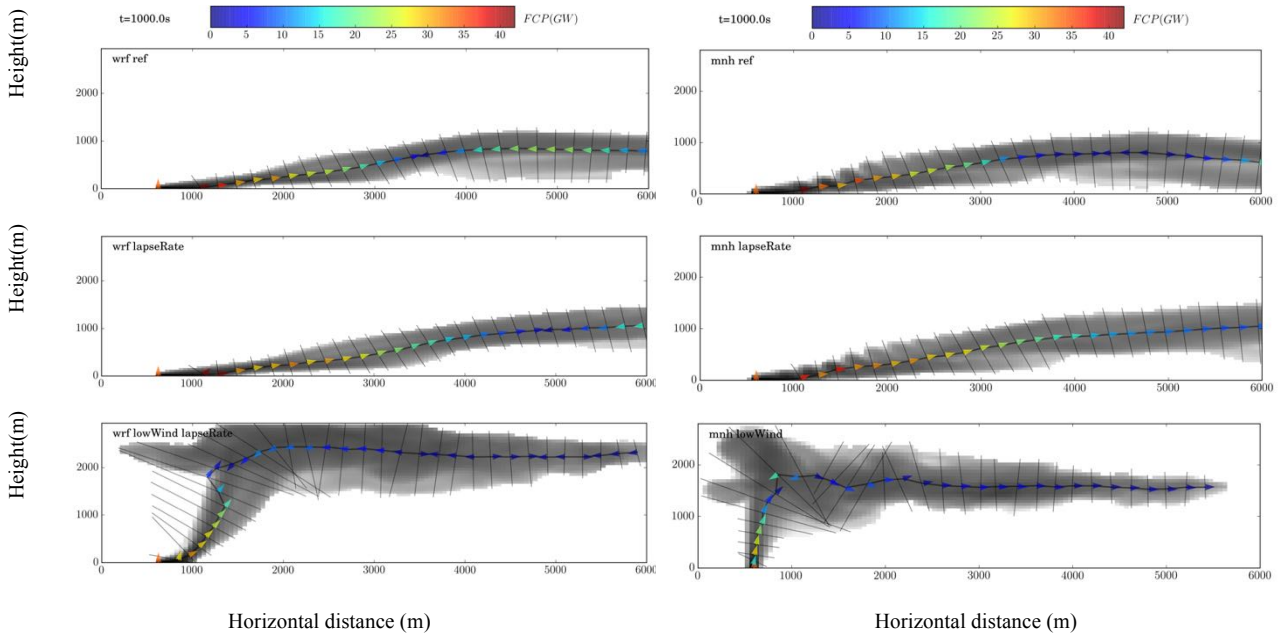


Figure 8. Comparison between X-Z plume cross sections 1000s into idealized burner simulations performed with WRF-SFIRE (left column) and Meso-NH (right column), for different atmospheric profiles. Top row: 0C/km lapse rate and high winds, middle row: -6C/km lapse rate and high wind speed, and bottom row: -6C/km lapse rate and low wind speed.

The impact of the horizontal model resolution on the maximum resolved updraft is shown in Figure 9. All the simulations presented here attempt to resolve the convective updraft generated by a 25m thick and 750m long stationary heat source (burner). The coarsest simulation with 400m mesh is clearly not capable of resolving the updraft magnitude, which in this case reaches only 5 m/s. The highest resolution case (25m) generates updraft velocities up to 20m/s, while simulations run at 200m, 200m and 100m resolutions show similar updrafts between 9 and

11m/s. The plume evolves much slower in the coarse resolution case (400m) than in other cases using finer grids. These results suggest a strong impact of the model resolution on the resolved vertical velocities within the plume. Interestingly, the strength of the most energetic updrafts is generally similar for the model resolutions of 50m, 100m and 200m.

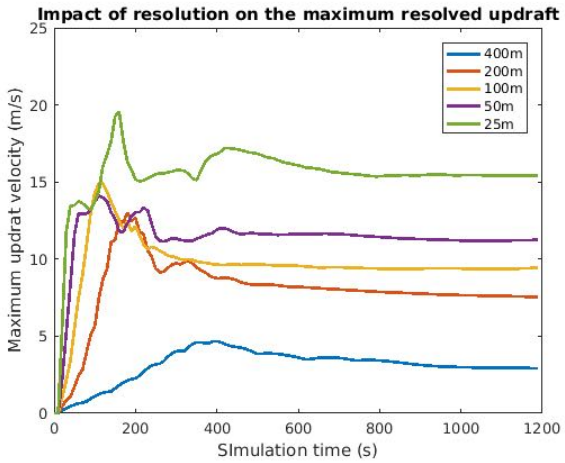


Figure 9. Time series of the maximum updraft velocity simulated at different model resolutions.

The model's resolution dependency for capturing extremes in the vertical velocity does not translate directly into its ability to represent plume height and vertical smoke concentration. As shown in Figure 10 a), all simulations executed with model grids of 100m or less provide very similar plume top heights, despite the fact that resolved maximum updrafts vary considerably. The vertical smoke profiles are also similar for these cases, with peak concentrations located at elevations around 1000m above the ground. In the context of this analysis, the burn simulation, presented in the next section, executed at 149m horizontal resolution, should be able to represent the vertical plume extent from experimental burns planned to be over 500 acres in size (~2km^2).

a) b)

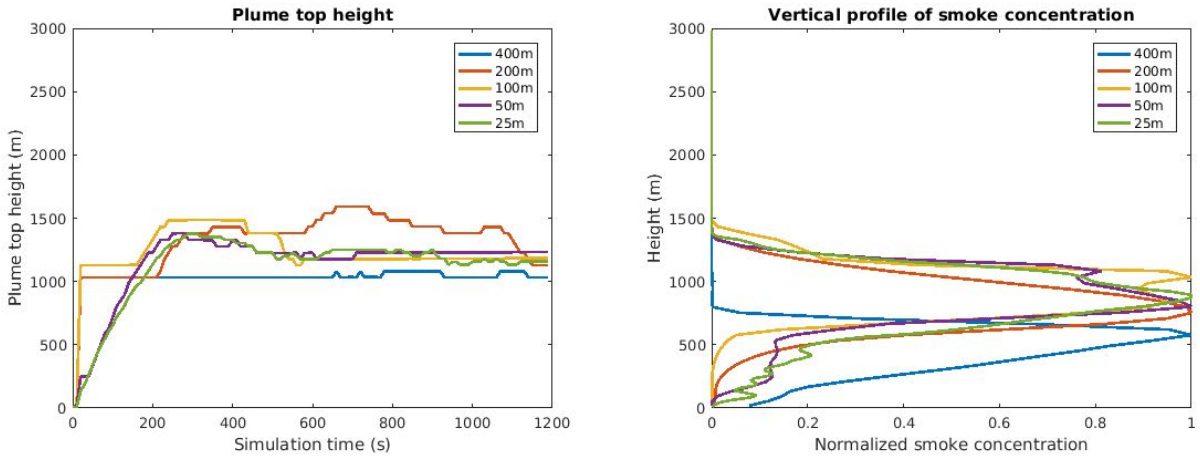


Figure 10. Impact of model resolution on the simulated plume top height a), and vertical smoke concentration profile b).

4.2.2. Simulations of planned experimental burns in Fishlake

The analysis of the expected plume characteristics has been performed for the Fishlake simulations. The simulations were run for day 09.03.2014, which has been identified based on the weather observations from the FSHU1 weather station as the most typical day meeting the burn criteria defined in Table 6:

1. reference simulation without fire,
2. standard simulation with fire, and
3. simulation with fire and doubled heat flux from the fire to the atmosphere.

The first run serves as a reference point for assessing the impact of the fire on the boundary layer when compared to the second run. The third run is intended to account for possible underestimation of the fire heat fluxes under high fuel load conditions characteristic for the Fishlake burn site, as well as to assess sensitivity of the simulated plume height and vertical velocities to the fire heat flux.

As shown in Figure 11, for the standard and doubled heat flux fire simulations, maximum updraft velocities tended to spike during ignition and then gradually rise over the next 4 hours.

a) b)

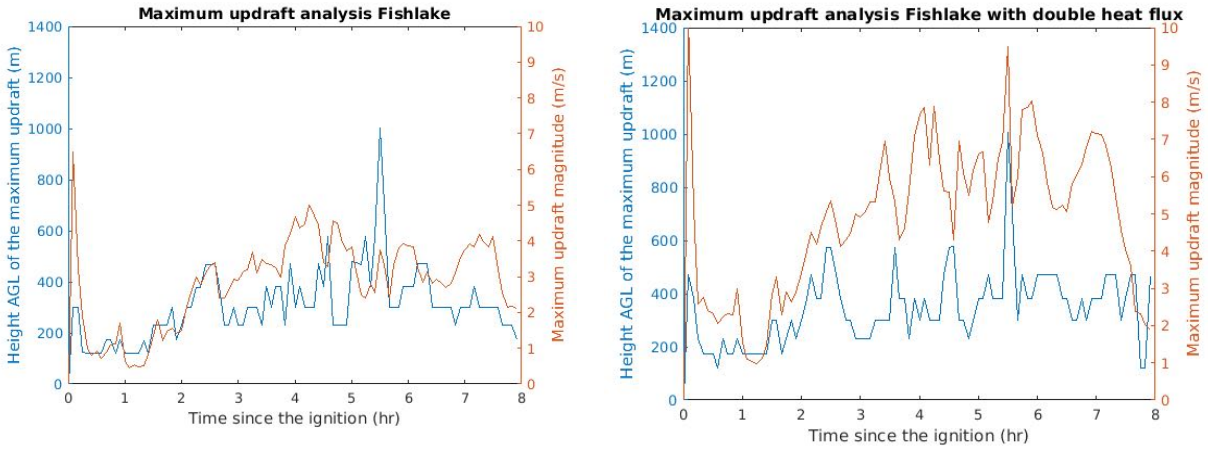


Figure 11. Time series of the maximum updraft strength and location from the Fishlake simulation for 09.03.2014. a) Standard simulation, b) Simulation with doubled fire heat flux released to the atmosphere.

The maximum updraft velocities reach 5 m/s in the standard simulation and twice as much in the simulation with the doubled heat flux. The elevations where maximum updrafts occur are significantly less sensitive to the fire heat flux enhancement. In both cases, they vary between 170m and 1000m. These simulations suggest that from the standpoint of the plume characterization it is important that the experimental burn lasts long enough so that the smoke column fully evolves (at least 4 hours). The vertical in plume velocity measurements (LIDAR vertical scans, UAV platforms) should allow for plume scanning up to at least 1.5 km and be able to withstand vertical velocities up to 10m/s. In order to provide a guideline for the in-plume measurements of chemical species, the simulations were executed with fire smoke represented as a passive tracer. Figure 12 a) shows vertical profiles of the smoke concentration normalized by

its maximum value for the baseline simulation and the case with doubled heat flux. Both cases show similar concentration profiles with a maximum near the ground and a single peak located around 800m above the ground level. In the double heat case smoke penetrates into higher elevations. The time series in Figure 12 b) show a rapid plume rise during the ignition, followed by a 2h period of constant plume top height, followed by a sudden drop and a gradual increase with some further fluctuations. As the heat released during the ignition procedure is also doubled in the double heat flux case, the differences in the plume top heights between the cases are most evident during the first 2h after ignition. Later, as the vertical mixing establishes, these differences reduce from about 550m to about 250m. The concentration plots presented in Figure 12 b) suggest that plume sampling should be focused on the layer 3000m AGL (Above Ground Level), with a particular focus on the 500m and 1500m AGL, where the simulations tend to produce the highest tracer concentrations.

a) b)

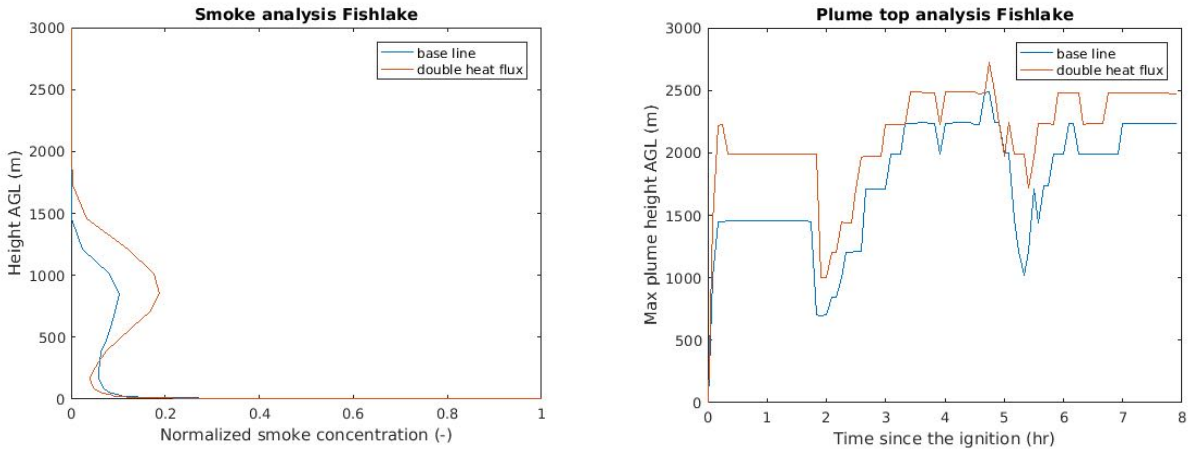


Figure 12. Smoke analysis for the Fishlake burn executed for 09.03.2014. a) Vertical profile of normalized smoke concentrations, b) Time series of the plume top height.

In order to assess the fire's impact on boundary layer flow in the baseline and double heat cases, the horizontal fire-induced winds are computed as a difference between the U and V wind speed components from the fire runs and the reference no-fire run. The time series of the levels where the maximum fire-induced winds occur and their magnitudes for both fire simulations are presented in Figure 13. The elevations of the fire-induced wind maxima are similar in both simulations, suggesting that the level at which the fire accelerates flow most rapidly is not particularly sensitive to the fire intensity (heat flux). However, the magnitudes of the fire-induced wind are. The maximum winds in the baseline simulation reach 7 m/s, while in the double heat flux case they reach 9m/s. The simulations indicate that the maximum wind flow is either close to the ground (15-25m AGL) or aloft, mostly between 800m and 1200m AGL. Therefore, from the measurement standpoint, the optimal sampling strategy would contain a combination of a 25-30m tower and Doppler lidar(s) providing wind speed measurements at elevations at least up to 1500m AGL.

a) b)

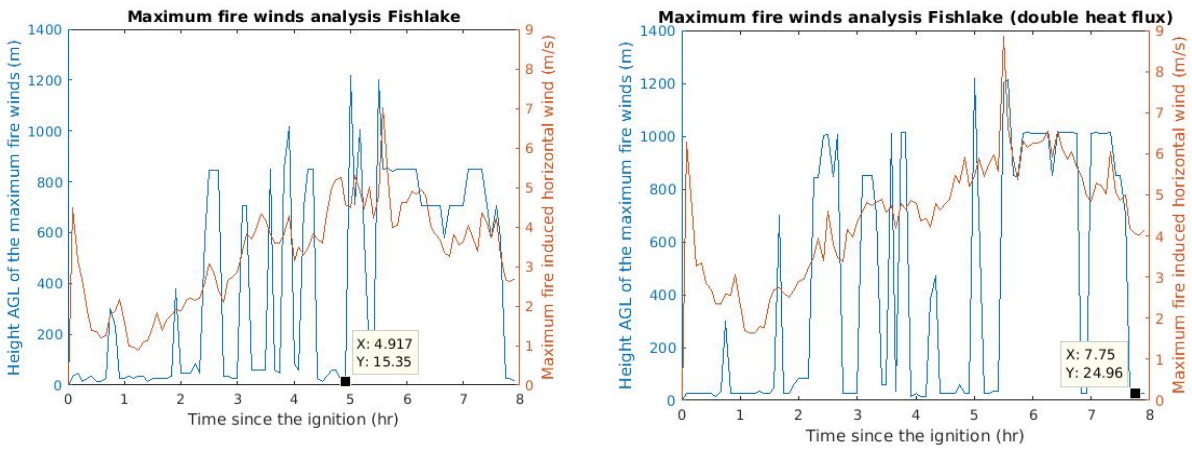


Figure 13. Analysis of the fire-induced winds for the Fishlake simulation executed for 09.03.2014. a) Time series of the maximum fire-induced horizontal winds and their heights in the a) baseline case and the b) double heat case.

4.2.3. Simulations of planned experimental burns in North Kaibab

To provide insight into the second western burn at the North Kaibab site, a simulation was performed for day 09.05.2008, identified based on the weather observations from the QLBA3 weather station as the most typical day meeting the burn criteria defined in Table 6 (presented in Appendix A). OBJ OBJ OBJ Figure 14.

a) b)

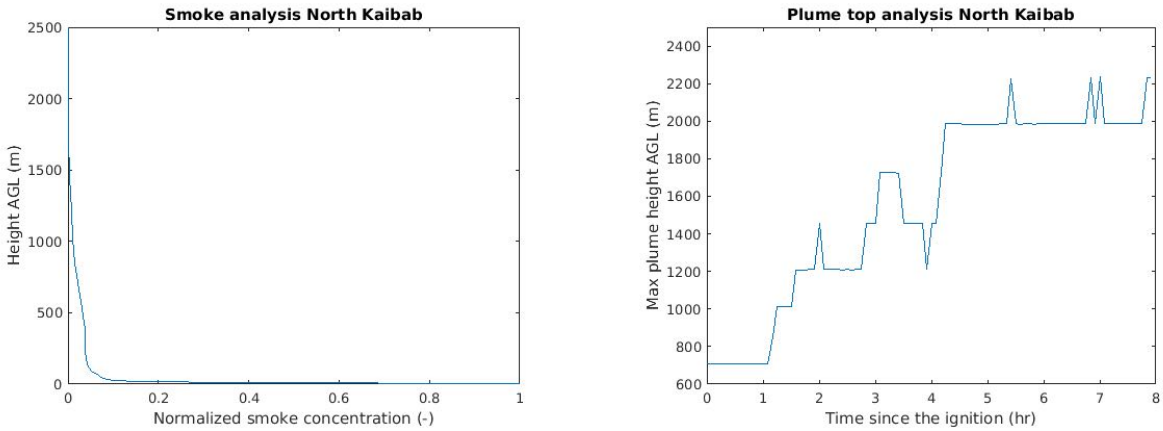


Figure 14. Smoke analysis for the North Kaibab burn simulation executed for 09.05.2008. a) Vertical profile of normalized smoke concentrations, b) Time series of the simulated plume top height.

In the North Kaibab case, Figure 14 a) shows that the smoke layer is more uniformly distributed across the first 1500m than in the Fishlake case, with no evident concentration peak. Figure 14 b) shows lower simulated plume top heights than in the Fishlake case. They increase gradually over the first 4 hours and stabilize at around 2000m, 200-700m lower than in the Fishlake runs.

The evolution of the fire-induced flow is presented in Figure 15. The fire-induced winds in this case are similar magnitude to the Fishlake baseline case, but their time evolution is different. Contrary to the winds from the Fishlake simulation showing a strong initial spike in fire-induced winds followed by a gradual increase, the North Kaibab winds ramp up to about 4m/s, reach maximum of about 7 m/s at 2.5h into simulation, and then keep decreasing. The timing of the peak surface winds corresponds to the time when the maximum updraft is located relatively close to the surface (below 500m AGL). The convective updraft evolves gradually in time, reaching maximum values close to 6m/s after 5h since ignition. Note that, as a result of a point ignition (less intense than the line ignition in the Fishlake burn), this evolution is not disturbed by the initial ignition-induced spike evident in the Fishlake run (see Figure 13). The updraft analysis shown in panel b) indicates that, over the first one-and-a-half hours, the maximum updrafts are located at 2500-3000m AGL (probably as an effect of fire-induced disturbance inducing vertical mixing at high elevations). As the fire evolves, updrafts in the plume core intensify and become dominant, which is indicated by the elevations of the maximum updrafts going back to 500-1000m AGL.

a) b)

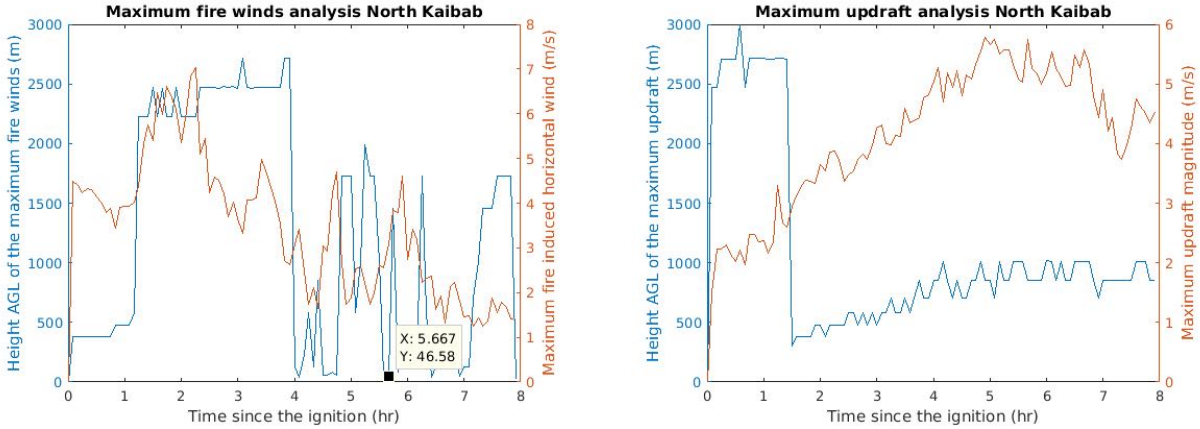


Figure 15. Analysis of the fire-induced wind and updraft for the North Kaibab simulation executed for 09.05.2008. a) Time series of the maximum fire-induced horizontal winds, b) Time series of the maximum fire-induced updraft.

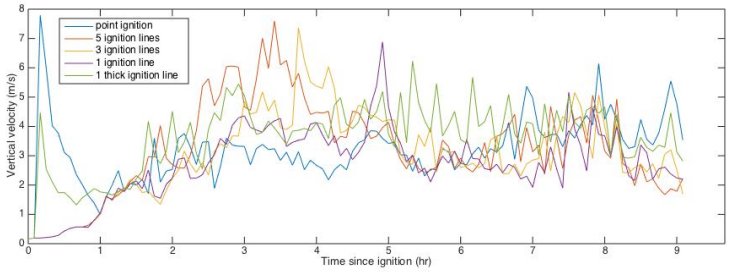
4.2.4. Simulations of planned experimental burns in Fort Stewart

Numerical simulations for Fort Stewart were performed with the main intent of providing an insight into the impact of the ignition procedure on plume evolution. For prescribed burns that require complex aerial ignitions, multiple ignition lines are formed in a relatively short period of time, with significant heat fluxes coming directly from the igniting agents like the gasoline mixture in Helitorch or ping-pong balls (DAID) systems. As the aerial ignition procedure is generally fast and difficult to precisely capture by scanning IR systems which do not provide a continuous static field of view, it is important to know how important the ignition itself is for further plume evolution. In order to assess that, 5 different ignition procedures were simulated: a single ignition point, a single ignition line of two different thicknesses, a set of 3 parallel lines, and a set of 5 parallel lines. All ignition lines were oriented approximately perpendicular to the

mean wind. The results from these simulations are presented in Figure 16. The time series of the maximum vertical velocity and the plume height indicate that the ignition itself plays a big in role in updraft evolution, especially during the initial phase of the burn. The time series from different ignition procedures only start to converge 7 hours after ignition. These results suggest that for experimental burns expected to last less than that, the ignition procedure should be precisely documented in order to allow for realistic representation of the plume evolution in subsequent numerical simulations.

It should be also noted that the initial interactions between ignition lines are hard to account for in numerical experiments performed using models with parameterized fire spread. In these models the ignition process itself is crudely simplified, limiting their capabilities in terms of rendering these effects. From that standpoint, simple ignition patterns are desired in experimental burns so that the fire can be can realistically initialized in the model.

a)



b)

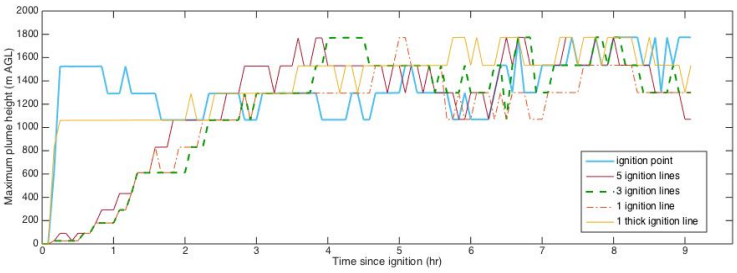


Figure 16. Results from numerical simulations for Fort Stewart (2013.02.14) with different ignition procedures. Time series of the simulated a) maximum updraft and b) plume top height.

Aside from the simulations targeting the ignition effects, additional simulations were performed for 94.22.2014 (a typical day based on statistical analysis of observations from the KCWV weather station; see Figure 3 c). Results from this numerical experiment are presented in Figure 17. The Fort Stewart site is significantly different from south-western Fishlake and North Kaibab sites, from both fuel and topography perspectives. The latter ones are in complex terrain where steep elevation strongly impacts both vertical and horizontal flows, making fire-flow effects harder to detect. It seems that in complex terrain the destabilization induced by the fire can impact the atmospheric state at significant distances from the plume core. Strong upslope winds can induce updrafts of similar magnitude as the fire itself and the interaction between the flow

and the mountain ridges may lead to significant variations in the horizontal winds that are clear only when differences between fire and no-fire cases are computed. As a consequence, the heights of maximum fire-induced winds and vertical velocities fluctuate strongly in the North Kaibab simulations, but are relatively steady in the Fort Stewart burn. In Fort Stewart for instance, the elevations of strongest fire-induced winds are confined to a shallow layer between 0 and 140m AGL, with the majority of strongest fire winds occurring within the 20m layer above the ground (see Figure 17 a). These results suggest that 20m towers and lidars located close to fire line should be able to sample most of the maxima in the fire-induced flow. In the Fort Stewart simulation, both fire-induced winds and updraft velocities evolve very quickly, reaching a quasi-steady state just after one hour from ignition. From the standpoint of these simulations, even though the fuel load and expected fire intensity are lower for Fort Stewart than for Fishlake burns, from the standpoint of characterization of the fire-induced circulation the Fort Stewart site has a potential to provide better and easier to interpret results than the south-western sites.

a) b)

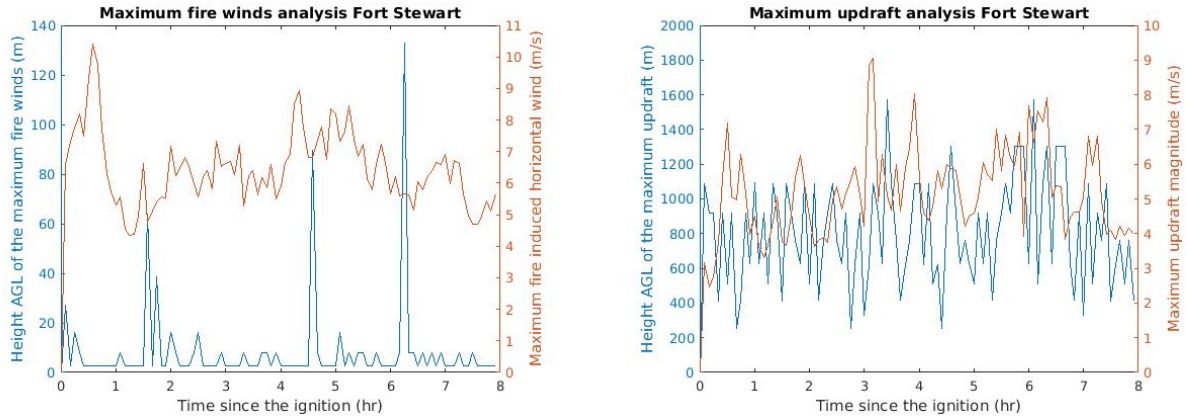


Figure 17. Analysis of the fire-induced wind and updraft for the Fort Stewart simulation executed for 04.22.2014. a) Time series of the fire-induced maximum a) horizontal wind speed and b) updraft.

4.2.5. Test simulation of the RxCADRE L2F burn

The RxCADRE L2F burn has been suggested as proxy for the south-eastern experimental burns in Fort Stewart, due to similar fuel conditions and ignition procedures. The simulated plume height compares well with airborne CO observations (Urbanski et al 2014), indicating the top of the plume at 1238m, with the threshold CO concentration set to 0.13ppm. The plume heights in this case are significantly lower than in the Fishlake case. As shown in Figure 18 b), in this case, the plume top height reaches only 1100m, compared to the Fishlake fire plume over twice as high (see Figure 12 b). These results suggest that for this burn and low intensity burns similar to RxCADRE L2F, Doppler lidars can be located closer to the burn unit in order to provide higher resolution data at the expense of the vertical range, which does not need to be as large as in the case of high-intensity Fishlake burn.

a) b)

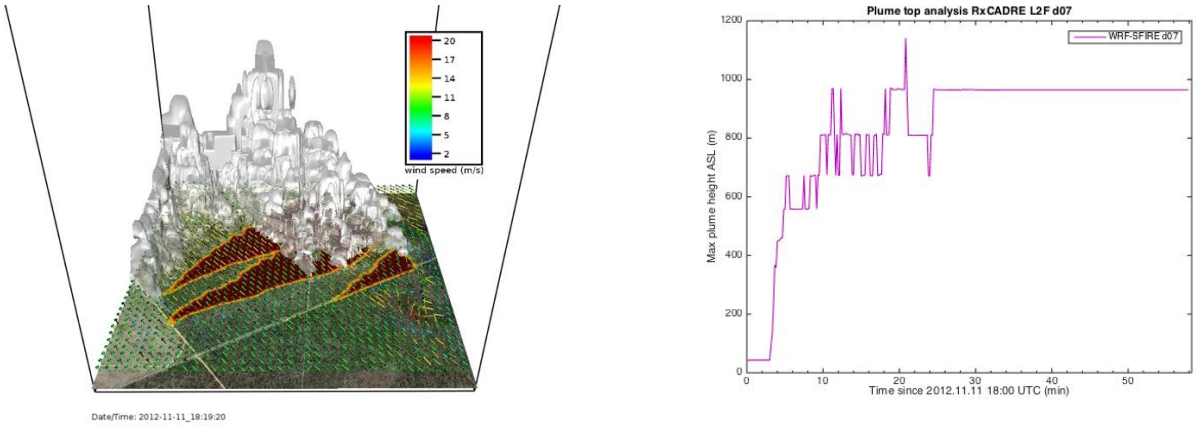


Figure 18. a) A snapshot from the RxCADRE L2F simulation with 3 ATV ignition lines, showing 10m winds color-coded according to the wind speed, fire area (dark red fill) and 3D volume rendered tracer representing smoke, b) time series of the plume top height simulated in fire domain (d07).

4.3. Sensitivity Study of Sensor Placement

We have performed a sensitivity study following the rLHS with variance decomposition methodology, described in Section 3.4. We have chosen the Fishlake simulation standard case from Section 4.2.2 as the base case.

4.3.1. Sampling Setup

We have chosen L=7 simulation parameters, which were:

1. 10h-fuel moisture content, varying from 0.04 to 0.14 water mass/dry fuel mass. 1h-fuel fuel moisture = 10h-fuel moisture – 0.01, 100h-fuel moisture = 10h-fuel moisture + 0.01, 1000h fuel moisture was 0.05, and live fuel moisture was 0.78. These values were entered as initial moisture values and did not change with time. See Mandel et al. (2012, 2014) for a further description.
2. Heat extinction depth, varying from 6m to 50m. The fire heat flux is entered into WRF boundary layer with exponential decay, rather than all into the bottom layer of cells. The heat is apportioned depending on the height of the cell center above the terrain, with weight 1 at the ground and $1/e$ at the heat extinction depth. This gradual heat insertion is a parameterization for unresolved mixing and radiative heat transfer.
3. Heat flux multiplier, varying from 0.5 to 2. The heat flux multiplier was chosen as a measure of the fire effect on the atmosphere; unlike the fuel load, it does not influence the rate of spread.
4. Rate of Spread (ROS) multiplier for the omnidirectional ROS component
5. ROS multiplier for the wind-induced component
6. ROS multiplier for the slope-induced component
7. The simulation day, selected from the "typical" burn days

The first 6 parameters were constant over the whole fire simulation domain. Since the 6 parameters are positive, first they were transformed by taking the logarithm and then a Gaussian distribution centered within the transformed interval was determined so that 90% probability was between the lowest and the highest value. The transformed interval was then divided into $N=5$ subintervals of equal probability, sampling points chosen as the middle probability values, and transformed back into the original physical space. The resulting sampling values are in Table 5.

Table 5: Sampling points for Latin Hypercube Sampling

Sampling point	1	2	3	4	5
10-fuel moisture (kg/kg)	0.0459	0.0613	0.0748	0.0914	0.1219
Heat extinction depth (m)	7.5830	12.3531	17.3205	24.2854	39.5620
Heat flux multiplier (-)	0.5827	0.8017	1.0000	1.2473	1.7161
Multiplier for the omnidirectional ROS component (-)	0.5827	0.8017	1.0000	1.2473	1.7161
Multiplier for the wind-induced ROS components (-)	0.5827	0.8017	1.0000	1.2473	1.7161
Multiplier for the slope-induced ROS component (-)	0.5827	0.8017	1.0000	1.2473	1.7161
Simulation day, selected from the "typical" burn days (-)	1 09.03.2014	2 09.11.2016	3 09.22.2012	4 09.26.2015	5 09.27.2015

Total r=200 repetition was done, resulting in rN=1000 simulations. The selection of the sample points in the first replicant is in Table 7.

For the purpose of the sensitivity analysis, the output Y of the model is the value of a quantity of interest at a node of the simulation mesh on the Earth's surface. We then display the values associated with the quantity of interest, such as the VCM (Variance of the Conditional Expectation being an estimate of the variability of the output due to the parameter change) as a function of location on the Earth surface.

We consider the following quantities of interest:

1. The vertical velocity vector component w, interpolated to a given height above the terrain, or to a given altitude above the sea level.
2. The smoke intensity (the concentration of WRF tracer `tr17_1`), interpolated to a given height above the terrain, or to a given altitude above the sea level.
3. Plume top height, derived from the smoke concentration

Note that in WRF, the vertical position of grid nodes is not fixed, rather it is derived from the geopotential height, which is a part of the solution, and changes with time.

Because some extreme values of the sampling parameters caused instability in the WRF surface scheme during and immediately after ignition (when the updraft carrying the heat away is not yet established), we have simulated the initial 30 minutes when the ignition occurs using the standard case and only then switched to the modified parameter values. We show the results

from simulations running for additional 7.5 hours, during which the model state was captured and analyzed at hourly intervals. The resultant 7 time frames were averaged to provide a picture corresponding to the 7h of the burn.

4.3.2. Computational results for the Fishlake burn simulation

In the figures below we show visualizations of selected results of the repeated Latin Hypercube Sampling (rLHS) and the analysis of variance described above. The vertical level of highest variances in the updraft velocities (1200m above the ground) have been selected for visualization based on the vertical velocity variance time series plotted in left panel of Figure 19. Based on a similar time series of the smoke concentrations and the plume top height estimates form the baseline case (Figure 11), the analysis level for smoke has been set to 1400m. The right panel in Figure 19 shows the variance of the vertical wind velocity at 1200m above the terrain, computed from the 25 rLHS-sampled simulations. Two vertical well-defined vertical velocity maxima are located outside of the northwestern plot boundary, indicating optimal locations for upper-level vertical velocity measurements. Local variance reaches up to 3 m^2/s^2, which corresponds to local deviations in the vertical velocity of about ±1.7 m/s between the sampled simulations, which is high enough to be sampled by a Doppler Lidar, Radar or Sodar. However, the fact that the updraft variances are concentrated over small regions indicate that optimal placement of the measurement devices may play a critical role in constraining model parameters based on vertical velocity observations.

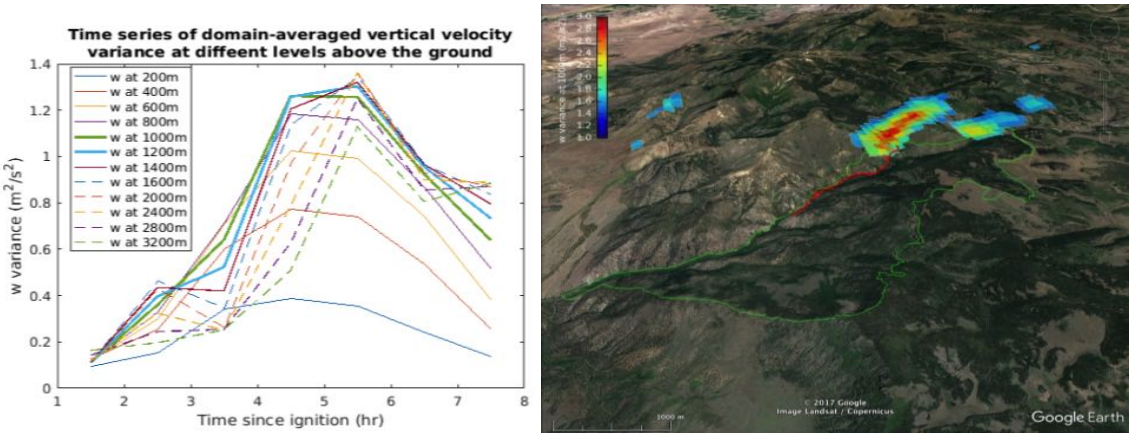


Figure 19. Left: The time series of the variance of vertical velocity at different heights above the ground from simulations executed with parameters from the rLHS sampling. Right: the map of the variance of the vertical velocity at 1200m with the indication of the burn plot (green contour) and the ignition (red line).

In Figure 21, we show maps of the mean and the normalized variance of the plume top height. The former informs about the expected vertical plume extent which should be taken into account when the plume sampling strategies are considered. The latter shows regions of most pronounced impact of the model parameters on plume rise. Interestingly, the region directly downwind from the fire where the plume reaches highest elevations is not the same as the region of highest plume height average variance, shown in Figure 20. In all computations reported here, "average variance" is computed for each simulation day separately, and then averaged over the simulation days. The plume height average variance exhibits high values over two symmetrical regions located on both sides of the plume axis and are pushed much more downwind from the fire

compared to the mean plume top height. These results indicate that while the localized vertical smoke sampling (corkscrew path) may be optimal for assessing the maximum plume top height, the sampling region must be significantly extended downwind in order to adequately sample the plume top height variance. Figure 20 shows maps for the mean smoke concentration at 1400m (left panel), and the smoke concentration variance (right panel). The mean smoke concentration is intended to inform where the probability of successful smoke sampling is the highest. The variance on the other hand, shows where the tested model parameters have strongest impact on the smoke concentrations. It is noteworthy that high values of smoke concentrations variances are found over much larger region than the vertical velocity variance shown in right panel of Figure 19, which suggest that vertical velocity measurements are more suitable for local platforms like ground lidars, while the smoke measurements could be performed from airborne platforms covering larger area.

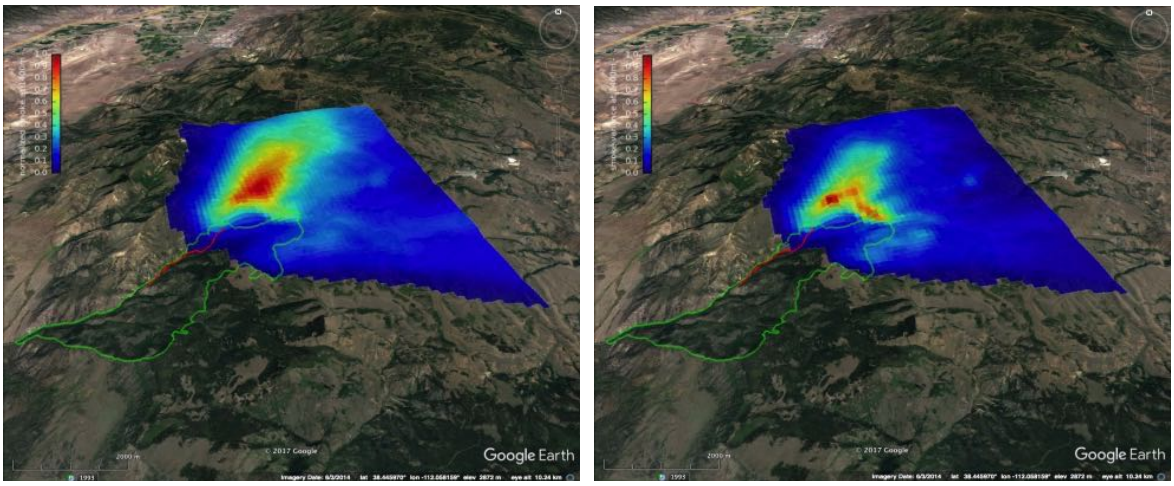


Figure 20. Left: The mean normalized smoke concentration at 1400m above the terrain, Right: The normalized smoke concentration average variance at 1400m above the terrain. The green contour represents the burning plot boundary, the red line shows the ignition.

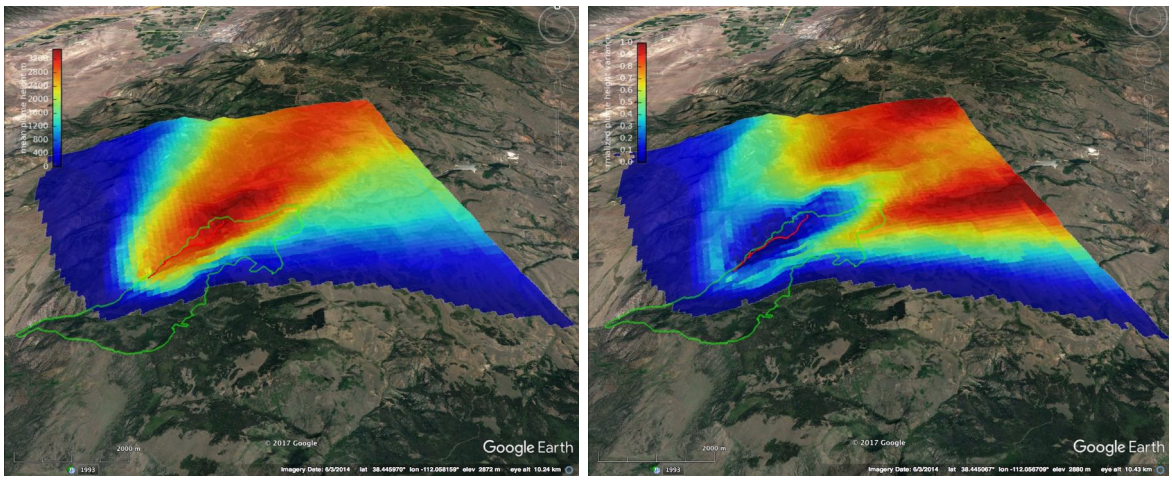


Figure 21. Left: The mean plume top height, Right: The normalized variance in the plume height. The green contour represents the burning plot boundary, the red line shows the ignition.

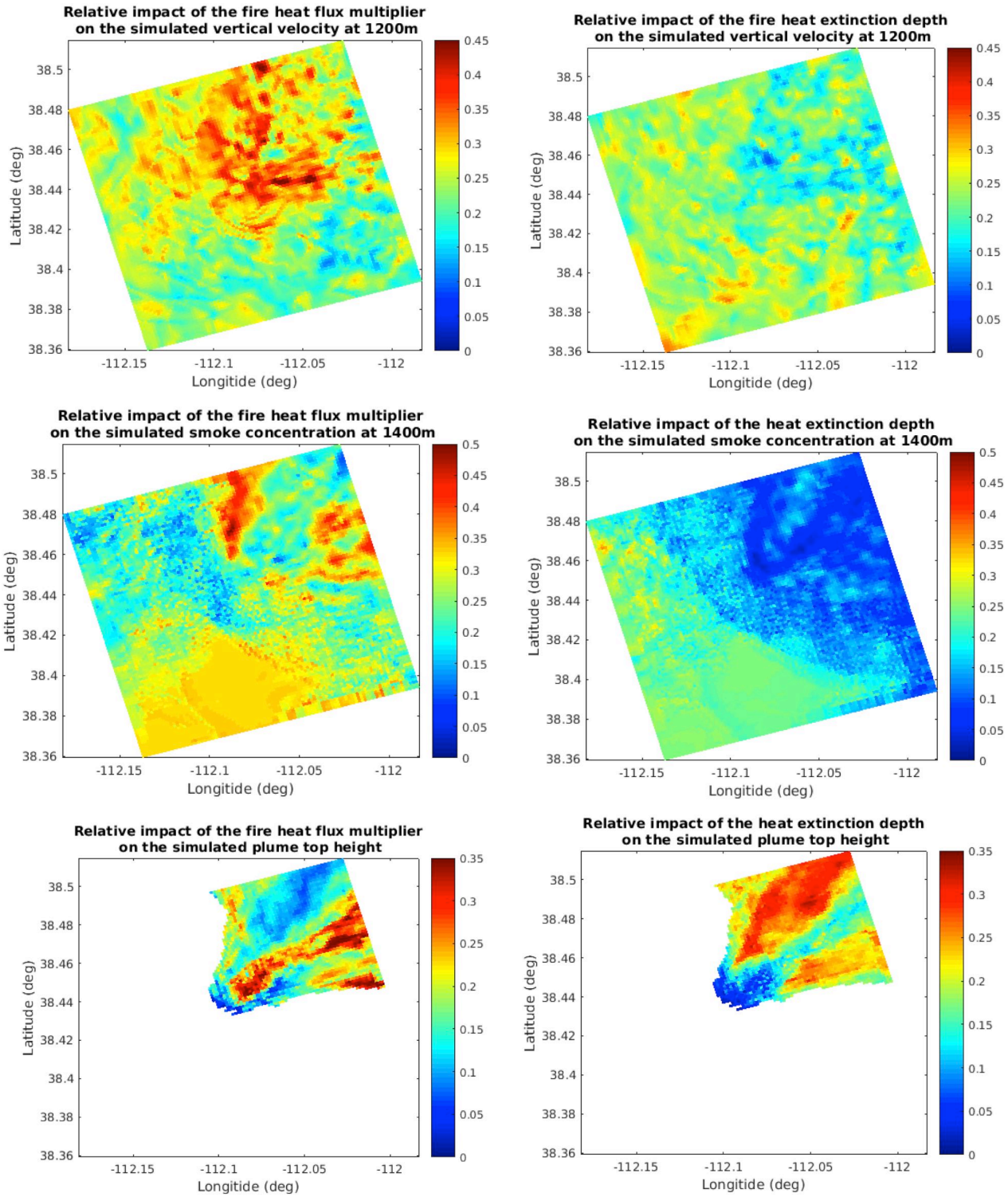


Figure 22. Contribution of the fire heat flux multiplier (left column) and the heat extinction depth (right column) on the vertical velocity at 1200m (top), smoke concentration at 1400m (middle), and plume top height (bottom).

The analysis of the rLHS, described in previous sections, not only informs where the measurement should be taken in order to constrain model parameters, but also allows to find relative contribution of the tested parameters to variances in the variables of interest. In figures

below we present results of the first-order variance decomposition of vertical velocity at 1200m, smoke concentration at 1400m, and the plume top height, attributed to the most important simulation parameters. These spatial plots of the sensitivity indices inform about the relative importance of the tested parameters (such as fuel moisture, heat extinction depth, fire heat flux, etc.) as well as indicate regions of highest impact of these parameters on the variables of interest. The most critical parameters controlling the vertical velocity, smoke concentration and the plume top height are shown in Figure 22. The heat flux multiplier contributes to the variance of vertical velocity, smoke concentration and the plume top height, in up to 50%. That is to be expected, as the fire heat flux plays the major role in driving the plume rise, and indicates the importance of detailed the fire heat characterization during experimental burns. The second largest contribution (up to 40%) comes from the heat extinction depth defining the depth over which the fire heat flux is distributed vertically in the model. The extinction depth seems to have similar magnitude-wise impact on the variables of interest like the heat flux multiplier, but its character is different. Looking at Figure 22 one can notice that the heat flux multiplier has in general opposite effect than the extinction depth. For instance, the vertical velocities are impacted by the heat flux multiplier mostly downwind (north east) from the fire (see left top panel). The extinction depth on the other hand affects mostly vertical velocities upwind from it (see right top panel). Similarly, the smoke concentration is impacted by the heat flux in the north-eastern part of the domain, while the extinction depth seems to impact mostly the south-western part. Similar situation can be observed in the bottom panels of Figure 22 showing the impact of these parameters on the plume top height. Here the heat flux impact is confined to the very core of the plume, while the extinction depth has most pronounced impact on the sides of the plume. The differences in these patterns indicate that the placement of the sensor optimal form the standpoint of constraining the heat flux may be not optimal for constraining the extinction depth and vice-versa. Also, the patterns of the heat flux contribution seem generally more organized that the ones of the extinction depth. The impact of the extinction depth seems to have non-local character especially when vertical velocity and smoke concentration is considered. spreading large areas upwind from the fire. That suggests that estimating the extinction depth through the observations of vertical velocities and smoke concentration may be difficult, and may require direct tower-based observations of the vertical attenuation of the fire heat flux.

The analysis presented above, focuses on the impact of model parameters on measurable variables like vertical velocity or plume height. In that sense, it informs where the signal coming from the changes in the parameters is expected to be highest and consequently, shows locations where measurements should be taken to detect it. As the weather conditions on the burn day cannot be predicated on the moment of experimental planning, the analysis presented in Section 3.1 has been employed to find most typical historical days for all the burn stations. These days have been included in this analysis so on the top of the baseline simulation, performed for 09.03.2014, analogous runs have been executed for other 4 typical days listed in Table 4. These 5 days are treated as an ensemble of most typical scenarios for the Fishlake experimental burn. Below we show means and average variances of the variables of interest computed across the simulations performed for these days. Figure 23 shows maps of mean and variance of the smoke concentration at 1400m. A pattern can be noticed there, indicating general north-northwest smoke dispersion. The fact that regions of maxima in both mean concentration and the variance can be easily identified, indicates that the typical days are similar to each other in terms of a general flow pattern in the Fishlake region. It should be noted that wind direction wasn't a part of burn requirements for this burn site which defined only the wind speed, temperature and relative

humidity. The fact that the typical days show relatively consistent flow pattern indicates that days typical in terms of the air humidity, wind speed, and moisture, are associated with a south-southwesterly flow.

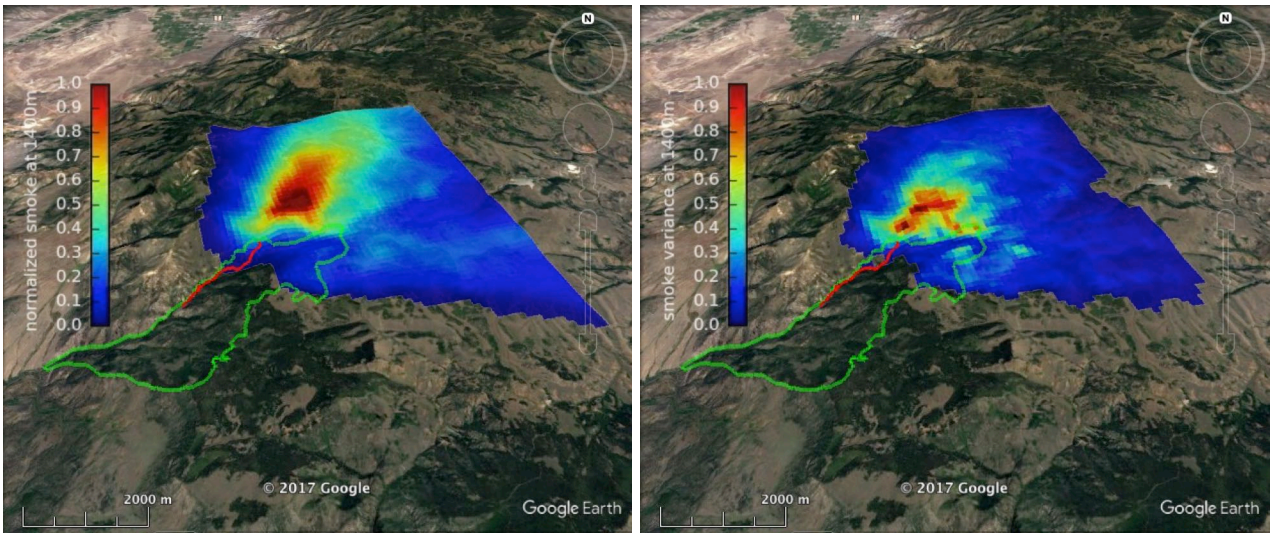


Figure 23. Left: The mean normalized smoke concentration at 1400m above the ground computed from 5 most typical days, Right: The normalized average variance of the smoke concentration at 1400m computed from 5 most typical days. Green contour represents burn unit boundaries, red line shows the ignition.

The maps of the vertical velocity variance and the mean plume top height are shown in Figure 24. A clear maximum in the updraft fluctuations is evident in north-west from the end of the ignition line, as well as well-defined north-northeasterly plume reaching 3200m. The statistical analysis of simulations performed for most typical day such as the one shown in Figure 23 and Figure 24, are intended to be used as a guideline in the planning stages of experimental burns when optimal locations of smoke and vertical velocity measurements are being considered.

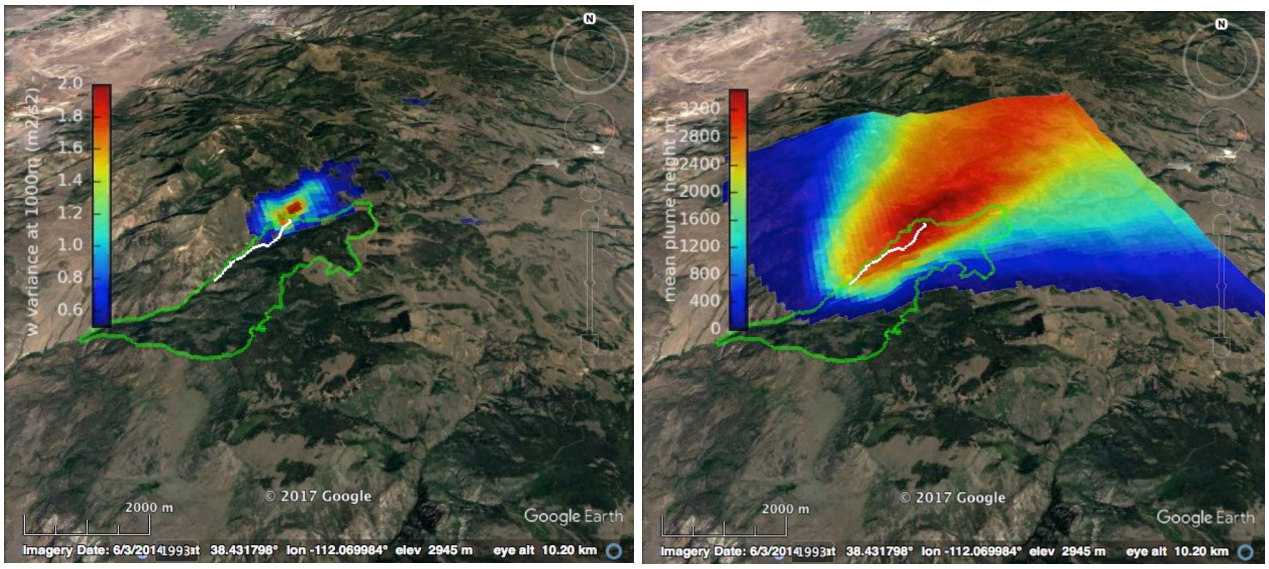


Figure 24. Left: The vertical velocity average variance at 1000m computed from 5 most typical days, Right: The mean plume top height computed from 5 most typical days. Green contour represents burn unit boundaries, white line shows the ignition.

5. Conclusions (Key Findings) and Implications for Future Research

5.1. Estimation of typical days for simulation initialization

If the meteorological state vector $\boldsymbol{x}$ had a multivariate Gaussian distribution, then the most typical day as defined in this work would also be the day with the most probable state. As Figure 7 shows, a very typical day need not occur very often; rather, it is a day when the state is in a rigorous sense very close to its sample-mean. Indeed, for a non-Gaussian distribution (very likely the case here), a day with state close to its mean need not occur very often, even approximately. From this standpoint, the analysis performed here assures that statistically, any historical day meeting the burn requirements will be similar to the identified typical day, which fits the purpose of this work. In future work, an alternative to consider would be to pick a state (and thus, the day it occurs) randomly from a multivariate bin containing states that do occur simultaneously very often.[6] But before following this alternative, one would have to choose among many competing theories about how to choose the sizes of such a bin, robustly. The definition of "typical" used here at least is non-parametric and in that sense, objective.

5.2. Results from Numerical Experiments

The numerical experiments presented in this study were performed and analyzed to aid with the preparation of the FASMEE experimental plan. Note that, as we do not know how similar the typical burn days are to the actual burn days, the presented results cannot provide deterministic forecasts, but examples only of possible realizations of the planned experimental burns. Also, as reported above, the model used to perform these simulations has important limitations. It cannot be validated fully as a smoke model before FASMEE provides the needed observational data. Despite model limitations, the results presented do offer valuable insight and guidance to FASMEE's field measurement campaign.

The results showing model sensitivity to resolution indicates that local updraft velocities may be significantly underestimated in the model simulations when the model grid is not fine enough to resolve small-scale plume features. The vertical velocities observed in the simulations should be treated as lower limits for the expected values. These velocities suggest that measurement platforms used during the experiments should be robust enough to sustain updrafts exceeding 10 m/s. Simulation suggests that the strongest updrafts will be observed in the Fishlake burn. However, due to the influence of complex local topography, exact identification of the fire impact on the local circulation may be difficult. From this standpoint, the Fort Stewart burn is possibly a great complementary burn, providing a cleaner picture of the fire-induced circulation.

---

[6] Mathematically, such an often-occurring state would maximize the multivariate conditional histogram $h(\boldsymbol{x}) = \sum_{n=1}^{N} B(\mathscr{A}(\boldsymbol{x}_n)) \prod_{j=1}^{5} B(-\Delta_j/2 \le x_{n,j} - x_j < \Delta_j/2)$ (see Appendix A for notation).

One of the most important aspects of the experimental burns is their durations. In the light of the simulations for the western sites, the burn experiments should last at least 5h in order to fully capture the fire plume evolution, and consequently the burn plots should be big enough to enable fire progression over this period of time. In the small experimental burns, the fire may reach the end of the burn plot before the plume fully evolves. The preliminary simulation of the RxCADRE L2F shows such situation. The plume top height becoming constant after 26min into the simulation (see Figure 18 a) does not indicate that the plume reached a quasi-equilibrium state; rather, the fire consumed all the fuel available in the burn plot.

From the standpoint of the tower measurements, in order to characterize the flow near the pyro convective column, winds and turbulent fluxes should be sampled at least up to 25m AGL. Short 6.1m towers are probably not high enough to sample the fire-induced inflow into the plume column.

Unlike the fire-induced horizontal winds expected to reach maximum speeds relatively close to the ground, maximum vertical velocities are expected at significantly higher AGL. Simulations of the experimental burns indicate strongest updrafts between 150m and 3000m AGL, so Doppler Lidars and other platforms providing vertical wind measurements should be located and employed ways that provide sufficient vertical coverage.

In terms of smoke sampling, the simulations suggest that the vertical smoke concentration profiles may differ significantly between the experimental burns. The Fishlake burn of expected highest heat and emission fluxes is associated with an elevated concentration peak located at about 1000m above the ground. The plume top heights also vary, from 1800m for the Fort Stewart, 2200m for North Kaibab to 2700m for Fishlake. From this standpoint, it seems beneficial to adjust the flight height accordingly in order to provide optimal plume sampling.

Numerical simulations of the Fort Stewart burn indicate strong sensitivity of the initial plume evolution to the fire ignition pattern. Therefore, from the standpoint of model validations, a detailed characterization of the ignition procedure is necessary. Also, as the fire initialization is crudely implemented in the model, interactions between simultaneously ignited fire lines are difficult to capture. Therefore, in experimental burns, simple ignitions are preferable, especially in cases when continuous IR observations of sufficient temporal and horizontal resolution are not available. Similarly, the sensitivity of the simulated plume top heights and vertical velocities to the fire heat flux (demonstrated based on the Fishlake simulations), indicates that detailed characterization of fire heat fluxes (radiative and convective) is critical to model evaluation. Not only radiative, but also convective heat fluxes from the fire must be observed during the experimental burns.

5.3. Sensitivity Study of Sensor Placement

The results presented in section 4.3 illustrate how numerical modeling combined with advanced statistical methods can objectively guide planning of experimental burns. There are multiple possible applications of this method in experimental planning. If an experiment is performed with the intention to constrain certain model parameters, such analysis can help assess feasibility of this approach, and help designing optimal sampling strategy. In the analyzed Fishlake burn for instance, the results indicate that assessing the extinction depth through the vertical and smoke concentration measurements may be not be feasible, which leads to a recommendation of direct measurements of the fire heat flux at various heights. This method also allows to identify the

most critical model parameters that have most pronounced effect on plume dynamics. Here the heat flux has been identified as the one of greatest importance from the plume development standpoint, which emphasizes the importance of accurate characterization of fire heat fluxes. As this method utilizes the results of statistical analysis of typical burn days it also informs about expected local flow pattern, plume dispersion and fire behavior in a probabilistic sense. If the decisions about sensor placements have to made in advance, such analysis may guide the sensor placement and flight paths or when combined with local constraints in terms of road access, filed of view etc. may help define best locations for experimental burns. In case of the Fishlake burn this analysis indicates that generally the area north-northeast from the burn plot should be considered for placement of measurement platforms.

We have performed only a pilot analysis of variations for several sampled variants of the Fishlake experimental burn. Here are some preliminary observations and implications for future research:

- Statistical analysis of runs with carefully sampled parameter sets was shown to be provide clear guideline on placing the measurements in space and time
- In addition, some measurements were found more affected by certain parameters, which can inform what parameters can be indirectly constrained by observations
- Analyses of the variance in smoke concentration vertical velocity and plume top height from runs executed for different days and with different parameter sets show similar patterns which indicates that typical day statistics managed to identified statistically similar days from the standpoint of plume rise and dispersion.
- The variability of the plume at higher altitudes (here, 1000-1400m above the ground) is concentrated up to couple kilometers downwind from the plot, which suggests that sampling right above the fire optimal from the fire heat flux measurement standpoint may not be optimal for sampling most vigorous parts of the plume.
- Additional parameters can be considered. The cost of one repetition does not increase with the number of parameters, but more repetitions need to be done for statistical convergence. The variability of the outputs will increase and the added parameters will help model additional uncertainty, which is always present in reality.

The computations reported here consisted of 1,000 simulations executed in about a week to reduce the sampling error. The wall clock time of one Fishlake simulation is about 2 hours on 3600 NCAR cores, and it produces about 15GB of data. Counting some repetition, e.g., for runs failed for various reasons, for development, and for experiments, about 1,000,000 core hours and 50TB of disk space was needed for the sensitivity analysis, with limited room for experiments. The statistical processing was done on an auxiliary Geyser cluster in MATLAB. It was performed in serial mode which significantly contributed to the total analysis time. Overall, several thousands of hours on the Geyser cluster were used.

As multiple simulations can be run concurrently, we suggest to perform similar analysis prior to the burn using available forecast data. With sufficient supercomputer allocation, this is feasible.

6. Literature Cited

Briggs, G. A., 1969: Plume rise. AEC Critical Review Series TID-25075, National Technical Information Service.

— 1971: Some recent analyses of plume rise observation. *Proceedings Second International Clean Air Congress*, Academic Press, 1029–1032.

— 1972: Discussion - Chimney plumes in neutral and stable surroundings. *Atmospheric Environment*, **6**, 507–510, doi:10.1016/0004-6981(72)90120-5.

Chugunov, N., T. S. Ramakrishnan, N. Moldoveanu, A. Fournier, K. S. Osypov, and Y. B. Altundas, 2014: Targeted survey design under uncertainty. United States Patent Application Publication US 2014/0278110 A1.

Clements, C. B., S. Zhong, S. Goodrick, J. Li, B. E. Potter, X. Bian, W. E. Heilman, J. J. Charney, R. Perna, M. Jang, D. Lee, M. Patel, S. Street, and G. Aumann, 2007: Observing the dynamics of wildland grass fires – FireFlux – a field validation experiment. *Bull. Amer. Meteorol. Soc.*, **88**, 1369–1382, doi:10.1175/BAMS-88-9-1369.

Hudak, A. T., M. B. Dickinson, B. C. Bright, R. L. Kremens, E. L. Loudermilk, J. J. O'Brien, B. S. Hornsby, and R. D. Ottmar, 2015: Measurements relating fire radiative energy density and surface fuel consumption – RxCADRE 2011 and 2012. *International Journal of Wildland Fire*, **25**, 25–37, doi:10.1071/WF14159.

Kochanski, A. K., M. A. Jenkins, J. Mandel, J. D. Beezley, C. B. Clements, and S. Krueger, 2013: Evaluation of WRF-SFIRE performance with field observations from the FireFlux experiment. *Geoscientific Model Development*, **6**, 1109–1126, doi:10.5194/gmd-6- 1109-2013.

Kochanski, A. K., M. A. Jenkins, K. Yedinak, J. Mandel, J. Beezley, and B. Lamb, 2016: Toward an integrated system for fire, smoke, and air quality simulations. *International Journal of Wildland Fire*, **25**, 534–546, doi:10.1071/WF14074.

Linn, R., J. Reisner, J. J. Colman, and J. Winterkamp, 2002: Studying wildfire behavior using FIRETEC. *Int. J. of Wildland Fire*, **11**, 233–246, doi:10.1071/WF02007.

Mandel, J., S. Amram, J. D. Beezley, G. Kelman, A. K. Kochanski, V. Y. Kondratenko, B. H. Lynn, B. Regev, and M. Vejmelka, 2014: Recent advances and applications of WRF-SFIRE. *Natural Hazards and Earth System Science*, **14**, 2829–2845, doi:10.5194/nhess-14-2829-2014.

Mandel, J., J. D. Beezley, and A. K. Kochanski, 2011: Coupled atmosphere-wildland fire modeling with WRF 3.3 and SFIRE 2011. *Geoscientific Model Development*, **4**, 591– 610, doi:10.5194/gmd-4-591-2011.

Mandel, J., J. D. Beezley, A. K. Kochanski, V. Y. Kondratenko, and M. Kim, 2012: Assimilation of perimeter data and coupling with fuel moisture in a wildland fire – atmosphere DDDAS. *Procedia Computer Science*, **9**, 1100–1109, doi:10.1016/j.procs.2012.04.119, Proceedings of ICCS 2012.

McKay, M. D., 1995: Evaluating prediction uncertainty. Technical Report NUREG/CR- 6311 LA-12915-MS, Los Alamos National Laboratory, http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/26/051/26051087.pdf retrieved October 16, 2016.

McKay, M. D., R. J. Beckman, and W. J. Conover, 1979: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**, 239–245, doi:10.2307/1268522.

Mell, W., M. A. Jenkins, J. Gould, and P. Cheney, 2007: A physics-based approach to modelling grassland fires. *Intl. J. Wildland Fire*, **16**, 1–22, doi:10.1071/WF06002.

Mesinger, F., G. DiMego, E. Kalnay, K. Mitchell, P. C. Shafran, W. Ebisuzaki, D. Jovic´, J. Woollen, E. Rogers, E. H. Berbery, M. B. Ek, Y. Fan, R. Grumbine, W. Higgins, L. Hong, L. Ying, G. Manikin, D. Parrish, and W. Shi, 2006: North American regional reanalysis. *Bulletin of the American Meteorological Society*, **87**, 343–360.

Regev, B., S. Amram, and A. Amit, 2012: Six hours ahead - predicting wild- fires in real time. *Innovation Exchange*, **16**, 34–37, available at http://mops.gov.il/English/HomelandSecurityENG/NFServices/Pages/FirePredictionSystem.aspx, retrieved September 2013.

Saltelli, A., S. Tarantola, F. Campolongo, and M. Ratto, 2004: *Sensitivity analysis in practice*. John Wiley & Sons, Ltd., Chichester, xii+219 pp.

Sobol′, I. M., 2001: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simulation*, **55**, 271–280, doi:10.1016/S0378-4754(00)00270-6.

The ”Matash” Fire-Prediction system, 2017: http://mops.gov.il/English/HomelandSecurityENG/NFServices/Pages/FirePredictionSystem.aspx, accessed May 14, 2017.

Urbanski, S. P., 2014: RxCADRE 2012: Airborne measurements of smoke emission and dispersion from prescribed fires. Forest Service Research Data Archive, Fort Collins, CO, doi:10.2737/RDS-2014-0015.

Wikipedia contributors, 2017: Dew point. Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Dew_point. accessed April 26, 2017.

Appendix A: Technical formulation of the statistical analysis

As summarized in section 3.1.1, the single-variable case suggests, given N samples $\{T\}_{n=1}^{N}$ of a random temperature-value T, to define a "typical" T with respect to its sample-mean value

$$\langle T\rangle = N^{-1}\sum_{n=1}^{N} T_n$$

and its sample-variance (square of standard deviation)

$$\sigma^2 = \langle (T - \langle T\rangle)^2\rangle.$$

That is, "typical" would mean a small value of the number

$$\delta = |T - \langle T\rangle|/\sigma$$

that is itself random with $\langle\delta\rangle = 0$ and $\langle\delta^2\rangle = 1$. The statistical $\langle\cdot\rangle$ operator is modified below, to include masking by a binary function

$$B(\mathscr{C}) \in \{0,1\}$$

denoting the Boolean value of the true-or-false criterion $\mathscr{C}$ that encodes acceptable conditions for burn operations.

The full data used in the present analysis comprise T conjoined with:

- relative humidity $0 \leq \phi < 1$;
- horizontal wind speed $s \geq 0$;
- wind direction $0° \leq d < 360°$; and
- wind gust $g > 0$.

To standardize the value ranges to[7] $(-\infty,\infty)$, transform from (ϕ,s,d,g) to (T_d,u,v,ℓ), where:

- T_d is dew-point temperature[8];

---

[7] Obviously in practice, $(-\infty,\infty)$ indicates a finite but sign-indefinite value range.

[8] The formula used here is

$$T_\mathrm{d} = \big(B(\phi \neq 0)/\big(b/(\ln\phi + (b - T/234.5\mathrm{K})T/(T + c)) - 1\big) - B(\phi = 0)\big)c,$$

where

$$b = B(T \geq 0°\mathrm{C})17.368\mathrm{K} + B(T < 0°\mathrm{C})17.966,$$

$$c = B(T \geq 0°\mathrm{C})238.88\mathrm{K} + B(T < 0°\mathrm{C})247.15\mathrm{K}$$

(Wikipedia contributors 2017). The inverse formula is

$$\phi = B(T_\mathrm{d} \neq -c)\exp(b/(c/T_\mathrm{d} + 1) - (b - T/234.5\mathrm{K})/(c/T + 1)).$$

See Matlab function `RELH2DWT.m`.

- $(u,v) = -(\sin d, \cos d)s$ is horizontal wind vector; and
- $\ell = \log_{10} g$ is gust logarithm.

Also, there needs to be a generalized re-definition of:

- state from a number T to a 5×1 column vector
  $$\boldsymbol{x} = (T, T_\mathrm{d}, u, v, \ell)^\mathrm{t};$$
- sample variance from a number σ^2 to a 5×5 sample covariance matrix
  $$\boldsymbol{\Sigma\Sigma}^\mathrm{t} = \langle(\boldsymbol{x} - \langle\boldsymbol{x}\rangle)(\boldsymbol{x} - \langle\boldsymbol{x}\rangle)^\mathrm{t}\rangle$$
  (product of inverse-standardizing matrix $\boldsymbol{\Sigma}$ with its transpose); and
- standardized difference from a number $|T - \langle T\rangle|/\sigma$ to a 5×1 standardized deviation vector[9]
  $$\boldsymbol{\delta} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \langle\boldsymbol{x}\rangle),$$
  each of whose rows is still unitless (because the $\boldsymbol{x}$ rows and covariance entries have mixed units that mutually compensate each other) with null sample-mean vector $\langle\boldsymbol{\delta}\rangle = \boldsymbol{0}$ and identity covariance-matrix $\langle\boldsymbol{\delta\delta}^\mathrm{t}\rangle = \mathbf{I}$.

The well-known Mahalanobis deviation norm is the root-sum-of-squares

$$\|\boldsymbol{\delta}\| = (\boldsymbol{\delta}^\mathrm{t}\boldsymbol{\delta})^{1/2}$$

over $\boldsymbol{\delta}$-vector entries. Its second sample moment $\langle\|\boldsymbol{\delta}\|^2\rangle = \dim \boldsymbol{x} = 5$ exactly, by construction. In interpreting Figure 7 it is useful to note that if $\boldsymbol{x}$ would have a multivariate Gaussian probability distribution, then $\|\boldsymbol{\delta}\|$ would have a χ_5 or "chi" probability distribution with 5 degrees of freedom, mean $(128/9\pi)^{1/2} \approx 2.13$ and standard deviation $(5 - 128/9\pi)^{1/2} \approx 0.688$.

As mentioned above, the sample mean operation $\langle\cdot\rangle$ should be modified to accommodate requirements for a state $\boldsymbol{x}_n$ at sample index (time) n being acceptable for a burn operation:

$$\langle\boldsymbol{x}\rangle = N_{\mathscr{C}}^{-1} \sum_{n=1}^{N} \boldsymbol{x}_n \, B(\mathscr{C}(\boldsymbol{x}_n)),$$

where:

- $\mathscr{C}(\boldsymbol{x})$ is true or false as $\boldsymbol{x}$ does or does not meet all the criteria; and
- the closer
  $$N_{\mathscr{C}} = \sum_{n=1}^{N} B(\mathscr{C}(\boldsymbol{x}_n))$$
  approaches to its upper bound N, the more reliable the statistical estimate is.

$N_{\mathscr{C}}$ cannot exceed the least of the enumerations in the legends of Figure 5 and similar figures. In this work, the criteria are logical conjunctions of lower and upper bounds for the data T, ϕ, s for each potential burn location and window[10] i:

$$\mathscr{C}_i(\boldsymbol{x}) = (T_{li} \le T \le T_{ui}) \;\&\; (\phi_{li} \le \phi \le \phi_{ui}) \;\&\; (s_{li} \le s \le s_{ui}),$$

---

[9] See item 7.b below. $\boldsymbol{\Sigma}^{-1}$ need not be symmetric.

[10] Note that i indexes all of the station name, days in window, and hours in window per day.

as enumerated in Table 6 representing burn requirements for the analyzed burn sites.

Table 6. Stations, time windows (mm/dd hh:00UTC) and value limits for potential burns (burn windows).

i		T_{li}	T_{ui}	ϕ_{li}	ϕ_{ui}	s_{li}	s_{ui}
FSHU1	9/ 1 16: 00UTC - 10/ 31 18: 00UTC	61°F	85°F	16%	22%	0	15mi h^{-1}
KCWW	2/ 20 16: 00UTC - 2/ 28 18: 00UTC	60	90	30	55	6mi h^{-1}	20
	4/ 15 13: 00UTC - 4/ 30 14: 00UTC	60	90	30	55	6	20
KLHW	2/ 20 16: 00UTC - 2/ 28 18: 00UTC	60	90	30	55	6	20
	4/ 15 13: 00UTC - 4/ 30 14: 00UTC	60	90	30	55	6	20
LCSS1	1/ 1 15: 00UTC - 2/ 1 19: 00UTC	60	90	30	55	6	20
	12/ 1 15: 00UTC - 12/ 31 19: 00UTC	60	90	30	55	6	20
QLBA3	9/ 1 17: 00UTC - 10/ 31 19: 00UTC	61	85	16	22	0	15
TT084	9/ 1 16: 00UTC - 10/ 31 18: 00UTC	61	85	16	22	0	15
i–burn season, T – temperature, ϕ – relative humidity, s – wind speed							

The main Matlab script for the statistical analysis is[11] `analXls.m`, which performs the following tasks.

1. Read burn-requirements criteria $\mathscr{C}_i(\boldsymbol{x})$ from file `bReqs.csv`.
    a. Convert T_{li}, T_{ui} to °C, and s_{li}, s_{ui} to m s^{-1}.
2. Read the historical weather-station (T,ϕ,s,d,g) data from a CSV file for each station.
    a. Delete extreme[12] $T < -63$°C or $T > 57$°C.
    b. Delete $\phi < 0$ or $\phi > 1$.
    c. Delete $g < 0$ or $g > 103$ m s^{-1}.
    d. Delete time values corresponding to deleted state values.
3. Call `analXls_figs.m` to produce Figure 5 and similar figures.
4. Transform from (T,ϕ,s,d,g) to $\boldsymbol{x} = (T,T_\mathrm{d},u,v,\ell)$ as described above.
5. As described above, compute the statistics of acceptable data:
    a. data count $N_{\mathscr{C}}$;
    b. 5×1 mean-state vector $\langle \boldsymbol{x} \rangle$;

[11] The script name stood for analyze Xls spreadsheets, before CSV spreadsheets were adopted.

[12] Extreme values chosen from dubious values actually occurring in CSV files.

c. sequence of 5 variances $\{\langle(x_j - \langle x_j\rangle)^2\rangle = \sigma_j^2 = \sum_{k=1}^{5} \Sigma_{jk}^2\}_{j=1}^{5}$;
d. sequence of 10 correlations
$$\{\{\rho_{jl} = \sigma_j^{-1}\sigma_l^{-1}\big(\langle(x_j - \langle x_j\rangle)(x_l - \langle x_l\rangle)\rangle = \sum_{k=1}^{5} \Sigma_{jk}\Sigma_{lk}\big)\}_{l=j+1}^{5}\}_{j=1}^{4}.$$

6. Produce Figure 6 ...
   a. Represent $\mathbf{\Sigma\Sigma}^t$ by ellipses (see `covEllip.m` and the caption of Table 6).
7. Call `analXls_Mdists.m` to compute:
   a. 15 covariances $\{\{\sum_{k=1}^{5} \Sigma_{jk}\Sigma_{lk}\}_{l=j}^{5}\}_{j=1}^{5}$ from $\{\sigma_j\}_{j=1}^{5}$ and $\{\{\rho_{jl}\}_{l=j+1}^{5}\}_{j=1}^{4}$;
   b. 5×5 standardizing matrix $\mathbf{\Sigma}^{-1}$
      i. as the Cholesky post-factor of the sample precision matrix $(\mathbf{\Sigma\Sigma}^t)^{-1}$ (usual case),
      ii. or if $\mathbf{\Sigma\Sigma}^t$ has any too-small eigenvalue (e.g., due to spurious correlation of too-few data, which only happened in a few experiments), then as the inverse matrix square-root of $\mathbf{\Sigma\Sigma}^t$ projected onto its significant eigenvectors;
   c. time series of $\|\boldsymbol{\delta}\|$,
      i. constrained by day-of-year and time-of-day burn requirements,
      ii. sorted ascending from the minimum $\|\boldsymbol{\delta}\|$ (to enable comparing low-$\|\boldsymbol{\delta}\|$ days),
      iii. written to text files `M_*-*.txt`, and
      iv. plotted in Figure 7.

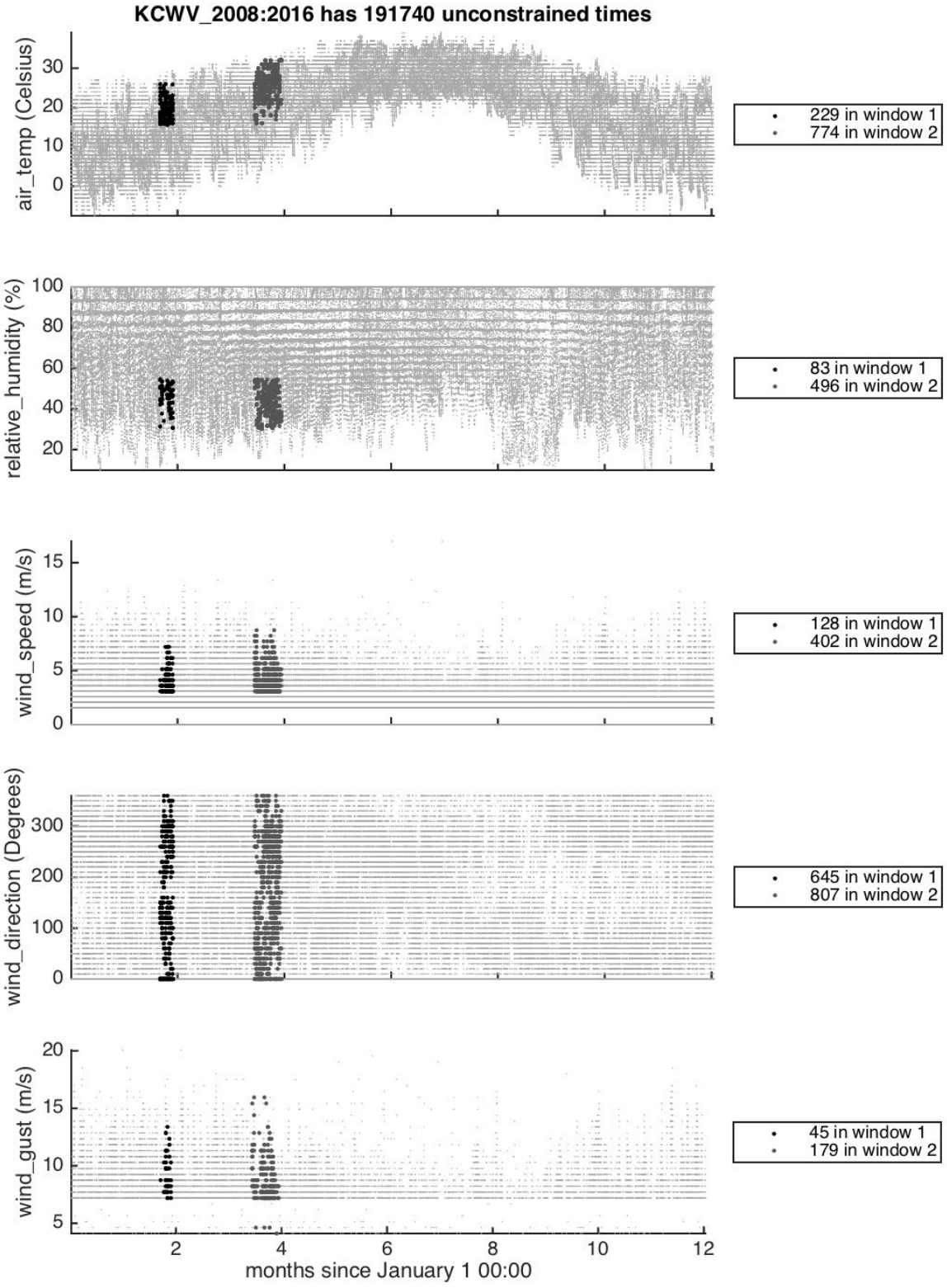


Figure 25. As in Figure 5 but for station KCWV (Claxton Evans County Airport, 32.19510°N, 81.86960°W 111 ft in Georgia) from 2008 to 2016 with 2 burn windows per year.

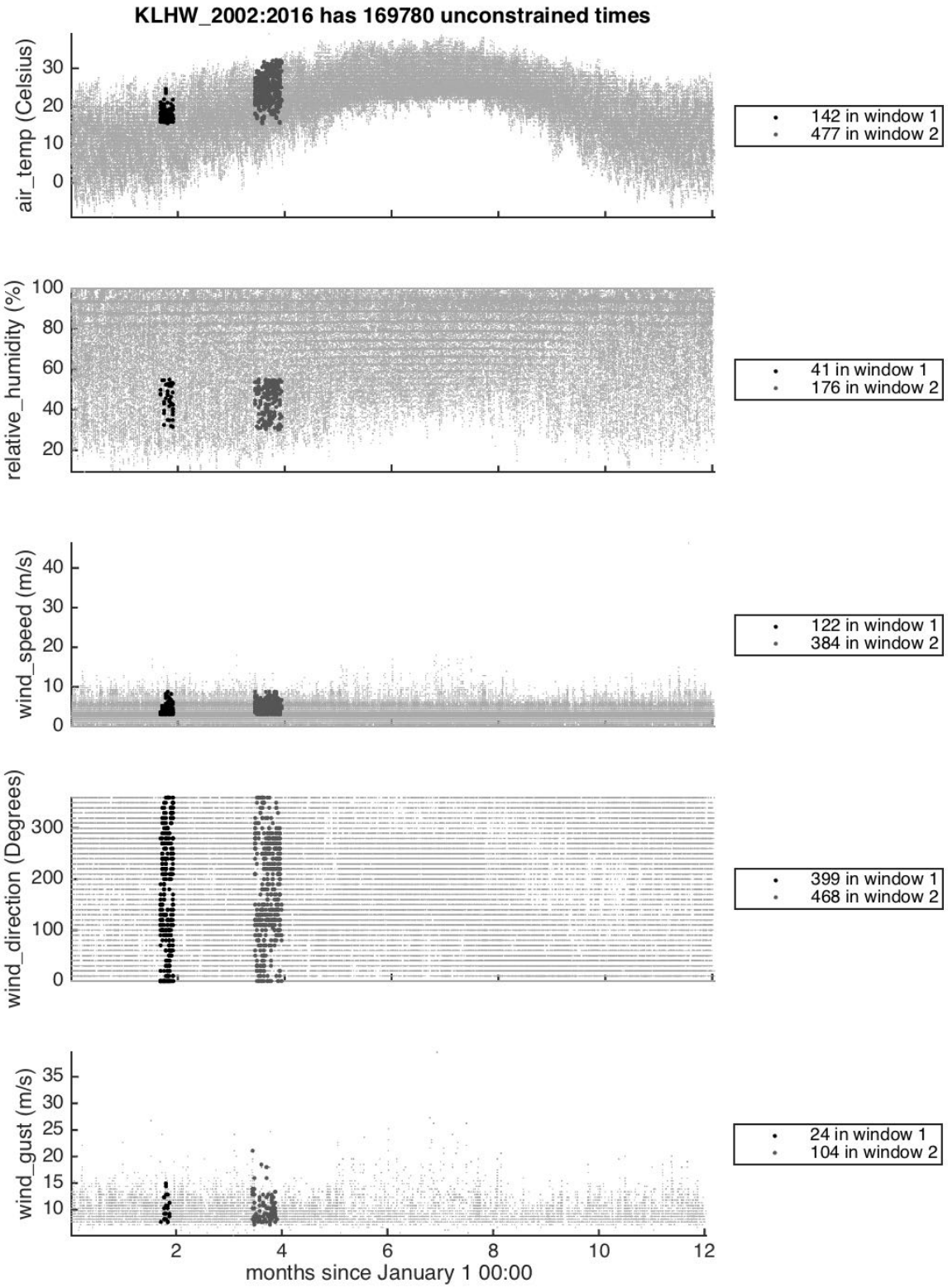


Figure 26. As in Figure 5 but for station KLHW (Ft. Stewart, 31.88333°N, 81.56667°W, 46 ft in Georgia) from 2002 to 2016.

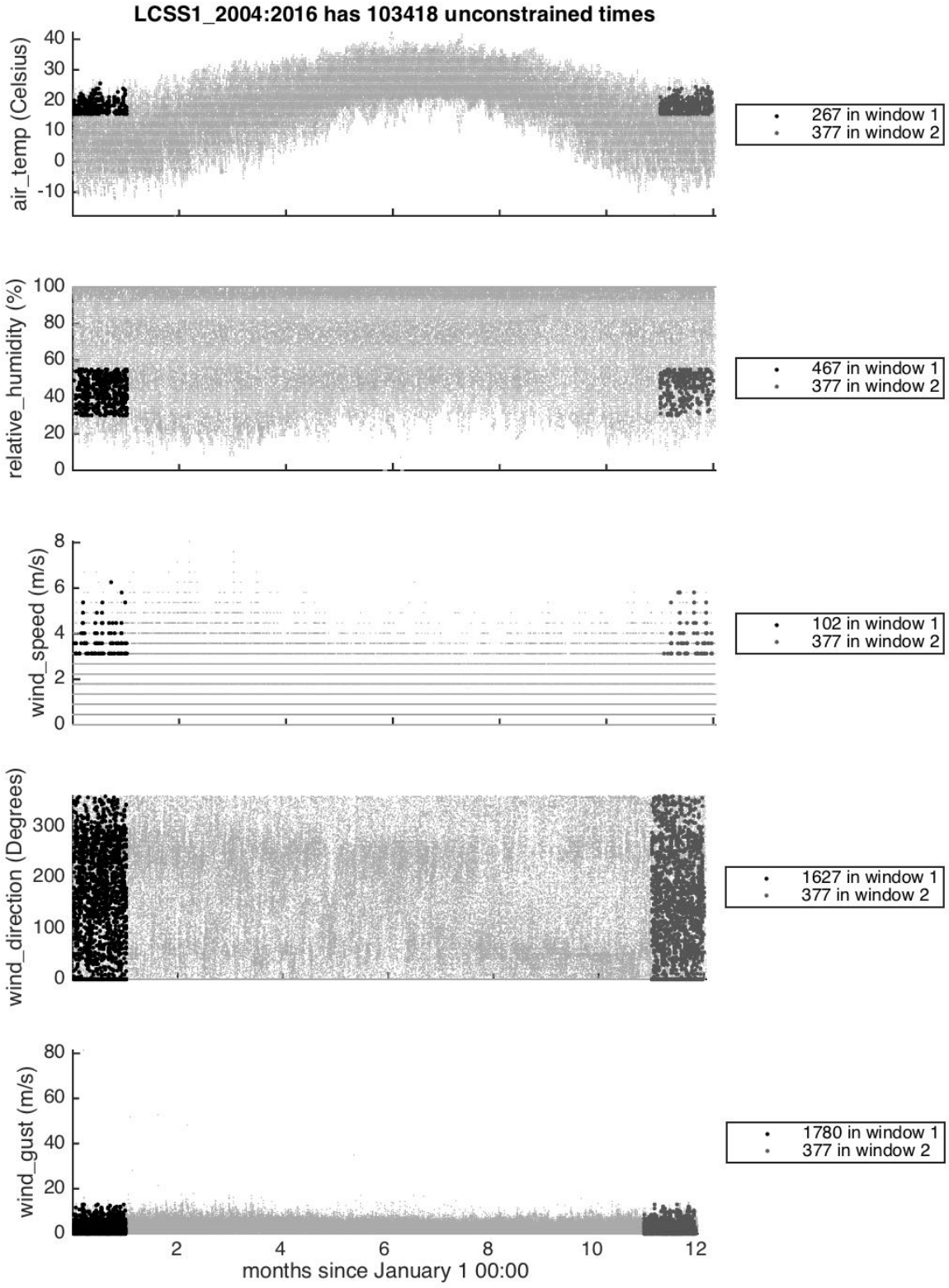


Figure 27. As in Figure 5 but for station LCSS1 (Savriv, 33.33305°N, 81.591667°W, 275 ft in South Carolina) from 2004 to 2016.

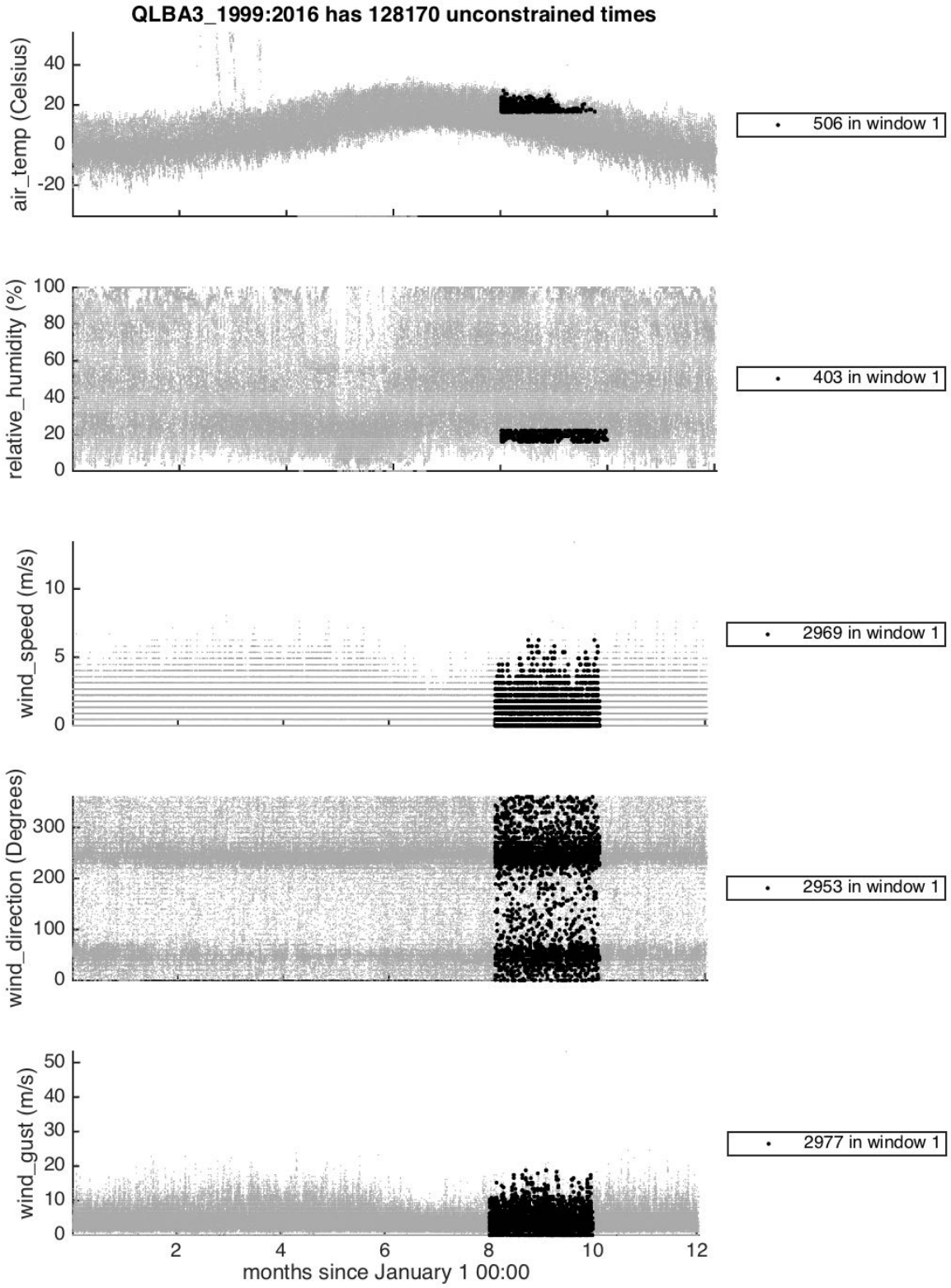


Figure 28. As in Figure 5 but for station QLBA3 (Lindbergh Hill, 36.285556°N, 112.078611°W, 8800 ft in Arizona) for 1999--2016.

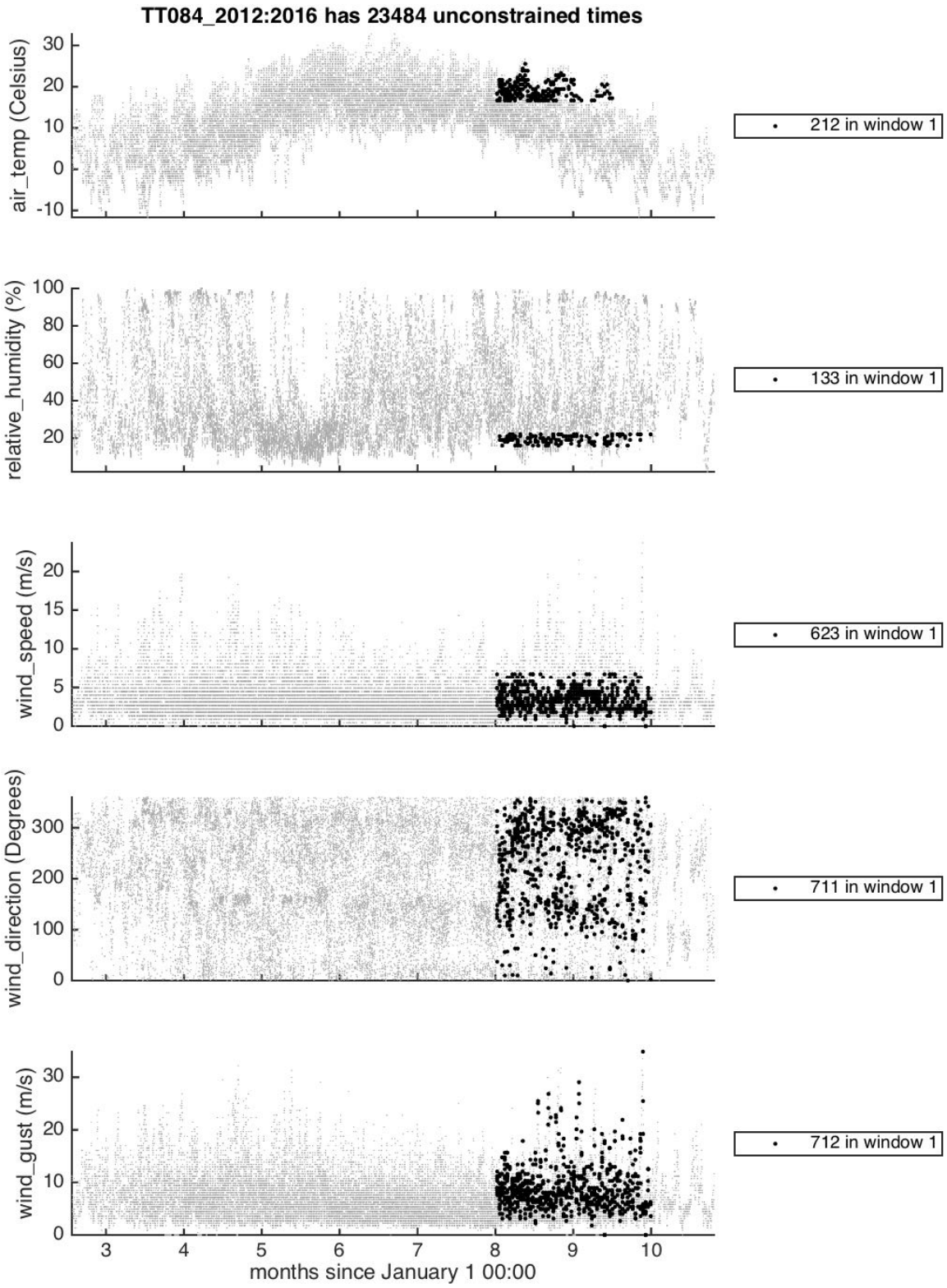


Figure 29. As in Figure 5 but for station TT084 (Fishlake D4 Pt #4, 38.960319°N, 111.405983°W, 8523 ft in Utah) for 2012--2016.

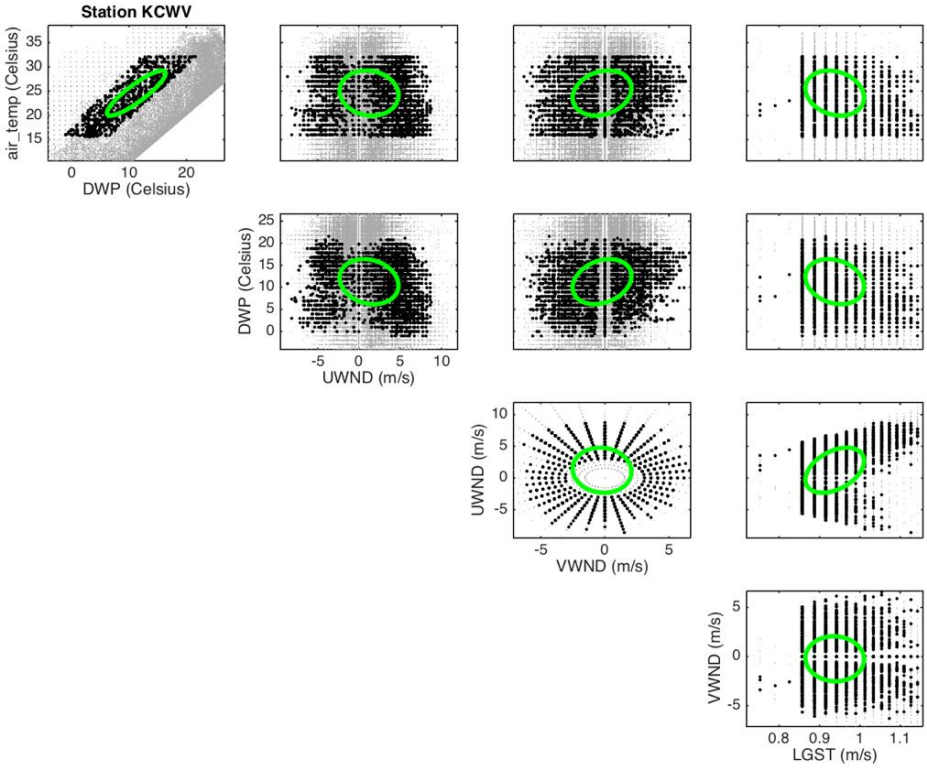


Figure 30. As in Figure 6 but for station KCWV.

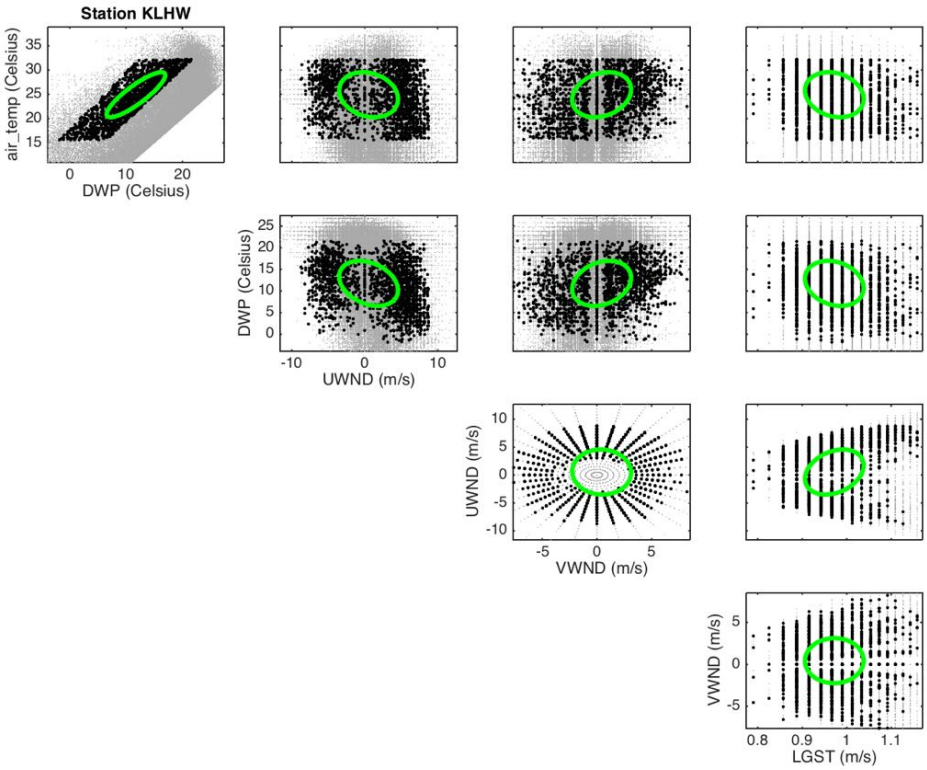


Figure 31. As in Figure 6 but for station KLHW.

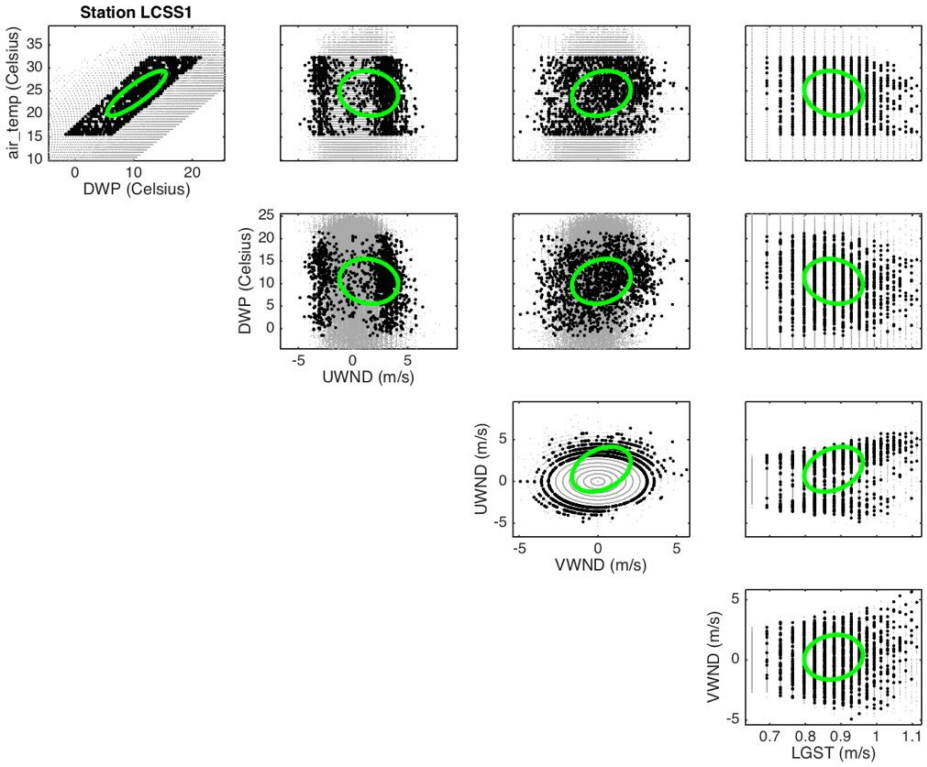


Figure 32. As in Figure 6 but for station LCSS1.

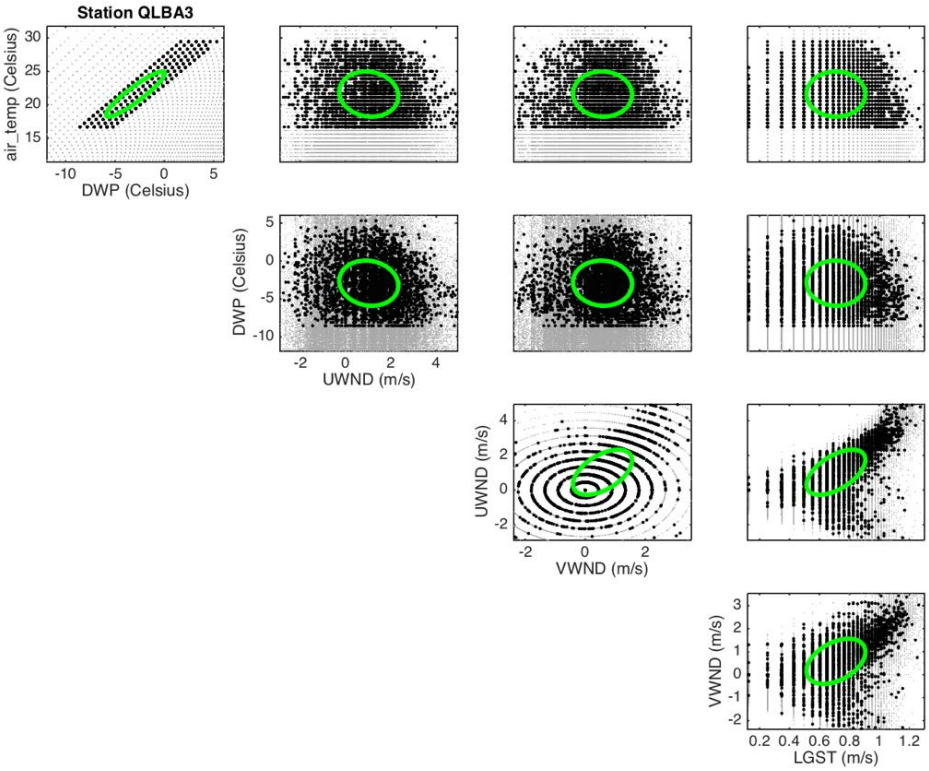


Figure 33. As in Figure 6 but for station QLBA3.

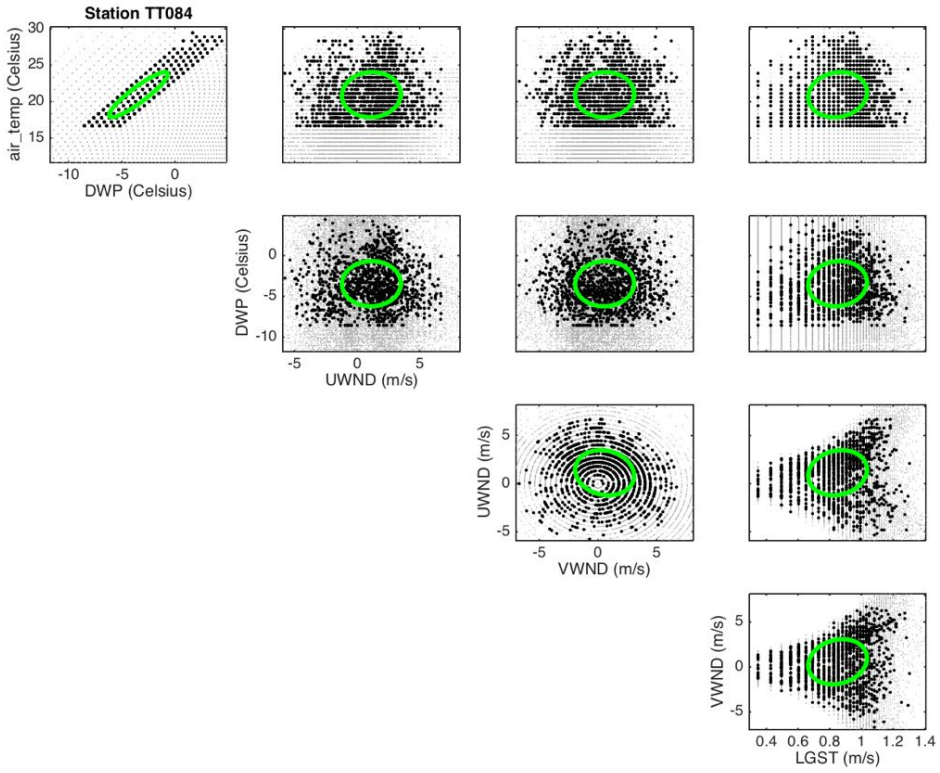


Figure 34. As in Figure 6 but for station TT084.

Appendix B: Latin Hypercube Sampling (rLHS) method used to define sets of parameters for the sensitivity study of sensor placement.

Latin Hypercube Sampling (LHS). Consider a model with L=7 parameters, divide the range of each parameter into N intervals of equal probability, and for each interval choose the probability midpoint as a sampling point 1,…,N. Then set up N parameter vectors by choosing the sampling

Table 7: Latin hypercube sampling. Each of L=7 parameters takes the values from its N=5 sampling points. The rows of the table are random permutations of the numbers 1 to 5, while the columns are the parameter vectors on which the simulations will be run.

3	1	2	4	5
3	1	5	4	2
4	3	2	1	5
3	5	2	1	4
1	3	2	4	5
4	5	1	3	2
5	2	4	3	1

points of each of the L parameter randomly, without repeating the same sampling point in any of the parameters. The simulation is run on each parameter vector. See Table 7 for an example.

rLHS consists of r repeats of LHS, with new random permutations of the choices of the sampling points for each parameter. The complete rLHS scheme requires rN simulations, with each simulation receiving a vector of L parameters.

Sobol's variance decomposition. Each of the rN simulations delivers its output Y. Now fix one of the sampling points x_{ij} of the parameter X_i and take the average of the outputs Y over all simulations where parameter had that value, $X_i = x_{ij}$. The result is a function of the value of the parameter X_i. This function is called the *conditional expectation of* Y *given* X_i, and we denote it by Y_i. Since the parameter X_i attains N sampling values with equal probability $1/N$, the conditional mean Y_i also attains N possible values with equal probability $1/N$. From those values of Y_i, we compute its variance $\mathrm{var}(Y_i)$ in the usual way except that we use the denominator N instead of $N+1$ in the expression for the variance. We also compute the variance $\mathrm{var}(Y)$ over all of its rN values, again using rN in the denominator instead of the usual $rN+1$.

The importance of this procedure lies in the following interpretation:

- $\mathrm{var}(Y)$ is the total variability of the output Y over the range of the parameters $X_1,\ldots,X_L$. We take this as the strength of the signal from the sensor.
- The Variance of the Conditional Expectation (VCM) $\mathrm{var}(Y_i)$ is an estimate of the variability of the output Y due to the parameter X_i.
- The ratio $\mathrm{eff}(Y_i) = \mathrm{var}(Y_i)/\mathrm{var}(Y)$ is an estimate of the relative effect of parameter X_i on the output Y. It is called correlation ratio (McKay 1995) or sensitivity index (Saltelli et al. 2004) of the parameter X_i.

We will take Y to be one of measured values at some location, e.g., the vertical wind velocity at a fixed time, longitude, latitude, and height. If one is interested in the effect of one particular parameter, it makes sense to look for locations where the sensitivity index of that parameter is large. Then the signal due to that parameter stands out from the rest.

A simple linear example. In the case of when the model is linear.

$$Y = a_1 X_1 + \cdots + a_n X_n$$

where the a_k are constant and the X_k are independent random variables, the variance of the conditional expectation is

$$\mathrm{var}\,\mathrm{E}(Y \mid X_i) = a_i \,\mathrm{var}\, X_i .$$

It can be shown that under suitable assumptions (which are certainly satisfied, e.g., when X_k have normal distribution), the rLHS procedure with Sobol's variance decomposition above satisfies

$$\operatorname{var}\left(Y_i\right) \to a_i \operatorname{var} X_i \text{ as } N \to \infty \text{ and } r \to \infty .$$

That is, the method recovers asymptotically the variance of one parameter from a signal that combines the effect of many independent parameters, by probing the model. Of course, if the parameters are not independent, or the model is not linear, the recovery is only approximate.

Inference with a small number of simulations. For a small number of sampling points and small number of repeats, we are far from the asymptotic range. But when different signals Y are taken from the same set of simulations and, in particular, they are same measurement at nearby locations, their sampling errors are close and comparing the variance of the conditional expectation for those signals is justified.

Time series. Since our quantities of interest will be associated with a fixed location while the fire is moving, there is a significant signal only when the fire or smoke cloud happens to move over the sensor location. Hence, we are also interested in the sums of time series of variances associated with a fixed location, which represents the total variability captured by the sensor over time.

Appendix C: Matlab scripts

```
% Author:               Aime' Fournier
% Project Title:        Modeling support for FASMEE experimental design using WRF-SFIRE-CHEM
% JFSP project number:  16-4-05-3
% File:                 analXls.m
% Purpose:              Analyze .xlsx spreadsheets as described below.

 clear
%
% The intention is to run this script section-by-section (between "%%") from
% the Matlab editor using the Command-Enter keystroke;
% however, it may work simply from the Matlab command window too.
% It will display what it is doing.
%
% The following few lines may need to be edited for customized use:
%
 ext = 'csv';                           % spreadsheet file extension
 bReqF = fullfile(ext, ['bReqs.' ext]); % burn requirements file
 matF = fullfile('mat', 'analXls.mat'); % .mat file name
 wd = '~/research/UCD/FASMEE/fasmee/aimefournier/matlab/'
 cd(wd)                                 % directory of scripts
 if ~exist(matF, 'file')                % spreadsheet processing not yet done
    datef = 'yyyy-mm-ddTHH:MM:SSZ';     % date format
    datList = {'air_temp' ...           % data types of interest
       'relative_humidity' 'wind_speed' 'wind_direction' 'wind_gust'}
    % datList = {'TMP' 'RELH' 'SKNT' 'GUST' 'DRCT'};
    %
    % Below here, everything should work automatically...
    %
    nDat = length(datList);             % nu. data types
    %
    % Assign station and data names from bReqF to cell array sta:
    %
    T = table2cell(readtable(bReqF, 'ReadVariableNames', false));
    bReqH = cellfun(@(x) ...            % burn requirements header
       sscanf(x, '%s %s%*s%*s'), T(1,2 : end), 'UniformOutput', false)
    nReq = length(bReqH)/2;             % assume min, max for each requirement
    fprintf('%s contains %d (min,max) burn-requirement pairs', bReqF, nReq)
    sta = cellfun(...                   % formatted read of station names:
       @(x) sscanf(x, 'ID = %s'), T(2 : end,1), 'UniformOutput', false)';
    l = ones(1, length(sta));           % allocate nu. time windows at stations
    for i = 2 : length(sta)             % seek multiple windows at each station
       j = find(strcmp(sta{i}, sta(1 : i - 1)));
       if ~isempty(j)                   % length(j) > 0 means duplicate names ...
          l([i j]) = length(j) + 1;     % assign equal counts to duplicates
       end
    end,clear i j
    [sta i] = unique(sta);
    %
    % Change unique(sta) into a structure:
    %
    sta = struct('name', sta, 'nWin', num2cell(l(i)));
    fprintf(' for stations\n\t%s\b.\n', sprintf('%s ', sta.name))
    lSta = 1 : length(sta);             % station list
    %
    % Assign burn required data min, max at each station:
    %
    k = 2;                              % skip header row
    for i = lSta                        % station loop:
       for j = 1 : sta(i).nWin          % window loop:
          fprintf('Station %5s, window %d requires\n ', sta(i).name, j)
          m = reshape(cellfun(@str2num, T(k,2 : 2*nReq - 1)), 2, nReq - 1);
          for l = 1 : nDat
             f = find(strcmp( ...       % compare to data type list:
                datList{l}, cellfun(@(x) x(4:end), bReqH(2*(1 : nReq - 1)), 'UniformOutput', false)));
             if isempty(f)
                sta(i).bReqs(1 : 2,l,j) = [-Inf Inf];
             else
                sta(i).bReqs(1 : 2, ... % min & max values of data:
                   l,j) = m(:,f);
                fprintf('%d<=%s<=%d, ', sta(i).bReqs(1,l,j), bReqH{2*f}(4:end), sta(i).bReqs(2,l,j))
                switch bReqH{2*f}(4:end)% convert units:
                   case 'wind_speed'    % convert mi/h to m/s:
                      sta(i).bReqs(:,l,j) = ...
                         sta(i).bReqs(:,l,j)*1609.344/60^2;
```

```
                    case 'air_temp'      % convert deg F to deg C:
                       sta(i).bReqs(:,l,j) = ...
                          (sta(i).bReqs(:,l,j) - 32)*5/9;
                 end
              end
           end
           for l = 1 : 2                % start and finish of time window:
              [~, sta(i).bmdh{l,1 : 3,j}] = datevec(T{k,2*nReq - 1 + l});
           end
           k = k + 1;                   % point to next row
           fprintf('%02d/%02d<=day<=%02d/%02d, %02d<=hour<=%02d.\n', ...
              cell2mat(sta(i).bmdh(:,1:2,j))', sta(i).bmdh{:,3,j})
        end
        sta(i).bmdh = cell2mat(sta(i).bmdh);
     end,clear f i j k l m T
     for i = lSta                       % station loop:
        %
        % Read data from spreadsheet for station i:
        %
        d = dir(fullfile(ext, sprintf('%s*.%s', sta(i).name,ext)));
        fprintf('Processing file %35s, %d of %d, takes ', d.name, i, lSta(end))
        tic
        f = fopen(fullfile(ext, d.name));
        for j = 1 : intmax              % loop over metadata rows (starting with '#'):
           T = fgetl(f);
           if strcmp(T(1), '#')
              g = find(T == ':', 1, 'last' );
              eval(['sta(i).' matlab.lang.makeValidName(regexprep(T(find(T == ' ', 1 ) + 1 : g - 1),'
','_')) ...
                 '=''' T(g + 2 : end) ''';'])
           else
              break;                    % now T is the main data header now
           end
        end
        sta(i).hdr = T;                 % header row
        T = strsplit(strrep(T, ...      % truncate useless suffices and
           '_set_1', ''), ',');         % split strings delimited by ','
        datI = find(cellfun( ...        % data column indexes
           @(x) sum(ismember(datList, x)), T));
        T = strsplit(fgetl(f), ',');
        sta(i).u = T(datI - 1);         % units row
        T = textscan(f, ['%s%s' ...     % read 2 strings, then some floats, then skip to EOL:
           repmat('%f', 1, max(datI) - 2) '%*[^\n]'], 'Delimiter', ',');
        fclose(f);
        sta(i).t = datenum(T{2}, datef); % time (days) in column 2
        % sta(i).yr = unique(num2cell(datestr(sta(i).t, 'yyyy'), 2));
        sta(i).d = cell2mat( ...        % data in other columns
           T(:,datI));
        k = find(strcmp(datList,'air_temp'));
        T = find(sta(i).d(:,k) < -63 |  57 < sta(i).d(:,k));
        sta(i).d(T,:) = [];             % eliminate extreme air_temp
        sta(i).t(T  ) = [];
        k = find(strcmp(datList,'relative_humidity'));
        T = find(sta(i).d(:,k) <   0 | 100 < sta(i).d(:,k));
        sta(i).d(T,:) = [];             % eliminate impossible relative_humidity
        sta(i).t(T  ) = [];
        [~, k] = intersect(datList, {'wind_speed', 'wind_gust'}, 'stable');
        T = find(any(sta(i).d(:,k) <   0 | 103 < sta(i).d(:,k), 2));
        sta(i).d(T,:) = [];             % eliminate extreme or impossible wind
        sta(i).t(T  ) = [];
        sta(i).nt = length(sta(i).t);   % nu. times
        fprintf('%5.1fs\n',toc)
     end, clear d datI f i j T
     save(matF)
  else
     tic
     fprintf('Loading %s takes ',matF)
     load(matF)
     fprintf('%5.1fs\n',toc)
  end

  %%
  analXls_figs                          % Don't run after transforming below!
  %%
  %
  % Print the analXls_figs figures:
  %
  for i = lSta
```

```
   print(i, '-dpng', sprintf('figures/%s_data', sta(i).name))
 end
%%
 fprintf('Transform (%s\b)\n', sprintf('%s ', datList{:}))
 d2c = @(x)-[sind(x),cosd(x)];          % direction-to-component transform
 %
 % Get indexes for data types:
 %
 kd = find(strcmp(datList,'wind_direction'));
 kg = find(strcmp(datList,'wind_gust'));% index for wind gust
 kr = find(strcmp(datList,'relative_humidity'));
 ks = find(strcmp(datList,'wind_speed'));
 kt = find(strcmp(datList,'air_temp')); % index for temperature
 %
 % Assign new data-type names:
 %
 datList([kr ks kd kg]) = {... %'LRH'
    'DWP' 'UWND' 'VWND' 'LGST'};
 fprintf('\tto (%s\b).\n', sprintf('%s ', datList{:}))
 for i = lSta                           % station loop:
    sta(i).d(:,kr) = ...                % change RELH to dew-point temperature:
       RELH2DWT(sta(i).d(:,kr), sta(i).d(:,kt)); %#ok<*FNDSB>
%      log10(sta(i).d(:,kr));           % _or_ just take its log10.
    sta(i).u{kr} = 'Celsius';           % update units
    sta(i).d(sta(i).d(:,ks) == 0,kd) ...% replace wind_direction NaNs when wind_direction not defined:
        = 0;
    sta(i).d(:,[ks kd kg]) = ...        % apply d2c and GUST -> LGST:
       [bsxfun(@times, sta(i).d(:,ks), d2c(sta(i).d(:,kd))) log10(sta(i).d(:,kg))];
    sta(i).u{kd} = sta(i).u{ks};        % change label units
 end,clear d2c i j kd kg kr ks kt
 %%
 if ~any(strcmp(fieldnames(sta), 'mom'))% Moments not computed yet
    %
    % Get indexes for data types:
    %
    kt = find(strcmp(datList(1 : nDat), 'air_temp' ));
    kd = find(strcmp(datList(1 : nDat), 'DWP'      ));
    ku = find(strcmp(datList(1 : nDat), 'UWND'     ));
    kv = find(strcmp(datList(1 : nDat), 'VWND'     ));
    cm = gray(max(arrayfun(@(x)x.nWin, sta)) + 2);
    cm(end,:) = [];                     % don't use white color
    ib = @(a,x) a(1) <= x & x <= a(2);  % in-between condition
    for i = lSta                        % station (with burn requirements) loop:
       %
       % logical f tests the burn requirements, 1st, not too hot or cold:
       %
       f = ib(sta(i).bReqs(:,kt,1), sta(i).d(:,kt));
       fprintf('%5s has %4dd OK air_temp, ', sta(i).name, sum(f))
       %
       % ... then not too wet or dry:
       %
       f = f & ib(sta(i).bReqs(:,kd,1), RELH2DWT(sta(i).d(:,kd), sta(i).d(:,kt), -1));
       fprintf('%4d OK relative_humidity, ', sum(f))
       %
       % ... then not too fast or slow:
       %
       f = f & ib(sta(i).bReqs(:,ku,1), sqrt(sum(sta(i).d(:,[ku kv]).^2, 2)));
       fprintf('%4d OK wind_speed, ', sum(f))
       f = bsxfun(@and, f, isfinite(sta(i).d));
       %
       % sample size n:
       %
       sta(i).mom.n = sum(f);
       %
       % Allocate sample mean m, std dev s, upper-diagonal correlation r:
       %
       sta(i).mom.m = NaN(  1, nDat             );
       sta(i).mom.s = NaN(  1, nDat             );
       sta(i).mom.r = NaN(  1, nDat*(nDat - 1)/2);
       fprintf('[%s\b] finite values\n', sprintf('%3d ', sta(i).mom.n))
       for k = 1 : nDat
          if sta(i).mom.n(k) > 0
             %
             % Compute the sample mean:
             %
             sta(i).mom.m(k) = mean(sta(i).d(f(:,k),k));
          else
             fprintf('%s %s has no data\n', sta(i).name, datList{k})
```

```
         end
         if sta(i).mom.n(k) > 1
            %
            % Compute the std dev:
            %
            sta(i).mom.s(k) = std(sta(i).d(f(:,k),k));
         else
            fprintf('%s %s has one datum\n', sta(i).name, datList{k})
         end
      end
      figure(lSta(end) + i)
      set(gcf, 'Units', 'normalized', 'OuterPosition', [1/8 0 7/8 1], 'Units', 'pixels')
      clf
      f = logical(prod(f, 2));         % 'AND' through all requirements
      [~,~,~,h] = datevec(sta(i).t);   % hour-of-day for each time
      if sum(f) > 1                    % at least 2 data pass requirements
         m = 0;                        % initialize correlation index
         for k = 1 : nDat - 1          % correlation row k
            for j = k + 1 : nDat       % correlation column j (upper diagonal)
               m = m + 1;              % update correlation index
               %
               % Compute the j,k--correlation:
               %
               a = corrcoef(sta(i).d(f,j), sta(i).d(f,k));
               sta(i).mom.r(m) = a(1,2);
               subplot(nDat - 1, nDat - 1, (nDat - 1)*(k - 1) + j - 1, 'align')
               plot(sta(i).d(~f,j), sta(i).d(~f,k), '.', 'Color', cm(end,:), 'MarkerSize', 1, 'ZData',
-ones(sum(~f), 1))
               l = 1;                  % same color for all windows
               %
               % Enforce data-value conditions:
               %
               g = f;% &   b(:,1) <= sta(i).t & sta(i).t <= b(:,2) & ib(sta(i).bmdh(:,3,l), h) ;
                  line(sta(i).d(g,j), sta(i).d(g,k), 'Color', cm(  l,:), 'LineStyle', 'none',
'Marker', '.', 'MarkerSize', 6, 'ZData', ones(sum(g), 1))
               set(covEllip(sta(i).mom.s([j k]), sta(i).mom.r(m), sta(i).mom.m([j k]), 32), ...
                  'Color', 'g', 'LineWidth', 3, 'ZData', 2*ones(32, 1))
               if j == k + 1
                  if k == 1
                     title(sprintf('Station %s', sta(i).name))
                  end
                  xlabel(sprintf('%s (%s)', strrep(datList{j}, '_', '\_'), sta(i).u{j}))
                  ylabel(sprintf('%s (%s)', strrep(datList{k}, '_', '\_'), sta(i).u{k}))
               else
                  set(gca, 'XTickLabel', '', 'YTickLabel', '')
               end
               axis([sta(i).mom.m(j)+[-1 1]*3*sta(i).mom.s(j) ...
                     sta(i).mom.m(k)+[-1 1]*3*sta(i).mom.s(k)])
            end
         end
         drawnow
         orient tall
         orient landscape              % needs to follow 'tall'
      else
         fprintf('\n%s has no good data\n\n', sta(i).name)
      end
   end, clear a f g i ib j k kd kt ku kv l m n
end
%%
for i = lSta                    % station (with burn requirements) loop:
   figure(lSta(end) + i)
   print('-dpng', sprintf('figures/%s_scat', sta(i).name))
end
%%
analXls_Mdists
```

```
% Author:                Aime' Fournier
% Project Title:         Modeling support for FASMEE experimental design using WRF-SFIRE-CHEM
% JFSP project number:   16-4-05-3
% File:                  analXls_figs.m

 cm = gray(max(arrayfun(@(x)x.nWin, sta)) + 2);
 cm(end,:) = [];                          % don't use white color
 ib = @(a,x) a(1) <= x & x <= a(2);       % in-between condition
 for i = 1Sta                             % station (with burn requirements) loop:
    figure(i)
    set(gcf, 'Units', 'normalized', 'OuterPosition', [1/8 0 7/8 1], 'Units', 'pixels')
    clf
    yr = datevec(min(sta(i).t)) : ...
         datevec(max(sta(i).t));          % years for station i
    nt = zeros(nDat, sta(i).nWin);        % number of OK times for each window
    for j = 1 : length(yr)                % year loop at station i:
       g = find(datevec(...               % times in year j:
          sta(i).t) == yr(j));
       Jan1 = datenum(sprintf('%4d-01-01T00:00:00Z', yr(j)), datef);
       t = (sta(i).t(g) - Jan1)*12/365; % months since January 1 00:00
       for k = nDat : -1 : 1              % data-type loop:
          subplot(nDat, 1, k)
          line( t, sta(i).d(g,k), 'Color', cm(end,:), 'LineStyle', 'none', ...
             'Marker', '.', 'MarkerSize', 1, 'ZData', -ones(size(t)));
          [~,~,~,d] = datevec(sta(i).t(g));
          for l = 1 : sta(i).nWin
             b = datenum(yr([j j])', ...% month-day limits for window l:
                sta(i).bmdh(:,1,l), sta(i).bmdh(:,2,l));
             %
             % Enforce month-day-hour and data-value conditions:
             %
             f = find(ib(       b          , sta(i).t(g)  ) & ...
                      ib(sta(i).bmdh(:,3,l), d            ) & ...
                      ib(sta(i).bReqs(:,k) , sta(i).d(g,k)) & ...
                      any(isfinite(sta(i).d(g,:)), 2)      );
             if isempty(f)
                fprintf('No OK %s data for %s_%d window %d of %d\n', datList{k}, sta(i).name,
yr(j), l, sta(i).nWin)
             else
                nt(k,l) = nt(k,l) + length(f);
                h(l) = line( t(f), sta(i).d(g(f),k), 'Color', cm(l,:), 'DisplayName', sprintf('%d
in window %d', nt(k,l), l), 'LineStyle', ...
                             'none', 'Marker', '.', 'MarkerSize', 1, 'ZData', ones(size(f)));
             end
          end
          axis tight
          if j == length(yr)
             legend(h(1 : sta(i).nWin), 'Location', 'eastoutside')
          end
          ylabel([strrep(datList{k}, '_', '\_') ' (' sta(i).u{k} ')'])
          if k == nDat
             xlabel('months since January 1 00:00')
          else
             set(gca, 'XTickLabel', '')
          end
       end
    end
    title(sprintf('%s\\_%d:%d has %d unconstrained times', sta(i).name, yr([1 end]), sta(i).nt))
    drawnow
    orient tall
 end,clear b cm d f g h i ib j Jan1 k l nt t yr
```

```
% Author:                Aime' Fournier
% Project Title:         Modeling support for FASMEE experimental design using WRF-SFIRE-CHEM
% JFSP project number:   16-4-05-3
% File:                  analXls_Mdists.m

 cm = gray(max(arrayfun(@(x)x.nWin, sta)) + 2);
 cm(end,:) = [];                          % don't use white color
 ib = @(a,x) a(1) <= x & x <= a(2);       % in-between condition
 figure(2*lSta(end) + 1)
 clf
 clear j
 for i = lSta                             % station loop:
    yr = datevec(min(sta(i).t)) : ...
         datevec(max(sta(i).t));
    subplot(lSta(end), 1, i)
    if any(sta(i).mom.n <= 1)
       text(.1, .5, sprintf('%s has <%d burn days.', sta(i).name, max(sta(i).mom(1).n) + 1),
'FontSize', 20)
       axis off
    else
       %
       % Assign the strictly upper triangular correlation matrix:
       %
       p = zeros(nDat);
       m = 0;
       for k = 1 : nDat - 1               % row loop:
          for j = k + 1 : nDat            % column loop (upper triangle):
             m = m + 1;
             p(k,j) = sta(i).mom.r(m);    % corr(d(k),d(j))
          end
       end, clear j k m
       q = diag(sta(i).mom.s);            % std(d)
       %
       % Add the id and lower triangle, and multiply in the std:
       %
       p = q*(p + p' + eye(nDat))*q;      % cov(d)
       [r j] = eig(p, 'vector');
       %
       % Change to sqrt of precision matrix:
       %
       if min(j) <= 0           % suspect spurious correlations
          fprintf('%5s %9.1e <=  eigenvalues <= %9.1e rectified.\n', sta(i).name, min(j), max(j))
          [r j] = deal(r(:,j > 0), j(j > 0));
          p = r*diag(1./sqrt(j));         % p*p' is the precision matrix orthogonal to the singular
directions
       else
          p = chol(inv(p), 'lower');      % p*p' is the full precision matrix
       end
       %
       % Time series of Mahalanobis deviations from the mean:
       %
       sta(i).Md.v = sqrt(sum(((sta(i).d - repmat(sta(i).mom.m, sta(i).nt, 1))*p).^2, 2));
       [~,~,~,h] = datevec(sta(i).t);     % hour-of-day for each time
       for l = 1 : sta(i).nWin            % station i, time-window loop:
          b = datevec(sta(i).t);          % year, month, day, hour, minute, second
          b = datenum(repmat(b(:,1), 1, 2), repmat(sta(i).bmdh(:,1,l)', sta(i).nt, 1) ...
                                          , repmat(sta(i).bmdh(:,2,l)', sta(i).nt, 1));
          %
          % Enforce month-day-hour conditions:
          %
          f = find(b(:,1) <= sta(i).t & ...
                   b(:,2) >= sta(i).t & ...
                   ib(sta(i).bmdh(:,3,l), h));
          [~, j] = sort(sta(i).Md.v(f));
          %
          % Minimum of time-conditioned Mahalanobis deviation, and its time:
          %
          sta(i).Md.x(l) = sta(i).Md.v(f(j(1)));
          sta(i).Md.t(l) = sta(i).t(   f(j(1)));
          %
          % month-fraction of each year:
          %
          sta(i).Md.m(l) = (sta(i).Md.t(l) - ...
```

```
                datenum(sprintf('%4d-01-01T00:00:00Z', datevec(sta(i).Md.t(l))), datef))*12/365;
            j = j(1 : 32);
            %
            % Write some smallest values:
            %
            t = num2cell([sta(i).Md.v(f(j)) sta(i).d(f(j),:)],1);
            t = {datestr(sta(i).t(f(j)), datef) t{:}};
            writetable(table(t{:}, 'VariableNames', {'time' 'M_dev' datList{:}}), ...
               sprintf('txt/M_%s-%d.txt',sta(i).name,l))
         end
         for j = 1 : length(yr)           % year loop at station i:
            Jan1 = datenum(sprintf('%4d-01-01T00:00:00Z', yr(j)), datef);
            f = datevec(sta(i).t);        % year, month, day, hour, minute, second
            f = f(:,1) == yr(j);          % times in year j:
            t = (sta(i).t(f) ...          % months since January 1 00:00
                 - Jan1)*12/365;
            line(t, sta(i).Md.v(f), 'Color', cm(end,:), 'LineStyle', 'none', 'Marker', '.',
'MarkerSize', 1, 'ZData', -ones(sum(f),1))
            for l = 1 : sta(i).nWin
               g = datenum(yr([j j])', sta(i).bmdh(:,1,l), sta(i).bmdh(:,2,l));
               %
               % Enforce month-day-hour and data-value conditions:
               %
               g = f & ib(g, sta(i).t) & ib(sta(i).bmdh(:,3,l), h);
               line((sta(i).t(g) - Jan1)*12/365, sta(i).Md.v(g), 'Color', cm(l,:), 'LineStyle',
'none', 'Marker', '.', 'MarkerSize', 6, 'ZData', ones(sum(g),1))
            end
         end
         line(sta(i).Md.m, sta(i).Md.x, 'DisplayName', 'min', 'LineStyle', 'none', 'Marker', 'v',
'MarkerSize', 2^3, 'ZData', ones(sta(i).nWin,1))
         set(gca, 'YScale', 'log', 'YTick', 2.^(floor(log2(sta(i).Md.x(1))) : 2 : ...

ceil(log2(max(sta(i).Md.v(isfinite(sta(i).Md.v)))))))
         axis tight
         g = axis;
         axis([0 12 g(3:4)])
%        set(legend('Location', 'eastoutside'), 'FontSize', 5)
         [u{1:4}] = datevec(min(sta(i).Md.t));
         title(sprintf( ...
               '{\\it\\delta}[%d,%d,%d,%d] = %5.2f \\leq (%5s-%d %d:%d M-dev.
{\\it\\delta}[{\\ity},{\\itm},{\\itd},{\\ith}])', ...
               u{:}, sta(i).Md.x(1), sta(i).name, l, yr([1 end])))
         if i == lSta(end)
            xlabel('months since January 1 00:00')
         end
      end
   end, clear f g i j Jan1 k l m p q r t u v
   orient tall
```

```
function h = covEllip(s,r,m,n)
%
% h = covEllip(s,r,m,n)
%
% Return handle h to line of ellipse [x-m(1), y-m(2)]*inv(c)*[x-m(1); y-m(2)]
% == 1 centered at m (default [0;0]) in the x-y plane, where c =
% [s(1)^2, s(1)*s(2)*r; s(1)*s(2)*r, s(2)^2] = v*d*v' is a 2D covariance
% matrix with correlation r (default 0), 2 real orthogonal eigenvectors
% v(:,1:2) and 2 positive eigenvalues d(1,1), d(2,2). Use 1 < n (default
% 32) points for the line.

% Author:               Aime' Fournier
% Project Title:        Modeling support for FASMEE experimental design using WRF-SFIRE-CHEM
% JFSP project number:  16-4-05-3
% File:                 covEllip.m
%
 if nargin < 4
    n = 32;
 end
 if nargin < 3
    m = [0;0];
 end
 if nargin < 2
    r = 0;
 end
 if any(s < 0)
    error('any(s < 0) not allowed')
 elseif abs(r) > 1
    error('|Correlation| > 1 not allowed')
 elseif n < 5
    error('Need n >= 5 line points')
 end
 x = sqrt(4*r^2*s(1)^2*s(2)^2 + (s(1)^2 - s(2)^2)^2);
 d(1,1) = (s(1)^2 + s(2)^2 + x)/(2*(1 - r^2)*s(1)^2*s(2)^2);
 d(2,2) = (s(1)^2 + s(2)^2 - x)/(2*(1 - r^2)*s(1)^2*s(2)^2);
 v = [s(1)^2 - s(2)^2 - x, s(1)^2 - s(2)^2 + x
           2*r*s(1)*s(2),       2*r*s(1)*s(2)];
 v = v*diag(1./sqrt(diag(v'*v)));
 u = (2*(0 : n - 1)/(n - 1) - 1)*pi;
 x = repmat(m(:), 1, n) + v*diag(1./sqrt(diag(d)))*[cos(u)
                                                    sin(u)];
 h = line(x(1,:), x(2,:));
```

```
function p = RELH2DWT( r, t, f )
%
% p = RELH2DWT( r, t [,  1])
% r = RELH2DWT( p, t  , -1 )
%
% At temperature t (Celsius), convert
% relative humidity r (0:100 %) to dew point temperature p (Celsius), or p back to r.
% See https://en.wikipedia.org/wiki/Dew_point.
%

% Author:               Aime' Fournier
% Project Title:        Modeling support for FASMEE experimental design using WRF-SFIRE-CHEM
% JFSP project number:  16-4-05-3
% File:                 RELH2DWT.m

 if nargin < 3
    f = 1;
 end
 b = 17.368*(0 <= t) + 17.966*(0 > t);
 c = 238.88*(0 <= t) + 247.15*(0 > t);
 d = 234.5;
 p = zeros(size(r));
 if f == 1
    z = r == 0;
    p(z) = -c(z);
    z = r ~= 0;
    g = log(r(z)/100) + (b(z) - t(z)/d).*t(z)./(t(z) + c(z));
    p(z) = c(z).*g./(b(z) - g);
 elseif f == -1
    z = r == -c;                        % r is p if f == -1
    p(z) = 0;
    z = r ~= -c;
    g = b(z).*r(z)./(c(z) + r(z));
    p(z) = 100*exp(g - (b(z) - t(z)/d).*t(z)./(t(z) + c(z)));
 else
    error('|%d| ~= 1',f)
 end
```