Data-enabled modeling of wildfire smoke transport

Intellectual Merit. We propose to implement weak constraint four dimensional data assimilation (4DVAR) to combine wind field, emission and concentration data with a partial differential equation that describes transport of $PM_{2.5}$ concentrations generated by wildfire smoke. Data will be from a range of numerical weather prediction (NWP) models, including NCEP and EMCWF, smoke emission models from NOAA and US Forest service, and concentration data from EPA.

The computational cost of weak constraint 4DVAR can be prohibitive so we will use the representer method [6] to reduce the search space for the optimal estimates from the state space to the data space. Implementation of the representer method will be improved by developing algorithmic advances for adaptive mesh refinement (AMR) in parallel with storage and checkpointing of adjoints. We will focus on global scale simulations and reliable estimates will be obtained using a high resolution, conservative finite volume scheme. We will improve approximation of the Dirac delta distributions, appearing in the adjoint method, with a new formulation inspired by the Immersed Boundary Method. This formulation of the Dirac delta function also has the potential to improve the efficiency of weak constraint 4DVAR when the covariance matrices have off diagonal elements. In addition to obtaining optimal $PM_{2.5}$ concentration estimates and their resulting covariances, we will optimize a cost function to obtain optimal wind field and emission estimates. These estimates will fit observations to the transport model within specified error covariances. These covariance matrices will satisfy by a χ^2 test with accurate degrees of freedom that are a function of the number of data.

Broader Impacts. The data assimilation procedure will quantify the uncertainty in operational smoke forecasts from historical wildfire events which can be used to estimate uncertainty in smoke forecasts for future wildfire events. Uncertainty estimation is especially important for predictions in rural and downstream communities where observations are not abundant. The methods we develop will be useful to other applications that require data assimilation with complicated physics over large temporal and spatial scales, e.g. transport problems such as those from pollution and water vapor.

Graduate and undergraduate students in an EPSCoR state will develop critical data skills in atmospheric fields through research experiences that include problem solving, critical thinking, teamwork, collaboration, data management, and computational skills. Students from PI Mead's data assimilation in climate course at the African Institute for Mathematical Sciences (AIMS) in Rwanda have been recruited to work on this project. The undergraduate research participants will be recruited in cooperation with the Institute for Inclusive and Transformative Scholarship's NSF-funded LSAMP program (Louis Stokes Alliance for Minority Participation) at Boise State University. The students will be encouraged and funded to attend research and career development conferences and workshops locally, nationally and internationally. As required by the Computing PhD program, the graduate students will each submit multiple lead author papers to peer reviewed journals.

1 Smoke3d: A Wildfire Smoke Transport Model

In the Western United States, Australia and many other parts of the world, wildfires are now a seasonal occurrence. Wildfires emit pollutants into the air creating poor air quality that is hazardous to people's health and the environment [36]. Fine particulate matter with diameters 2.5 micrometers and smaller, referred to as $PM_{2.5}$, is the pollutant of principle concern in wildfire smoke because it can enter the lungs and bloodstream.

We will simulate long-range regional transport of wildfire smoke by estimating $\mathrm{PM}_{2.5}$ concentrations $q(\mathbf{x},t)$, $\mathbf{x} = [x,y,z]^T$, in the near surface with a 3d transport model on an Eulerian grid $[\mathbf{x}_a, \mathbf{x}_b]$:

$$\begin{gathered} q_t(\mathbf{x},t) + \nabla \cdot u(\mathbf{x},t)q(\mathbf{x},t) - \nabla \cdot (\mathbf{K} \cdot \nabla q(\mathbf{x},t)) + cq(\mathbf{x},t) = Q(\mathbf{x},t) \\ q(\mathbf{x},0) = q^0(\mathbf{x}), \end{gathered} \tag{1}$$

with $q^0(\mathbf{x})$ the initial concentration and subject to periodic or outflow boundary conditions. The source term, $Q(\mathbf{x},t)$, denotes the rate of mass influx per unit volume per unit time into the smoke plume (e.g. $\mathrm{kg\,m^{-3}\,s^{-1}}$). It represents the dynamics of near-source smoke plumes, including fuels and fire behavior. Dispersion by ambient wind is represented by $u(\mathbf{x},t)$, turbulent diffusion is represented with 3d diffusivity constants $\mathbf{K} = \mathrm{diag}(K_x, K_y, K_z)$, and fall out due to rain or temperature has constant decay coefficient c.

If inputs $u(\mathbf{x},t), \mathbf{K}, c$ and $Q(\mathbf{x},t)$ are represented accurately, (1) contains all of the factors that contribute to wildfire smoke behavior. This includes, as described in [22], (i) the emission source, (ii) plume rise, (iii) transport and dispersion by ambient wind, and (iv) chemical reactions in the smoke. Multiple operational smoke forecasting systems incorporate all of these factors into their forecasts. For example the BlueSky Framework [29] from the US Forest Service and HRRR-Smoke from NOAA. However, uncertainty in these predictions due to data availability and accuracy has not been well studied.

We will develop an analysis model to simulate near surface smoke caused by wildfires as modeled by (1). The starting point for our model is the Ash3d code, developed by the United States Geological Survey (USGS) for transport and dispersion of ash following volcanic eruptions[50]. In the Ash3d model, volcanic ash is passively transported via downwind advection and turbulent diffusion using a second-order, mass conserving, finite volume scheme. A key component of the Ash3d code is a library for reading obtaining wind data any one of a number of numerical weather prediction (NWP) models and interpolating the data to numerical grids. Using these . wind fields, ash, or in our case, wildfire smoke, is advected with this time-varying wind data. Options for NWP data include the European Community Medium-Range Weather Forecasting (ECMWF) and NOAA's National Centers for Environmental Prediction (NCEP) models. In addition to these data, we will test our methods by running simple simulations with estimates of the wind fields that assume a constant wind direction with speed varying by elevation according to a Gaussian profile.

While Ash3d has many features that we will be able to take full advantage of, we will need to make some modifications to account for source term models for smoke plume emission rate. The source term for smoke from wildfire emissions will depend on the burn area, fuel consumption and an emission factor that reflects a broad range of fire behavior and topography. We will use emission estimates from BlueSky and HYSPLIT-based Emissions Inverse Modeling System (HEIMS) recently developed through ARL. We will also use an initial estimate of fire emissions by specifying a series of individual plumes, decomposed as a sum of instantaneous sources, as described in [54]. The vertical emission profile (plume rise) will first be characterized as a linear distribution, and then the Suzuki distribution in [50] will be modified to characterize wildfire smoke plumes. These initial estimates will be updated with data during data assimilation so that we can blend observations with existing physical knowledge.

In this proposal, we will refer to this modified version of Ash3d adapted to smoke modeling as Smoke3d.

Task 1. *Include appropriate models for smoke source terms in Smoke3d, and model initial wind direction as constant with varying speed, rather than NWP data. In addition, develop a source model that is the sum of plumes from both linear and Suzuki distributions. Investigate choices* *of fall out and diffusion constants c, $\mathbf{K}$ and other constants such as the height of the plume, the start time and duration of the plume, that lead to realistic behavior of wildfire smoke.*

2 Smoke Transport Data

There is an identified need and considerable effort to produce reliable data that supports smoke forecasting systems [37]. The Fire and Smoke Model Evaluation Experiment (FASMEE) [49] is a large scale inter-agency effort for effective collection, warehousing and sharing of data for operational fire and smoke models. Our proposed simulation of the long range transport of smoke requires meteorological, emission and concentration data. We will use wind data from NWP models available through Ash3d, Emission data from BlueSky [29] and HRRR-Smoke, and concentration data from multiple sources including the EPA AirNow's centralized data system. We will assimilate these data using 4DVAR so that the data can inform the transport model (1). Once data assimilation is complete we will also provide uncertainty estimates for these data.

Wind field data. A key component of our modeling effort is the realistic transport of wind in forecasted wind fields. While wind fields are generally available for download on servers from NOAA and other sources, post-processing the data into a form that can be used by the model equations can be a major challenge. To obtain this data for the present model, we will use the MetReader library available in Ash3d. PI Calhoun will draw on her experience developing a parallel version of Ash3d in ForestClaw to incorporate the MetReader library into Smoke3d.

Emission data. BlueSky is a model developed by the Forest Service to predict fires. It also produces emission estimates for chemical transport models on a spatial and temporal Eulerian grid when combined with the Sparse Matrix Operator Kernel Emission Processor (SMOKE). SMOKE uses the Fire Emission Production Simulator (FEPS) for the buoyancy flux in plume rise. These emission estimates will be used for the forcing term ($Q(\mathbf{x}, t)$) in the transport model (1). Emission estimates from NOAA's High- Resolution Rapid Refresh Smoke model (HRRR-Smoke) will also be assimilated into (1). HRRR-Smoke uses satellite and sensor data, and a forecast model based on the Weather Research and Forecasting (WRF) coupled with chemistry (WRF-chem). The emissions are derived from satellite-measured fire radiative power (FRP).

In operational forecast systems, emission estimates from BlueSky and HRRR-Smoke are typically used with HYSPLIT [51] to simulate smoke transport. HYSPLIT estimates air particle trajectories in a Lagrangian framework and calculates concentrations on a fixed grid. It has been widely used to simulate pollutant events, including volcanic ash, smoke from wildfires, radioactive nuclei dispersion, and emissions of anthropogenic pollutants. BlueSky and HRRR-Smoke can produce significantly different $\text{PM}_{2.5}$ concentration predictions [20]. Uncertainties are estimated by running an ensemble of models with a variety of change options and parameters. Alternatively, here we will simulate smoke transport using (1) and incorporate uncertainties in a 4d variational data assimilation (4DVAR) framework as described in Section 3.

In [25], efforts are described to invert HYSPLIT for wildfire emissions using satellite data and compare it to BlueSky emission data. The results provide accurate fire emission inputs for smoke forecast systems. ARL is developing this HYSPLIT-based Emissions Inverse Modeling System (HEIMS) to estimate wildfire emissions from the transport and dispersion of smoke plumes inferred from NOAA GOES satellite observations. HEIMS also uses a 4DVAR data assimilation scheme to assimilate the satellite data into HYSPLIT. We will assimilate HEIMS emission data near the smoke plume into the transport equation (1) to estimate $\text{PM}_{2.5}$ concentration on a global scale.

Task 2. *Obtain and post-process emission data from BlueSky, HRRR-Smoke and HEIMS in a form that can be used by the model equations (1) for multiple wildfire seasons and estimate their uncertainties.*

Task 3. *Run Smoke3d with NWP data using the Metreader library and emission (source) data from Task 2 and identify fall out and dispersion constants* $c, \mathbf{K}$ *that produce results similar to previous wildfire smoke events.*

Concentration data. The accuracy of model predicted surface $\mathrm{PM}_{2.5}$ is typically determined using ground-based $\mathrm{PM}_{2.5}$ monitors. These observations are sparse and especially in rural areas where transported smoke reaches the lower layers of the atmosphere and impacts the air quality, it is difficult to determine model accuracy. Surface observations will be supplemented with concentrations that are inferred from aerosol Optical depth (AOD) and aerosol index (AI) satellite data see e.g. [21]. This near global, high resolution data is a measure of the extinction of light from aerosols in the atmosphere. $\mathrm{PM}_{2.5}$ concentration is estimated from AOD and AI data using weighted regression and the uncertainties in these calculations will be incorporated into our 4DVAR system. Surface and satellite $\mathrm{PM}_{2.5}$ data are available from a number of sources including AirNow, EPA, NOAA and US Forest Service.

Task 4. *Obtain and post-process* $PM_{2.5}$ *concentration data from AirNow, EPA, NOAA and US Forest Service for multiple wildfire seasons.*

3 Weak Constraint 4DVAR with Representers

We propose to assimilate wind fields, emission and $\mathrm{PM}_{2.5}$ concentration data directly into the transport equation (1). This is a more thorough assimilation process than comparing the concentration output from (1) to observations of $\mathrm{PM}_{2.5}$ concentration. For example, in [21] they used a two-stage Bayesian ensemble model to combine EPA ground measurements and remote sensing data with a chemical transport model simulation (CTM). They used Markov Chain Monte Carlo (MCMC) simulations to find weights on $\mathrm{PM}_{2.5}$ estimates in Colorado from April to September over the years 2011–2014. Roughly speaking, methods that use this approach effectively use regression and find weights w_1, w_2, w_3 that fit model simulations $\mathbf{q}_{CTM}$ to ground measurements $\mathbf{q}_{EPA}$ and satellite data $\mathbf{q}_{SAT}$, according to

$$\mathbf{q}_{CTM} = w_1 + w_2\mathbf{q}_{EPA} + w_3\mathbf{q}_{SAT}.$$

Alternatively, we will use 4DVAR and allow the data to alter the dynamics in (1), as illustrated on a simple example in Figure 3. Altering the dynamics is not possible in most 4DVAR methods because they use the strong constraint approach which does not assume there are any errors in the governing equation (1), only its initial estimate $q^0(\mathbf{x})$. Strong constraint 4DVAR works well for short term forecasts, but here we propose to estimate concentrations on a global scale. Therefore we assume there are errors in the governing equations due to parameterizations, unresolved processes, and forcing. We will implement weak constraint 4DVAR which results in more accurate forecasts, but it can be computationally prohibitive. Therefore we propose use use the representer method [5] with adaptive mesh refinement (AMR) in parallel to make the method computationally feasible.

We define a linear operator L representing $\nabla \cdot u(\mathbf{x},t)q(\mathbf{x},t) - \nabla \cdot (\mathbf{K} \cdot \nabla q(\mathbf{x},t)) + cq(\mathbf{x},t)$ in (1) and re-write it as

$$\begin{aligned} q_t(\mathbf{x},t) &= L[q(\mathbf{x},t)] + Q(\mathbf{x},t) + f(\mathbf{x},t) && (2) \\ q(\mathbf{x},0) &= q^0(\mathbf{x},t) + i(\mathbf{x}) && (3) \end{aligned}$$

with errors f and i in the dynamics and initial conditions. We assume there are exact periodic or outflow boundary conditions with no errors. In addition assume we have an initial estimate of the emission Q along with data for it at points $(\mathbf{x}_Q, \mathbf{t}_Q)$ and a limited number of concentration data at points $(\mathbf{x}_q, \mathbf{t}_q)$. These all have errors or uncertainties associated with them so we write

$$\begin{aligned} Q(\mathbf{x},t) &= Q^0(\mathbf{x},t) + E(\mathbf{x},t) && (4)\\ \mathbf{d}_Q &= H_Q Q(\mathbf{x}_Q, \mathbf{t}_Q) + \boldsymbol{\epsilon}_Q && (5)\\ \mathbf{d}_q &= H_q q(\mathbf{x}_q, \mathbf{t}_q) + \boldsymbol{\epsilon}_q && (6)\end{aligned}$$

where H_* is an observation operator that transforms either $Q(\mathbf{x},t)$ or $q(\mathbf{x},t)$ into observation equivalents while $E(\mathbf{x},t)$, $\boldsymbol{\epsilon}_Q$ and $\boldsymbol{\epsilon}_q$ are corresponding errors.

We assume normally distributed errors $f, i, E, \boldsymbol{\epsilon}_Q$ and $\boldsymbol{\epsilon}_q$ each with covariances $C_f(\mathbf{x},t)$, $C_i(\mathbf{x})$, $C_E(\mathbf{x},t)$, $\mathbf{C}_{\epsilon_Q}$ and $\mathbf{C}_{\epsilon_q}$. Weak constraint 4DVAR minimizes these errors in a least squares sense and produces the maximum likelihood estimate by minimizing the cost function

$$\begin{aligned} \mathcal{J} = &\int_0^T \int_\Omega \int_0^T \int_\Omega f(\mathbf{x},t) C_f^{-1}(x,t,\mathbf{x}',t') f(\mathbf{x}',t')\ d\mathbf{x}'dt'd\mathbf{x}dt && (7)\\ &+ \int_0^T \int_\Omega \int_0^T \int_\Omega E(\mathbf{x},t) C_E^{-1}(x,t,\mathbf{x}',t') E(\mathbf{x}',t')\ d\mathbf{x}'dt'd\mathbf{x}dt && (8)\\ &+ \int_\Omega \int_\Omega i(\mathbf{x}) C_i^{-1}(\mathbf{x},\mathbf{x}') i(\mathbf{x}')\ d\mathbf{x}'d\mathbf{x}\ +\ \boldsymbol{\epsilon}_Q \mathbf{C}_{\epsilon_Q}^{-1} \boldsymbol{\epsilon}_Q\ +\ \boldsymbol{\epsilon}_q \mathbf{C}_{\epsilon_q}^{-1} \boldsymbol{\epsilon}_q && (9)\end{aligned}$$

where Ω is the spatial domain.The first two inverses are defined in a convolutional sense where

$$\int_0^T \int_\Omega C_f^{-1}(x,t,\mathbf{x}',t') C_f(\mathbf{x}',t',\mathbf{x}'',t''))\ d\mathbf{x}'dt' = \delta(\mathbf{x}-\mathbf{x}'')\delta(t-t'') \qquad (10)$$

where $\delta(\mathbf{x})$ is the Dirac delta function. A similar definition holds for C_i^{-1} while $bfC_{\epsilon_Q}^{-1}$ and $bfC_{\epsilon_q}^{-1}$ are matrix inverses that are guaranteed to exist since they are SPD covariance matrices.

If we use the calculus of variations to minimize the cost function with respect to q, the result is the following coupled Euler-Lagrange system of equations [6]

$$\begin{aligned} -\lambda_t(\mathbf{x},t) &= \left[\frac{\partial L}{\partial q}[\hat{q}(\mathbf{x},t)]\right]^T \lambda(\mathbf{x},t) - \sum_{m_1=1}^{M} \sum_{m_2=1}^{M} (\mathbf{C}_{\epsilon_q}^{-1})_{m_1 m_2} (H_q \hat{q}(\mathbf{x}_{m_1}, t_{m_1}) - (d_q)_{m_1}) H_q^T \delta(\mathbf{x}-\mathbf{x}_{m_2}) \delta(t - t_{m_2})\\ \lambda(\mathbf{x},T) &= 0\\ \hat{q}_t(\mathbf{x},t) &= L\hat{q}(\mathbf{x},t) + Q(\mathbf{x},t) + C_f \bullet \lambda(\mathbf{x},t)\\ \hat{q}(\mathbf{x},0) &= i(x) + C_i \circ \lambda(\mathbf{x},0) \end{aligned} \qquad (11)$$

with $\hat{q}(\mathbf{x},t)$ minimizing the cost function and $\lambda(\mathbf{x},t)$ the adjoint variable defined by

$$\lambda(\mathbf{x},t) = \int_\Omega W_f(x,t,\mathbf{x}',t') f(\mathbf{x}',t')\ d\mathbf{x}'dt'.$$

The convolution is defined

$$C_f \bullet \lambda(\mathbf{x},t) = \int_\Omega C_f(x,t,\mathbf{x}',t') \lambda(\mathbf{x}',t')\ d\mathbf{x}'dt'$$

with similar definition for $C_i \circ \lambda(\mathbf{x},0)$.

The coupled system of equations (11) requires λ to be solved for backwards in time and the optimal estimate $\hat{q}$ forward in time. We will simplify this system of equation by using the representer method [6]. In that method the optimal solution is a combination of an initial estimate $q_F(\mathbf{x},t)$ and a linear combination of representer functions $r_m(\mathbf{x},t)$, $1 \le m \le M$:

$$\hat{q}(\mathbf{x},t) = q_F(\mathbf{x},t) + \sum_{m=1}^{M} \beta_m r_m(\mathbf{x},t). \tag{12}$$

Substituting this into the Euler-Lagrange equations we get a decoupled system of equations

$$\begin{aligned} -(\alpha_m)_t(\mathbf{x},t) &= L^T \alpha_m(\mathbf{x},t) + H_q^T \delta(\mathbf{x}-\mathbf{x}_m)\delta(t-t_m) \\ \alpha_m(x,T) &= 0 \\ (r_m)_t(\mathbf{x},t) &= L r_m(\mathbf{x},t) + C_f \bullet \alpha_m(\mathbf{x},t) \\ r_m(\mathbf{x},0) &= C_i \circ \alpha_m(\mathbf{x},0) \end{aligned} \tag{13}$$

We first solve for $\alpha_m(x,t)$ backward in time and then use it to solve forward in time for r_m, $1 \le m \le M$. We find the representer coefficients β_m by substituting (12) back into the Euler-Lagrange equations. The result is M equations for unknowns $\boldsymbol{\beta} = [\beta_1, \dots, \beta_m]^T$ which can be written in matrix notation as

$$\left(\mathbf{R} + \mathbf{W}_\epsilon^{-1}\right)\boldsymbol{\beta} = \mathbf{d} - \mathbf{q}_F \tag{14}$$

where $\mathbf{R}$ is the representer matrix with representer functions evaluated where there are data, i.e. $\mathbf{R}_{m_1 m_2} = r_{m_1}(\mathbf{x}_{m_2}, t_{m_2})$. This gives the optimal estimate

$$\hat{q}(\mathbf{x},t) = q_F(\mathbf{x},t) + (\mathbf{d}-\mathbf{q}_F)^T(\mathbf{R}+\mathbf{C}_{\epsilon_q})^{-1}\mathbf{r}(\mathbf{x},t) \tag{15}$$

with $\mathbf{r}(\mathbf{x},t)$ the vector of the m representers evaluated at $(\mathbf{x},t)$.

When we substitute (12) into the Euler-Lagrange equations (11) we get that

$$\lambda(\mathbf{x},t) = \sum_{m=1}^{M} \beta_m \alpha_m(\mathbf{x},t).$$

In addition we get estimates for the errors f and i, namely

$$f(\mathbf{x},t) = C_f \bullet \lambda(\mathbf{x},t) \quad \text{and} \quad i(\mathbf{x}) = C_i \circ \lambda(\mathbf{x},0). \tag{16}$$

The representer method finds the optimal state estimate by solving the linear transport model backward in time M times for $\alpha_m(\mathbf{x},t)$ and forward M times for $r_m(\mathbf{x},t)$. This makes weak constraint 4DVAR more computationally feasible because the optimal adjoint in the state space is determined by the optimal representer coefficients in the data space, which is much smaller than the state space. In addition, the representer coefficients can be computed in parallel since they are independent of each other.

Task 5. *Assimilate $PM_{2.5}$ concentration data into the transport model (1) using weak constraint 4DVAR with representers and estimate the errors in the dynamics and initial conditions.*

We can also obtain optimal estimates of the emissions $Q(\mathbf{x},t)$ that result by minimizing the cost function (9) with respect to Q. Similarly we will assume there are errors in the wind field data $u(\mathbf{x},t)$ diffusivity and fall out constants $\mathbf{K}$ and c and add their weighted errors to the cost function, i.e. we will assume

$$\begin{aligned} u(\mathbf{x},t) &= u^0(\mathbf{x},t) + E_u(\mathbf{x},t) && (17) \\ \mathbf{K} &= \mathbf{K}^0 + \Gamma && (18) \\ c &= c^0 + \gamma && (19) \\ \mathbf{d}_u &= H_u u(\mathbf{x}_u, \mathbf{t}_u) + \boldsymbol{\epsilon}_u. && (20) \end{aligned}$$

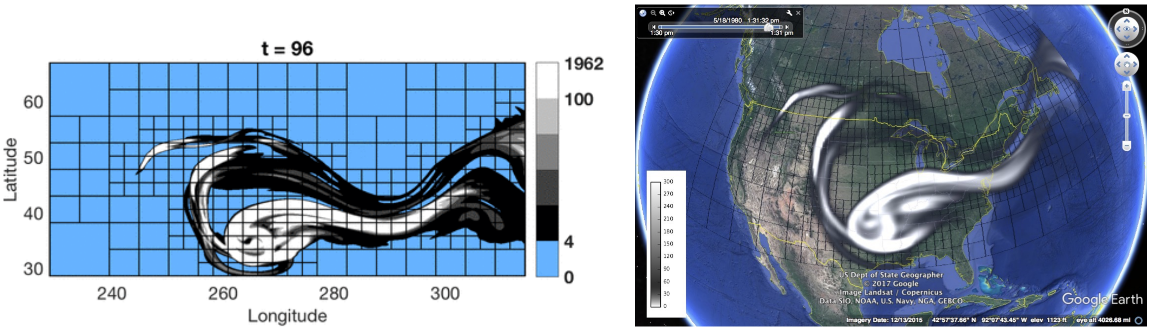
t = 96
Latitude
60
50
40
30
240
260
280
300
Longitude
1962
100
4
0
US Dept of State Geographer
© 2017 Google
Image Landsat / Copernicus
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Google Earth

Figure 1: Simulations of volcanic ash transport in the aftermath of the 1981 eruption of Mt. St. Helens (Washington State, USA). The simulations were done using the parallel, adaptive version of Ash3d (USGS, Schwaiger et al.), implemented in the ForestClaw software package. The left figure shows the adaptive mesh following the ash trajectory. The right figure shows the images as overlays in Google Earth. Image credit : D. Calhoun, ©Google Earth Image.

These errors $E_u(\mathbf{x}, t)$, Γ, γ and $\boldsymbol{\epsilon}_u$ will have corresponding specified covariances as described in Section 5. The weighted errors will be added to the cost function (9) which will be minimized with respect to q, Q, u, $\mathbf{K}$ and c using the calculus of variations.

Task 6. *Derive the Euler-Lagrange system that results when optimizing for $Q(\mathbf{x}, t)$, $u(\mathbf{x}, t)$, K and c and formulas for their errors (similar to (16)).*

4 Efficient numerical solution of the forward and adjoint equations

Like many hazardous geophysical flows, smoke transport takes places over large spatial domains and for long times frames of days and weeks. To solve the forward and adjoint equations as efficiently as possible over these large spatial and temporal domains, we will use high resolution mass conserving finite volume methods with dynamic adaptive mesh refinement. PI Calhoun is the principle developer of the ForestClaw code, a parallel adaptive mesh refinement library for hyperbolic conservation laws. In a recent collaboration with the developers of Ash3d, Calhoun has ported the original serial, single grid Ash3d code to the parallel, adaptive ForestClaw platform. See Figure 1 for results of this effort. This parallel, adaptive version of Ash3d will be the starting point for the development of Smoke3d.

4.1 Direct and iterative solution approaches

To solve the Euler-Lagrange equations presented in (11), we numerically integrate (using methods described below) the equations in (13) to get pairs (α_m, r_m) of adjoint and representer functions. In a direct approach, we form the matrix $\mathbf{R}$ in (14) explicitly by solving for M pairs (α_m, r_m). Entries $\mathbf{R}_{ij}$ in column j of $\mathbf{R}$ are obtained by interpolating the representer $\mathbf{r}_j(\mathbf{x}, t)$ to data locations $(\mathbf{x}_i, t_i)$. The system in (14) is solved directly for $\boldsymbol{\beta}$ and a final forward and backward numerical integration step using the source term

$$\sum_{m=1}^{M} \beta_m \delta(\mathbf{x} - x_m)\delta(t - t_m) \tag{21}$$

for the adjoint equation in (11) is carried out to obtain the adjoint and inverse pair $(\lambda(\mathbf{x}, t), \hat{q}(\mathbf{x}, t))$. The advantages of this direct approach is that matrix inversion using a direct method is often preferable to iterative solutions. Furthermore, the matrix $\mathbf{R}$ is symmetric positive definite

(SPD), and so we need to compute and store only half of the entries of $\mathbf{R}$, and can use direct solvers specifically designed for solving SPD systems. However, for large numbers of observables, an iterative approach to solving (14) may be preferable. To solve (14) iteratively, we do not solve for all M representer pairs, but instead compute matrix-vector multiplies $(\mathbf{R} + \mathbf{W}_\epsilon^{-1})\boldsymbol{\beta}^k$ by integrating the adjoint equation for candidate $\lambda^k(\mathbf{x}, t)$, using iterate $\boldsymbol{\beta}^k$ in (21), and solving for a candidate inverse function $\hat{q}^k(\mathbf{x}, t)$. The residual is the computed as the difference between the inverse $\hat{q}(\mathbf{x}, t)$ and the observable data. The linear system to be solved is SPD, we can use the conjugate gradient method, for example. The iterative approach will be competitive with a direct approach only if the number of iterations required to reach a desired tolerance is reliably less than the number of observables M. Several choices of pre-conditioners for this system are suggested by Bennett [5].

Task 7. *Investigate both direct and iterative approaches and determine under what conditions an iterative approach will be reliably more efficient.*

4.2 Numerical integration using the wave propagation algorithm

Key to making the solution of the Euler-Lagrange system presented in (11) tractable is a computationally efficient, low storage approach for accurately integrating the forward and backwards models to obtain adjoint/representer pairs $(\alpha_m(\mathbf{x}, t), r_m(\mathbf{x}, t))$. To address the issue of accuracy, we will use the second order, high resolution wave propagation algorithm originally developed by R. J. LeVeque and implemented in the Clawpack, AMRClaw and GeoClaw software packages [31, 7, 30, 8]. While finite volume schemes were originally developed to resolve shocks in compressible gas dynamics, they have also been shown to be very effective in resolving sharp transitions at the edges of tracer fields. In the fully unsplit mode, the wave propagation algorithm solves Riemann problems in both normal and transverse directions. Second order corrections are computed from the waves, speeds and fluctuations returned from the Riemann solver. To mitigate the presence of negative values and other non-physical extrema in the simulated tracer field, we will use standard wave limiters to suppress numerical oscillations. In practice, the wave propagation algorithm can usually be run with a CFL number close to 1 (≈ 0.8 is a typical value).

To obtain velocity values from weather data sources, we will use the open source MetReader library that is part of the Ash3d utilities. The velocity fields from this library are provided as values at the center of computational mesh cells, and so to use these values directly, rather than averaging these values to faces (as is often done), we will use the F-wave variation of the wave propagation algorithm, developed by Bale et al. [3]. Calhoun has shown that when used with the flux function $f(q) = \mathbf{u}(\mathbf{x}, t)q$, the F-wave approach accurately captures stationery waves, and restores full second order accuracy to the wave propagation algorithm, when used with cell-centered data. Furthermore, by limiting fluxes rather waves, as is done in the original wave propagation algorithm, the F-wave approach does not exhibit unphysical artifacts near sonic points (i.e. points where the velocity changes sign).

Task 8. *Develop solver engine for Smoke3d. This engine will use the F-wave approach. The engine will use the same interface as that provided by Ash3d so that other routines in Ash3d can be used.*

4.3 Integrating the representer equations

To solve the equation given in (13), we will replace the Dirac delta distributions appearing in (13) with a function amenable to numerical treatment. Using discrete representations of delta functions dates back to the original work of C. Peskin [47] and has since been used extensively in biological fluid dynamics for modeling elastic membranes immersed in fluids. Inspired by this idea, and related approaches, including the Immersed Interface Method [32, 35, 11], we model a discrete $\delta(x)$ appearing in the source term for $\alpha_m(\mathbf{x}, t)$ using the heat kernel

$$\widetilde{\delta}(\mathbf{r}) = \frac{1}{(4\pi\epsilon)^{d/2}} e^{-\mathbf{r}^2/(4\epsilon)} \tag{22}$$

where d is the dimension of the problem, and $\epsilon << 1$ is a smoothing parameter which determines the sharpness of the discrete delta function. A key property of this choice for a discrete delta function is that

$$\int_{R^d} \widetilde{\delta}(\xi)\, d\xi = 1. \tag{23}$$

With the above discrete delta function, we can now evaluate the right hand side $\delta(\mathbf{x} - \mathbf{x}_m)\delta(t - t_m)$ at any value (x, t) without the need for interpolation. Using the wave propagation algorithm, we evolve the system backwards in time to obtain $\alpha_m(\mathbf{x}, t)$. The computation of the representer $r_m(\mathbf{x}, t)$ requires one forward solve using $\alpha_m(\mathbf{x}, t)$ as a forcing term. The results showing this approach in 1d are shown in Figure 3.

In the fully general equations, the source term in the representer equation is a convolution of the covariance function $\mathbf{C}$ with the adjoint variable α_m. Depending on the covariance model chosen, this convolution may be quite expensive to compute. If the covariance model is taken to be Gaussian, then we can take advantage of fast convolution operators which reduced the operational cost from $\mathcal{O}(n^2)$ (in 2d) to $\mathcal{O}(n)$. See for example [33] and [34] for a discussion of the Fast Recursive Marching method. Underlying these fast algorithms is the non-uniform FFT, which will be needed for the adaptive meshes, we describe below.

Task 9. *Investigate the use of a discrete delta function for the adjoint equation and determine if this approach is competitive with other approaches suggested in the literature for the representer method.*

Task 10. *Investigate approaches to efficiently evaluating convolution integrals. The approach suggested by Bennett [5] and others in which a pseudo-heat equation is solved will serve as a starting point.*

4.4 Adaptive mesh refinement using ForestClaw

Solving the Euler Lagrange equations places enormous demands on computational resources. The solution to the adjoint equation must be solved to the final time before the forward problem can be solved. Furthermore, the entire time history of the adjoint equation must be stored or otherwise be made available as a source term in the forward problem. The direct and iterative methods, described above, have essentially the same computational demands per iteration or (α_m, r_m) pair, and so regardless of which approach is taken, efficient, low storage options for solving a forward and backward problem are crucial. We propose taking advantage of the highly localized features of the adjoint solution in the representer method and use adaptive meshing strategies for the solutions to adjoint/representer pairs (α_m, r_m). There has been some recent work to use the solutions to adjoint equations to guide adaptive mesh strategies when solving partial differential equations [14, 15, 44]. To our knowledge, however, relatively little research has gone into addressing the computational challenges involved in using AMR for solving the full Euler-Lagrange equations. As part of this proposal, we will address these challenges.

Adaptive mesh refinement (AMR) is a computational approach that allows us to dynamically vary the resolution of the grid so that the mesh has higher resolution in regions of the grid where the solution error is likely to be largest. Several flavors of adaptive mesh refinement are currently in use. In cell-based refinement, individual mesh cells or elements are subdivided in regions where resolution demands are the highest. Recently, researchers have developed cell-based refinement strategies in the wildfire codes fireFoam and diffusionFireFoam, both based

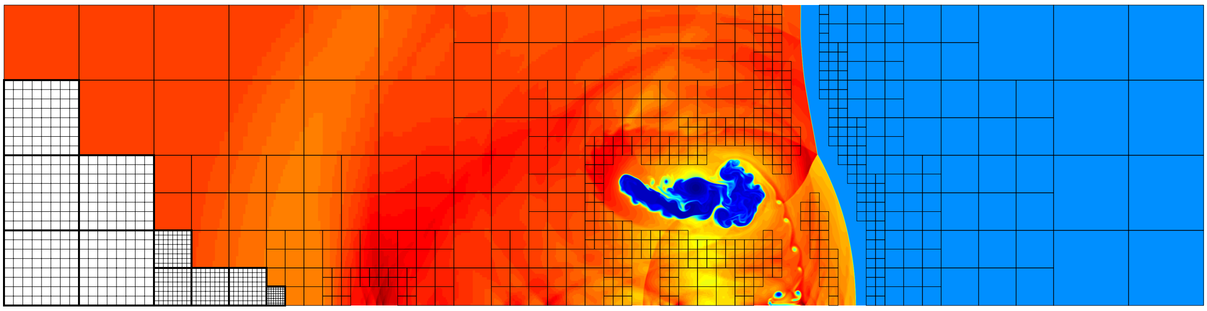

Figure 2: A slice through an axis-symmetric simulation of shock interacting with low density smoke ring using the adaptive, quadtree code ForestClaw. White quadrants in lower left corner reveal underlying fixed finite volume mesh in each leaf of the quadtree. This simulation was done using the Clawpack extension of ForestClaw. The equations solved above are the Euler equations for compressible gas dynamics and illustrate the flexibility of the solvers available using the wave propagation algorithm.

on the open source software package OpenFoam [28]. In a second "patch-based" approach to AMR, an adaptive hierarchy of Cartesian grids is created in which finer smaller grids are layered on top of larger coarser grids. Patch-based AMR is available in several widely used software packages [9, 2, 8, 46, 38, 53].

In hybrid approaches combining the ideas above, the adaptive mesh is generated by subdividing the domain recursively into quadrants or octants. The resulting mesh is a partition of the computational domain in which the solution is stored at the leaf level only. Cell-based algorithms on quadtree/octree meshes are essentially unstructured algorithms with one degree of freedom, or one element per leaf node in the tree [48, 52], while in patch-based algorithms for quadtree/octrees, an entire grid is stored at the leaf nodes [38, 55, 19, 26, 24].

ForestClaw, developed by Co-PI Calhoun and the code to be used in the work proposed here, is an example of a hybrid patch-based code based on quadtree refinement [12, 13]. Figure 2 shows a simulation of a slice through a smoke ring simulated using the adaptive quadtree/octree meshing capabilities in ForestClaw. ForestClaw uses the p4est meshing library [10], widely used among other high performance, adaptive codes, including the deal.II Finite Element Library used extensively for geophysical applications [4]. The wave propagation algorithm described above is fully implemented in ForestClaw, along with adaptive time stepping so that a constant CFL constraint can be maintained across all levels of refinement. ForestClaw makes use of the fast searches available to it from the underlying p4est meshing library to implement numerical gauges that can query the solution at any non-grid aligned gauge location. To handle the turbulent diffusion term in (1), an interface to the ThunderEgg [1] library for solving elliptic and parabolic equations on adaptively refined quadtree/octree meshes has been recently developed for ForestClaw. ThunderEgg uses a Krylov space solver (BiCGStab) with multigrid pre-conditioning based a block-Jacobi smoother and fast patch solvers.

The main challenge in using AMR for solving the Euler-Lagrange equations is that once we have solved for an adjoint $\alpha_m(\mathbf{x}, t)$, we need to solve for the representer using the adjoint as the forcing. When solving for the adjoint on adaptive meshes, the dynamically evolving mesh for the adjoint equation will not in general be the same mesh used when solving for the representers, and so the solution for the adjoint must be transferred to the grid hierarchy used by the representer solution. This procedure is similar to what is already done in ForestClaw when the solution is transferred from one mesh hierarchy to another after a dynamic regridding step, and so all of the routines for the transferring of solutions is already in place.

A second related challenge is the management of the checkpointing schedule needed for making the adjoint solution available when needed. A standard approach in adjoint methods is to store checkpoints of the adjoint solution at scheduled time increments. These "checkpointed" solutions will then be used to initialize the adjoint solution to obtain an intermediate solution in checkpoint time intervals. Libraries such as Revolve [23] exist for managing scheduling and storage of the check point solutions. However, these libraries will typically assume that each checkpointed solution requires a fixed amount of storage that can be prescribed ahead of time. With AMR, the storage costs, while much lower than what would be required for a fully resolved grid, will not be fixed in time. To take advantage of the storage savings, any checkpointing library will likely need to be modified to handle the AMR data structure. In addition, the entire mesh hierarchy, along with solution will need to be stored.

Task 11. *Develop a checkpoint strategy for efficiently storing and retrieving the AMR hierarchy and investigate whether the Revolve library can be used to schedule checkpointing. Both online and offline modes will be considered.*

4.4.1 Parallel implementation

ForestClaw is fully parallelized, and so grids generated during the dynamic regridding process can be distributed to available processors and independently updated. Preliminary results using ForestClaw to solve an adjoint problem with a adaptive hierarchy distributed to four cores on desktop machine show a dramatic speed-up (approx. 30x) over a single grid, serial run at the same effective resolution. Furthermore the storage requirements for the adaptive solution are about a third to a half of the required storage for the single uniformly resolved grid. The parallel, adaptive implementation of Ash3d in ForestClaw led to about an 8x speed-up over the original single grid, serial Ash3d code. ForestClaw scales to thousands of processors, and because all grids in octree/quadtree hierarchy are the same size, load balancing is relatively straightforward [12]. The representer method, at least when used with the direct method described above, is ideally suited for parallel computations, since each representer can be calculated independently of the others. This obvious use of task based parallelism is already explored and described in Bennett. If we use an iterative approach, we can take advantage of the data parallelism available in ForestClaw, since the matrix-vector multiply can be distributed across available processors.

Collaborators Collaborator C. Burstedde, developer of the p4est library, has agreed to provide support for p4est related tasks such as creating checkpoints of the ForestClaw hierarchy. Collaborator Hans Schwaiger will provide any technical support on using the MetReader library in Ash3d, and any other Ash3d related issues.

5 4DVAR Covariances

The transport equation given in (1) can be used for many different types of applications, such as ash cloud modeling. For wildfire smoke modeling, it describes smoke transport and dispersion by ambient wind, but local perturbations to the uplift or wind field due to a widespread fire may not be captured by the NWP wind field data. In addition, the chemical evolution of the drifting cloud and diffusivity in the planetary boundary layer are not represented well. Lastly, we rely on emission data to capture forest vegetation and fire-forest-atmosphere interactions. We account for potential omissions by assuming errors in the dynamics, initial condition, wind fields, emissions, and diffusivity and fall out constants. The weights, or covariances of these errors will direct the optimal solution and thus will be a significant aspect of the proposed work. In [27], for example, they identified how choice of error covariance matrices impacted the convergence and accuracy of 4DVAR with representers in a nonlinear nearshore ocean circulation problem.

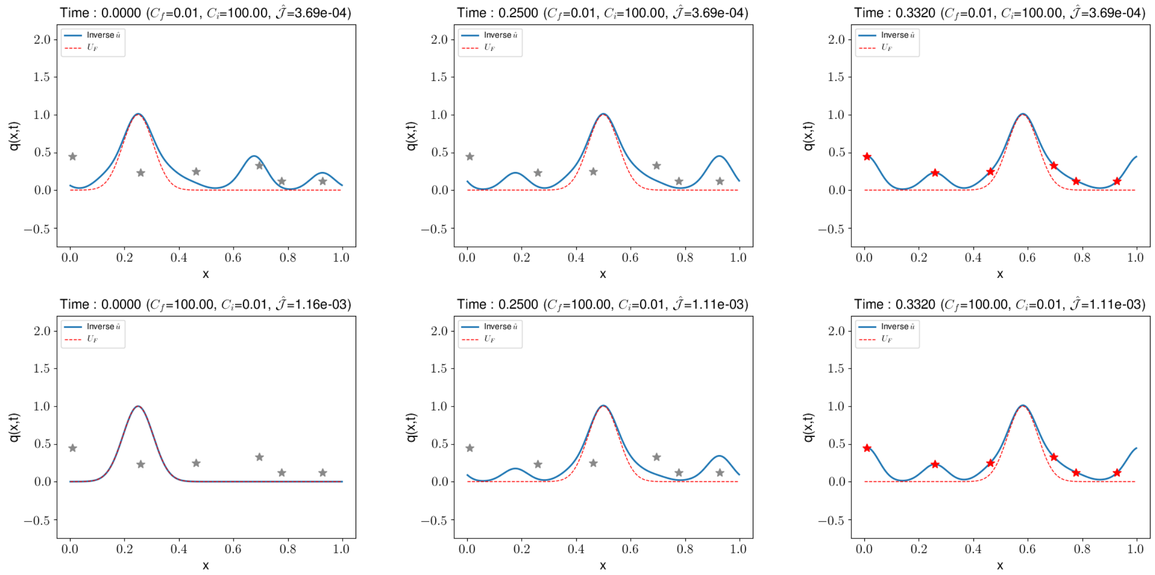

Figure 3: The top row shows the inverse solution obtained from the representer method under the assumption that the PDE model is trusted, but the initial conditions are not. The bottom row shows the inverse solution when the initial conditions are trusted, but the model is not. All six data measurements were taken at time $t = 0.331$.

A range of covariance matrices will be formed for each set of errors. This includes matrices that are constant, diagonal and some with off diagonal elements. The constants will be chosen according to the order of the numerical methods in the case of the dynamics, and estimates of standard deviations in the data as determined in Section 2. Diagonal matrices will be formed in the situation where the standard data deviation of the data changes significantly either in space or time.

Covariance matrices with off diagonal elements will be specified by

$$C(\mathbf{x}, t, \mathbf{x}', t') = \sigma^2 \exp\left\{ -\frac{1}{2}\left(\left(\frac{\mathbf{x}-\mathbf{x}'}{\Delta_x}\right)^2 + \left(\frac{\mathbf{x}-\mathbf{x}'}{\Delta_t}\right)^2 \right) \right\}. \tag{24}$$

This form of the covariance matrix requires us to specify σ, Δ_x and Δ_t. We will specify σ as we did for the constant covariance case and use Δ_* to estimate the correlation in errors across space or time. Depending on the choice of Δ_*, this form of the covariance is similar to the Dirac delta function approximation described here in (22). In each case, this function has the effect of smoothing the representers and adjoints. The convolution $C_f \bullet \lambda$ is very expensive to calculate when C_f is dense. Therefore we will investigate to potential to smooth the adjoints α_m using (22) rather than using a dense C_f to calculate the representers r_m.

Task 12. *Solve the representer equations using (22) to smooth α_m and determine if this sufficiently smooths r_m as the covariance C_f would.*

We illustrate the effect of constant covariance choice for the dynamics (1) on a simple 1d scalar advection equation in a periodic domain with a constant coefficient velocity $u = 1$. The initial conditions are a Gaussian hump, and the exact solution is a translation of the initial conditions at the prescribed velocity. We generate artificial data at uniformly distributed random spatial locations. To construct data values at these points, we perturb the exact solution with 25% normally distributed noise and use the corresponding covariance matrix to weight the data. For this illustration, we assume all data measurements were taken at the one time step.

In Figure 3, the optimal estimate obtained from the representer method is shown for constant covariances $C_f = 0.01$ and $C_i = 100.0$ in the top row, and $C_f = 100$ and $C_i = 0.01$ in the bottom row. A small constant value for C_f and large value for C_i, say $C_f \ll 1$ and $C_i \gg 1$, more strongly enforces the dynamics as a constraint and specifies that the initial condition is not very accurate. We see how this covariance information propagates in a 4DVAR simulation in the top row of Figure 3 where at time $t = 0$ the Gaussian hump is significantly altered in the optimal estimate. A large covariance for the initial condition allowed it to be altered so that the dynamics could match the noisy data at a later time, as seen in the last column at $t = 0.3320$. Alternatively in the second row where $C_f \gg 1$ and $C_i \ll 1$, the initial condition is treated as a stronger constraint and the dynamics are allowed to change. In this case, the optimal estimate at time $t = 0$ essentially stays the Gaussian hump. Even though the initial condition remained the same, the optimal estimate fit the data at $t = 0.3320$ because the dynamics were specified as a weak constraint. This error in the dynamics acted as a forcing term that guided the solution towards the measured data. Interestingly at $t = 0.25$, both simulations produced similar results.

χ^2 **Hypothesis Testing.** As seen in Figure 3 the choice of covariance will guide our optimal estimate. However, we also saw in that example that two very different choices of covariances resulted in the model fitting the data. Rather than just fitting the model to data, we will evaluate our covariance choices by a χ^2 hypothesis test [6]. The null hypothesis is the specified error covariance for the dynamics ($C_f(\mathbf{x}, t)$), initial condition ($C_i(\mathbf{x})$), initial estimates of emissions, wind fields, diffusivity and fall out constants ($C_E(\mathbf{x}, t)$,$C_{E_u}(\mathbf{x}, t)$,$\mathbf{C}_K$, C_c), and data for emissions, concentration and wind field ($\mathbf{C}_{\epsilon_Q}$, $\mathbf{C}_{\epsilon_q}$, $\mathbf{C}_{\epsilon_u}$). If our optimal estimate passes the χ^2 test described below, then we fail to reject these covariance choices. If our optimal estimate fails the χ^2 test then our model differs from the data more than we hypothesized.

It was shown in [6] that when the cost function that sums errors in the dynamics and data e.g. terms in (7) and (9), is evaluated at the optimal estimate (i.e. $\mathcal{J}(\hat{q})$) it is a χ^2 random variable with M degrees of freedom, where M is the number of data, i.e.

$$\mathcal{J}(\hat{q}) = \int_0^T \int_\Omega \int_0^T \int_\Omega \hat{f}(\mathbf{x}, t) C_f^{-1}(x, t, \mathbf{x}', t') \hat{f}(\mathbf{x}', t') \; d\mathbf{x}' dt' d\mathbf{x} dt \tag{25}$$

$$+ \int_\Omega \int_\Omega \hat{\imath}(\mathbf{x}) C_i^{-1}(\mathbf{x}, \mathbf{x}') \hat{\imath}(\mathbf{x}') \; d\mathbf{x}' d\mathbf{x} \; + \; \hat{\boldsymbol{\epsilon}}_q \mathbf{C}_{\epsilon_q}^{-1} \hat{\boldsymbol{\epsilon}}_q \tag{26}$$

$$\sim \chi^2_M. \tag{27}$$

This means that the cost function has mean M and we should expect $\overline{\mathcal{J}(\hat{q})} \approx M$. It is also shown in [6] that when M is large, the cost function is χ^2 even if the errors are not normally distributed. Therefore once assimilation is complete, the χ^2 test says that the cost function evaluated at the optimal estimate should be equal to the number of data, within the appropriate standard deviation. For example in Figure 3 the cost function $\hat{\mathcal{J}}$ is reported on each row. With the given values of the covariances the cost function at the optimal estimate is nearly zero, i.e. 10^{-3} and 10^{-4}. Since there are six measurements, the mean of the cost function is 6 and the standard deviation is approximately 3.5. In this example the choice of covariances does not pass the χ^2 test so the misfit between the data and the dynamics is bigger than we hypothesized.

Task 13. *Assume the wind fields, emission estimates and diffusivity and fall out constants are exact and identify covariance estimates $C_f(\mathbf{x}, t)$, $C_i(\mathbf{x})$ and $\mathbf{C}_{\epsilon_q}$ that pass the χ^2 test (27).*

The χ^2 test has been studied further by PI-Mead in [40, 39, 41]. There it was noted that the commonly used discrepancy principle [43] for ill-posed inverse problems is a χ^2 test, but with the incorrect degrees of freedom. The issue arises when viewing the problem stochastically, because the terms in the cost function are not all independent, and the degrees of freedom must be reduced to reflect that. Here we propose to assimilate the emissions and wind field data, in addition to concentration data, and optimize with respect to a number of variables. In this case it is an open problem to identify the appropriate degrees of freedom in the χ^2 test.

Task 14. *Formulate a cost function that assumes errors in dynamics, initial conditions, initial estimates of emissions, wind fields, diffusivity and fall out constants; and data for emissions, concentration and wind fields. Determine the degrees of freedom in the cost function when it is evaluated at the optimal estimate.*

Posterior Covariance An approximation of the posterior covariance for the optimal estimates found by 4DVAR is available [6]. Consider the expression for the optimal estimate that was used in the representer method (12): $\hat{q}(\mathbf{x},t) = q_F(\mathbf{x},t) + \sum_{m=1}^{M} \beta_m r_m(\mathbf{x},t)$ and the resulting formulas. They allow us to write

$$\mathrm{Cov}(\hat{q}(\mathbf{x},t)) = \mathrm{Cov}(q_F(\mathbf{x},t)) - \mathbf{r}(x,t)^T(\mathbf{R}+\mathbf{C}_{\epsilon_q})^{-1}\mathbf{r}(\mathbf{x}',t').$$

Thus if we have an estimate of the covariance of our first guess q_F we can use it along with values from the representer method to estimate a covariance for our optimal estimate. A similar calculation will give us covariance estimates for the estimated errors f, i and $\boldsymbol{\epsilon}_q$.

Task 15. *Derive posterior covariance estimates for the errors assumed in the wind fields E_u, emissions E_u, diffusivity and fall out constants Γ and γ, and data $\mathbf{d}_Q$, $\mathbf{d}_q$, $\mathbf{d}_u$.*

We will follow the approach in [45] where the covariance is estimated from an ensemble of perturbations are applied to the optimal adjoint solution. Since we will use the representer method, perturbing the adjoint solution is achieved by perturbing the representer coefficients. This involves only the cost of a single 4DVAR assimilation, and the ensemble can be run in parallel.

Task 16. *Estimate errors in data from NWP, NOAA, US Forest Service and EPA ($\mathbf{d}_Q$, $\mathbf{d}_q$, $\mathbf{d}_u$) along with their covariances.*

6 Project Management

Technical Outcomes. The PIs and students will contribute to the development of weak constraint 4DVAR using a newly developed Smoke3d solver with operational weather and smoke data using appropriate error covariances. Drs. Mead and Calhoun will co-supervise two PhD students in the Computing PhD. One student in the Computational Mathematics Science and Engineering emphasis will focus on the Smoke3d solver and the parallel AMR implementation of 4DVAR. The second student in the Data Science emphasis will focus obtaining emission data from operational smoke forecasting systems, deriving new adjoint equations and methods for covariance estimation. While the focus of their dissertations will be distinct, their work will require a collaborative effort. Together they will oversee an undergraduate student who will post-processing the data into a form that can be used by the model equations. The team of PIs and students will meet bi-weekly in the early stages of the project and then weekly when the students are able to work more independently.

Project Timeline. The PIs propose a three-year project in order to complete the research objectives, with both PhD students and the undergraduate student commencing in Fall 2021. The timeline below identifies which tasks the student in Computational Mathematics Science and Engineering (CMSE) will lead and which tasks the student in Data Science will lead, and which tasks they will do collaboratively. Each numbered Task is described chronologically in this document.

Task	2021	2022	2022	2022	2023	2023	2023	2024	2024
	Fall	Spring	Summer	Fall	Spring	Summer	Fall	Spring	Summer
1	🟦								
2	🟨	🟨	🟨						
3		🟦							
4	🟨	🟨							
5			⬛	⬛					
6			🟨	🟨					
7		🟦	🟦	🟦					
8,9	🟦	🟦	🟦	🟦	🟦				
10					🟦	🟦	🟦	🟦	🟦
11						🟦	🟦	🟦	🟦
12				🟨	🟨				
13					🟨	🟨	🟨		
14						🟨	🟨		
15							🟨	🟨	
16								⬛	⬛
	⬛ All	🟦 CMSE	🟨 Data Science						

7 Results from Prior NSF Support

PI Mead: DMS-1720472 *Algorithms for Assessing and Improving Joint Inversion*; 9/2017-8/2021; $204,457. Intellectual Merit: We use joint inversion to combine observations of energy transfer to image the structure of Earth's subsurface. A single objective function with information from multiple types of data is optimized using least squares [18, 17, 16]. Project quantified the benefits of increased data collection and the effectiveness of combining different data types [42]. Broader Impacts: This approach aids in building safe structures, allows us to locate minerals, hydrocarbons, groundwater and contaminants; map tunnels, pipes and mines, and can be extended to other fields such as wireless communication and video processing. The work supported a Geophysics PhD dissertation (Domenzain, 2019) and Computing PhD dissertation (Ferdous, in progress).

Co-PI Calhoun: DMS-1819257 *Parallel, adaptive Cartesian grid algorithms for natural hazards modeling*; $315,586. 6/18-6/2021. Intellectual Merit: We are working to implement elliptic solvers for adaptively refined quadtree meshes. The funding is supporting **Computing PhD student** Damyn Chipman (Start date: 1/2020-present). **Undergraduate student** Scott Aiton (BSU 2020; Applied Mathematics and Computer Science). **Undergraduate Hannah Spero** (BSU 2021). Spero is using GeoClaw and the GeoClaw extension of ForestClaw to develop a model of the 1976 Teton Dam failure in Eastern Idaho.

Co-PI Calhoun: DMS-1419108 *A parallel algorithmic framework for flexible time discretization on adaptive Cartesian meshes.* 6/2014-8/2018; $194,000. Development of ForestClaw software and implementation of multirate time stepping for multistage time stepping methods. Work supported Masters student Talin Mirzakhanian (2017) and student intern Melody Shih (Columbia/NYU; 2017). Development of Ash3d was funded through this award and in collaboration with Hans Schwaiger (USGS).