

Towards a Dynamic Data Driven Application System for Wildfire Simulation

Jan Mandel¹, Lynn S. Bennethum¹, Mingshi Chen¹, Janice L. Coen²,
Craig C. Douglas³, Leopoldo P. Franca¹, Craig J. Johns¹, Minjeong Kim¹,
Andrew V. Knyazev¹, Robert Kremens⁴, Vaibhav Kulkarni¹, Guan Qin⁵,
Anthony Vodacek⁴, Jianjia Wu⁵, Wei Zhao⁵, and Adam Zornes³

¹ University of Colorado Denver, Denver, CO 80217-3364, USA

² National Center for Atmospheric Research, Boulder, CO 80307-3000, USA

³ University of Kentucky, Lexington, KY 40506-0045, USA

⁴ Rochester Institute of Technology, Rochester, NY 14623-5603, USA

⁵ Texas A&M University, College Station, TX 77843-1112, USA

Abstract. We report on an ongoing effort to build a Dynamic Data Driven Application System (DDDAS) for short-range forecast of wildfire behavior from real-time weather data, images, and sensor streams. The system should change the forecast when new data is received. The basic approach is to encapsulate the model code and use an ensemble Kalman filter in time-space. Several variants of the ensemble Kalman filter are presented, for out-of-sequence data assimilation, hidden model states, and highly nonlinear problems. Parallel implementation and web-based visualization are also discussed.

1 Introduction

We report on an ongoing effort to build a Dynamic Data Driven Application System (DDDAS) for short-range forecast of wildfire behavior with models steered by real-time weather data, images, and sensor streams. The forecast should be based on all available data, such as fuel maps, airborne images, internet weather data, field sensor data stream, and raw weather station streams. The forecast should change quickly when new data arrives. Delayed data should be taken into consideration as well and the system should provide measurement steering. The complete system should provide animated visualization and user steering over the internet. In this paper, we are mainly focusing on new developments, perspectives, and practical issues that have emerged since the previous paper [1], where additional details and further references can be found.

The core of the system is a coupled weather-fire model [2], developed by adding a semi-empirical fire model, based on Rothermel fire spread formulas [3], to the Clark-Hall mesoscale atmosphere model [4]. The data needs to be assimilated into the coupled model. However, this software is a legacy FORTRAN code, which is very hard to modify. Instead, we have encapsulated the code and we use it in an Ensemble Kalman Filter (EnKF) scheme [5]. Steering of data

acquisition can be achieved from the Kalman filter formulas, which can be used to indicate how the variance would be reduced by measurements of different quantities. Assimilation of out-of-order data will be achieved by maintaining the composite time-space state of the simulation over a number of time steps and applying Kalman filter algorithms to the composite state rather than to the state at a single analysis time.

The weather feedback is essential to fire behavior: it is well known that fire creates its own weather. Related work in the literature that combines weather and fire modeling includes FIRETEC [6], which uses a much more detailed (and computationally demanding) physics and chemistry based model, and an implementation of the same fire model as here within the Weather Research and Forecasting (WRF) Model [7]. Practical field tools, which require the input of weather information, such as the wind, include FARSITE [8].

2 Building a Foundation for DDDAS

Current tools used in the field include calculator-based algorithms that predict point-based fire spread from a wind speed, terrain slope, and fuel type as well as two-dimensional PC-based extensions of these formulas [8]. Building awareness of more advanced capabilities that currently exist and of work in progress toward sophisticated systems based on DDDAS is a part of this project.

Thus, we undertook to demonstrate first that atmosphere-fire models several orders of magnitude more complex than current operational tools can be run in real time even on single processor machines. A suite of simulations in which 3–6 nested domains were used to telescope from a 10-km regional-scale weather model to 1.1 km - 41 m horizontal grid spacing respectively were compared, as well as simulations in which the feedback from the fire to the atmosphere was prevented. These demonstrated that this fire-atmosphere feedback was necessary to capture the evolution of the fire shape but that even the relatively coarse grid resolutions could capture the overall growth and progression of the fire. Such simulations with a few hundred meter horizontal grid size can easily be done on a single processor, i.e. such as a PC at a command center, at faster than real time.

Then, in the first application of its kind [9], we have demonstrated that this model could be applied routinely to real fires in Colorado during summer 2004. Upon the operator learning the location of a fire start, the model was configured to establish 3 nested modeling domains over the ignition point. Input data included a locally-run 48-hr regional weather model forecast (MM5), information about the fuel types available for fires across Colorado, regionwide terrain elevation, and fuel moisture noted in incident reports and diagnosed from weather station data. The predicted fire behavior and growth throughout the remainder of the 48-hr forecast was then modeled. Although many refinements can be made in this approach, it is the foundation upon which DDDAS techniques can be built.

3 Software Structure

The DDDAS is built from three main components interacting only through specified interfaces, so the components are independent and any of them can be exchanged with ease. There are multiple versions of each component.

1. The *model* component encapsulates all interaction with a specific model, such as advancing a single simulation to a specified time (Sec. 4), random perturbation of the simulation state, and data acquisition and comparing the data to the model state, i.e., the observation function (Sec. 7, 8). The model component will also interact with the visualization and model steering interface in future (Sec. 10).
2. The *EnKF* component contains an implementation of the filter algorithms in a model independent form (Sec. 5).
3. The *dynamic data assimilation driver* component [1] calls the model and the filter components. Its main function is to coordinate the advancement in time of system states and to present the EnKF with the task of updating an ensemble of state vectors. For this purpose, the driver combines states at multiple times into a single composite state and decides which time snapshots to keep and which to recreate on demand, and controls the advancing of all simulations in time. This idea is related to out-of-sequence particle filters in target tracking applications [10, 11], but similar ideas seem not to have been investigated in the context of meteorology and the much more efficient EnKF. The driver also handles any additional terms (Sec. 5, 6) by calls to model specific functions in the model component.

4 Fire Model

In the current fire model, the burning region in each fuel cell is represented by a quadrilateral defined by four points, called tracers. This submesh representation allows the cell to ignite gradually. The shape of the fireline can be quite complex. However, the EnKF assumes that there is a state u which can be meaningfully increased or decreased, and the tracer information is not such a quantity; for example, the fireline will need to be adjusted over several cells. Therefore, work is in progress to represent the location of the fire as the set of all points x such that $g(x) > 0$, where g is a level function from a finite element space. Level functions have been successfully used in PDE models with free boundary [12]. This work is also the foundation of eventually replacing the empirical fire model with one based on differential equations.

We have developed a simple standalone fire model, based on a finite difference discretization of the reaction-convection-diffusion equation [1]. This model, while admittedly very simple and not quite physical, does produce a reasonable fire behavior with an advancing fire front. The purpose of the simple fire model is to test the performance of ensemble filters on equations with the type of nonlinear behavior that can be expected in a fire, and to serve as the background for

the development of further fire models, with more realistic features such as, e.g., multiple species of fuel and modeling of the reaction intensity as a separate variable from the temperature. Such improved PDE based models will be calibrated against the existing empirically calibrated model, and they will be implemented using parallel Finite Elements and coupled with the weather model.

5 Ensemble Kalman Filter

In the EnKF approach, the model state is a probability distribution, represented by an ensemble of simulations, and the data probability distribution is represented by the measurement values and the associated error estimates. The model state is then updated using the Bayes theorem assuming that the probability distributions are approximately normal, which makes the filter much more efficient.

Consider a simulation state vector u , a measurement vector y , and an observation function h . The value $h(u)$ is what the measurements should be if the simulation and the measurements were accurate. Of course in reality always $h(u) \neq y$, so one looks for u^a , called the analysis, so that magnitudes of the residual $y - h(u^a)$ and of the change in the model state $u - u^a$ are in some sense proportional to the uncertainty in the measurements y and in the simulation state u , respectively. The uncertainty of the measurements is assumed to be known. The uncertainty of the simulation u is estimated by running an ensemble of simulations from initial conditions with random perturbations and observing how far apart they spread. Under the assumption that the probability distributions are normal, the EnKF formulas [5] follow from the Bayes theorem, which coincides in this case with the least squares estimates.

Our first EnKF implementation follows [13]. Further improvements in the EnKF sampling strategies and implementations of the reduced rank analysis scheme based on eigenvalue and singular value techniques have been gaining popularity recently [5, 14]. However, with a few exceptions, e.g., [15], modern efficient eigenvalue and singular value solvers are apparently not well known in the data assimilation community. We are examining the use of the eigensolver LOBPCG [16] and its modifications for low-rank approximation of covariance matrices in the ensemble filter and to reduce the ensemble size by an improved generation of the initial ensemble. Modifications of the EnKF for distributions that are not even approximately normal were presented, e.g., in [17]. The modular structure of our software system allows an easy replacement of the ensemble filter and experiments with various filters, without changing the other software components.

The EnKF is known to work well in oceanography and meteorology; however, its application even to the simple fire model (Sec. 4) fails. The reason is that the simulations in the ensemble become spatially rough and randomly exceed the ignition temperature, then soon the whole ensemble is on fire everywhere. Therefore, we have developed a special filter [18], which controls the value of the spatial gradient of the solution by adding to the least squares function to be

minimized a quadratic form of the difference between the gradient of the solution and the gradient of the mean of the ensemble members prior to the update. This filter was observed to result in low residuals of the analysis ensemble with no spurious fires.

6 Hidden Model Updating

The EnKF methodology assumes that the complete state of the model is modified by the filter, and any modified state is valid. Unfortunately, there are various dependencies between the state variables and constraints on which states are meaningful. For example, the weather model consists of several loosely coupled nested layers with complicated dependencies between them [4]. Just describing these dependencies and constraints outside of the model code would mean bringing much of the physics to the interface and duplicating a significant part of the code. Often, the constraints implemented in the model cannot be described even in principle because the consistency between variables in the model may be enforced only over time by techniques such as nudging [19]. For example, instead of imposing the constraint $u = v$, the evolution equation $\frac{\partial u}{\partial t} = F(u, \dots)$ for u may be modified to read

$$\frac{\partial u}{\partial t} = F(u, \dots) - c(u - v), \quad c > 0. \quad (1)$$

Difficulties with the validity of updated states are not widely discussed in the literature; [20] is a notable exception. The common solution seems to be to update some of the variables and simply hope for the best, that the change will propagate to all variables properly after some time.

In addition, it is not desirable to update all variables because there are too many of them. The part of the system state that is not being updated is hidden from the Kalman filter, contrary to its mathematical derivation. We propose to update only a small part of the state and to improve the chance of the hidden variables to settle to their natural values by a combination of nudging and the EnKF. We will compute the new value of the state vector u in each simulation by an ensemble Kalman algorithm, then restart each simulation from an earlier time and impose the new value as v in (1).

7 Assimilation of Airborne Images and Sensor Streams

Airborne near and thermal infrared images can be processed to show the presence of fire quite reliably [21, 22]. The challenges in assimilation of images include the large number of pixels, matching geographical coordinates of the pixels to the coordinates in the model, and recognizing that the errors are correlated because different cameras may be used to image the scene. The ongoing development of algorithms for extracting fire temperature and predicting the direction of fire propagation will provide further data for comparison to the model output [23,

24]. Image processing also allows the transmission of only the pertinent pixels containing a fire signal, greatly reducing bandwidth requirements, while taking into account that false positive fire detections are more likely than false negatives.

Ground sensor data streams that can provide direct fire measurement are now available, but they also must be appropriately matched to the model time step [25]. Techniques used in this area include averaged time derivatives or Fourier coefficients in time.

8 Weather Data Input

Assimilation of weather data requires creating an observation function h such that for a weather model state u , $h(u)$ are the same physical variables and at the same points in space as the available data y . This process involves the transformation of physical variables and units and spatial interpolation. The separation of the observation function into linear and nonlinear terms should be also given, as well as error bounds for the weather data. The error bounds are known from instrument accuracy or from the statistics of the ensembles used for the forecasts.

These physical variables include temperature, pressure, wind speed and direction, and atmospheric humidity. They come from weather data sources that range from three-dimensional weather model analyses and forecasts to point-based weather station data. These are delivered in near real time via the NOAA-PORT satellite broadcast.

9 Parallel Implementation

The code under development will run using coarse grain parallelism. A master controls a very large number of clients that have two tasks: (1) run a weather and wildfire simulation a short period of time, and (2) perform the EnKF update with the weather and wildfire state from (1). The ensemble filter update only consists of matrix-matrix multiplications (dense and sparse). Parallel iterative equation solvers and eigensolvers are also used with distributed data on the clients.

As we incorporate the WRF weather model [7] into our code, we will use its highly tuned parallel implementation. This will allow the weather simulation to execute on a group of nodes. The current wildfire model is strictly serial. Parallelization of the wildfire model consists of replacing the current model with a new, parallel computer friendly PDE based model.

10 Web Based Interface and Visualization

We currently dump physical fields into files out for visualization and/or analysis during the simulation at about 30 second intervals of model time. The 3D fields

are the wind velocity components, buoyancy, pressure perturbation, cloud microphysics fields (water vapor, cloud water, rain, etc.), and smoke concentration. The 2-D fields are the fire heat and water vapor fluxes at each fuel cell and the fuel remaining. These fields can be visualized, e.g., with IDL or Matlab.

In future, we will visualize simulation results on a web site using Java applets. The software structure will support real time visualization with the data streamed to the client, near real-time visualization, where the client periodically retrieves the output of an on-going simulation and examines the process of the spreading fire, as well as offline playback. The same software structure will support clients from powerful workstations, which can provide full 3D visualization for offline analysis, to PDAs and cell phones, which may be limited to 2D animated maps or only periodically refreshed 2D static maps. In any case, the user will be able to pause, rewind, forward, zoom in and out, switch angles, and interact with the visualization to input data.

Separate visible layers will present information such as maps, fuel, wind, fire location and intensity, and probability of burning (from the ensemble forecast). The user will be able to turn the layers on and off, change their order, or modify their transparency. Also, transparency may be used to visualize the probability of the fire, while the color can represent fire intensity.

Acknowledgement

This research has been supported by the National Science Foundation under grants ACI-0325314, 0324989, 0324988, 0324876, and 0324910.

References

1. Mandel, J., Chen, M., Franca, L.P., Johns, C., Puhalskii, A., Coen, J.L., Douglas, C.C., Kremens, R., Vodacek, A., Zhao, W.: A note on dynamic data driven wildfire modeling. In Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J., eds.: *Computational Science - ICCS 2004*. Volume 3038 of *Lecture Notes in Computer Science*. Springer (2004) 725–731
2. Clark, T.L., Coen, J., Latham, D.: Description of a coupled atmosphere-fire model. *Intl. J. Wildland Fire* **13** (2004) 49–64
3. Rothermel, R.C.: A mathematical model for predicting fire spread in wildland fires. *USDA Forest Service Research Paper INT-115* (1972)
4. Clark, T.L., Farley, R.D.: Severe downslope windstorm calculations in two and three spatial dimensions using anelastic interactive grid nesting: A possible mechanism for gustiness. *J. of the Atmospheric Sciences* **41** (1984) 329–350
5. Evensen, G.: The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics* **53** (2003) 343–367
6. Linn, R., Reisner, J., Colman, J., Winterkamp, J.: Studying wildfire behavior using FIRETEC. *Int. J. of Wildland Fire* **11** (2002) 233–246
7. Patton, E.G., Coen, J.L.: WRF-Fire: A coupled atmosphere-fire module for WRF. In: *Preprints of Joint MM5/Weather Research and Forecasting Model Users' Workshop*, Boulder, CO, June 22–25. NCAR (2004) 221–223

8. Finney, M.A.: FARSITE: Fire area simulator-model development and evaluation. Res. Pap. RMRS-RP-4, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 47 p., <http://www.farsite.org> (1998)
9. Coen, J.L.: Simulation of the Big Elk Fire using using coupled atmosphere-fire modeling. *International J. of Wildland Fire* **14** (2005) in print
10. Mallick, M., Kirubarajan, T., Arulampalam, S.: Out-of-sequence measurement processing for tracking ground target using particle filters. In: *Aerospace Conference Proceedings, 2002. Volume 4., IEEE (2002)* 4-1809-4-1818
11. Orton, M., Marrs, A.: A Bayesian approach to multi-target tracking and data fusion with out-of-sequence measurements. In: *IEE International Seminar Target Tracking: Algorithms and Applications. Volume 1., IEE, London, UK (2001)* 15/1-15/5
12. Beneš, M.: Mathematical and computational aspects of solidification of pure substances. *Acta Mathematica Universitatis Comenianae. New Series* **70** (2000) 123-151
13. Burgers, G., van Leeuwen, P.J., Evensen, G.: Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review* **126** (1998) 1719-1724
14. Evensen, G.: Sampling strategies and square root analysis schemes for the EnKF. *Ocean Dynamics* (2004) 539-560
15. European Centre for Medium-Range Weather Forecasts: Integrated Forecast System (IFS) documentation (CY28r1). V. The Ensemble Prediction System (2004) <http://www.ecmwf.int/research/ifsdocs/CY28r1/Ensemble>.
16. Knyazev, A.V.: Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput.* **23** (2001) 517-541 (electronic) Copper Mountain Conference (2000).
17. Bengtsson, T., Snyder, C., Nychka, D.: Toward a nonlinear ensemble filter for high dimensional systems. *J. of Geophysical Research - Atmospheres* **108(D24)** (2003) Art. No. 8775
18. Johns, C.J., Mandel, J.: A two-stage ensemble Kalman filter for smooth data assimilation. *Environmental and Ecological Statistics* (2005) submitted
19. Kalnay, E.: *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press (2003)
20. Bertino, L., Evensen, G., Wackernagel, H.: Sequential data assimilation techniques in oceanography. *International Statistical Review* **G 71** (2003) 223-241
21. Radke, L.R., Clark, T.L., Coen, J.L., Walther, C., Lockwood, R.N., Riggan, P.J., Brass, J., Higgans, R.: The wildfire experiment (WiFE): Observations with airborne remote sensors. *Canadian J. Remote Sensing* **26** (2000) 406-417
22. Vodacek, A., Kremens, R.L., Fordham, A.J., VanGorden, S.C., Luisi, D., Schott, J.R.: Remote optical detection of biomass burning using a potassium emission signature. *International J. of Remote Sensing* **13** (2002) 2721-2726
23. Ononye, A., Vodacek, A., Kremens, R.: Improved fire temperature estimation using constrained spectral unmixing. In: *Remote Sensing for Field Users. Am. Soc. Photogram. Remote Sens. (2005) CD-ROM Proc. 10th Biennial USDA Forest Service Remote Sensing Applications Conference. Salt Lake City, UT.*
24. Vodacek, A., Ononye, A., Wang, Z., Li, Y.: Automatic estimation of direction of propagation of fire from aerial imagery. In: *Remote Sensing for Field Users. Am. Soc. Photogram. Remote Sens. (2005) CD-ROM Proc. 10th Biennial USDA Forest Service Remote Sensing Applications Conference. Salt Lake City, UT.*
25. Kremens, R., Faulring, J., Gallagher, A., Seema, A., Vodacek, A.: Autonomous field-deployable wildland fire sensors. *International J. of Wildland Fire* **12** (2003) 237-244