

COMPUTATIONALLY EFFICIENT WILDLAND FIRE SPREAD MODELS

Alan Lattimer^a, Jeff Borggaard^a, Serkan Gugercin^a, Kray Luxbacher^b, and Brian Lattimer^c

^aDepartment of Mathematics, Virginia Tech

^bDepartment of Mining & Minerals Engineering, Virginia Tech

^cJensen Hughes, Inc.

ABSTRACT

The modeling and simulation of fires is a complex, multi-scale problem. The need to create accurate models extends from prevention to prediction to damage control management. Since these problems are so complex, even models of moderately-sized fires in relatively small domains require significant computational resources and time to solve. Further, the actual physics are often difficult to model exactly, e.g. the exact fuel loading and distribution in a wildland fire. To this end, much of the focus in the fire community has been on simplifying the physics of the problem to obtain reasonable models. Current model-reduction techniques for wildland fire-spread models only realize modest computational improvements due to difficulties reducing the nonlinear portion of the model. By addressing this issue, the resulting reduced-order model is more efficient to solve and maintains the accuracy associated with current model-reduction methods.

INTRODUCTION

Large-scale models for real-time simulations, as required for predicting wildland fires, are thus avoided due to limited computational resources. On the other hand, lower spatial resolution limits the physics that can be captured by the models. In this paper, a reduced-order model (ROM) technique that retains the underlying physics was employed to predict the solution to the advection-reaction-diffusion equation, a common component in fire models. This technique also projects the nonlinearity, resulting in ROMs that *decrease the online computational cost by 2-3 orders of magnitude*. This ROM technique can be applied in other large-scale fire applications, such as ventilation for mine fires. The accuracy and improved computational efficiency are demonstrated by building a ROM for a wildland fire spread model.

Proper orthogonal decomposition (POD) is the most commonly used technique to produce reduced order models for complex nonlinear dynamics (see Kunisch and Volkwein¹, Hinze and Volkwein², Volkwein³). This technique has been applied to wildland fire models (e.g. Mandel et al.⁴) in Sharma et al.⁵, Guelpa et al.⁶, where their ROMs exhibited an order of magnitude speedup in computational time while maintaining a high degree of accuracy. Though ROMs using only POD are effective, they can be significantly improved by addressing the so-called *lifting bottleneck*. The lifting bottleneck occurs when computing the reduced nonlinear term, since standard POD first lifts the reduced variables up to the full-order dimension, evaluates the nonlinear term, then projects the result back down to reduced dimension. Since nonlinear terms are computed at full order dimension, the computational gains compared to the original model are limited. To improve the performance, the discrete empirical interpolation method (DEIM), described in Chaturantabut and Sorensen^{7,8}, is used to perform an additional projection on the nonlinear function so that the nonlinearity is calculated on this new projected space instead of the full order space. We apply the work in Drmac and Gugercin⁹ to more efficiently determine the DEIM selection operator.

We emphasize that one must consider the goals for the reduced-order model to appropriately choose the model-reduction method for the situation. For other POD resources, we refer the reader to several good resources in addition to the ones above (see Bui-Thanh et al.¹⁰, Willcox et al.¹¹, Willcox and Peraire¹², Willcox¹³). While our focus in this paper is on POD, we recommend Benner et al.¹⁴, and the references therein, as excellent resources for other reduced-order modeling strategies.

WILDLAND FIRE-SPREAD MODEL

In this paper, we apply the POD with DEIM model reduction technique to simulate advection-reaction-diffusion equations that can be represented as

$$\frac{\partial \theta}{\partial t} = \kappa \nabla^2 \theta - \mathbf{v} \cdot \nabla \theta + \mathcal{S}(\theta) \quad [1]$$

where θ could be a concentration of chemical species or a temperature that transported throughout the domain. Typically, $\mathcal{S}(\theta)$ is a nonlinear function representing source and/or sink terms such as species production, species consumption, temperature increases due to reaction kinetics, etc.

The basic form of Equation [1] is present in the phenomenological model given by Equations [2]-[3] suggested in⁴ to predict flame front propagation in wildland fires. The one-dimensional version of this coupled system was studied to assess the benefits of POD with DEIM over the standard POD approach.

$$\frac{\partial T(x, t)}{\partial t} = \kappa \frac{\partial^2 T(x, t)}{\partial x^2} - v \frac{\partial T(x, t)}{\partial x} + \alpha \left(S(x, t) e^{-\beta/(T(x, t) - T_0)} - \gamma(T(x, t) - T_0) \right), \quad [2]$$

$$\frac{\partial S(x, t)}{\partial t} = -\gamma_S S(x, t) e^{-\beta/(T(x, t) - T_0)}. \quad [3]$$

These equations model the temperature ($T(x, t)$) and mass fraction of fuel ($S(x, t)$) in a propagating fire. Using constant parameters and wind velocity, the nonlinearity of this model occurs via a reaction term. We used the parameter values specified in Mandel et al.⁴ (i.e. $\kappa = 2.1360 \times 10^{-1} m^2 s^{-1} K^{-3}$, $\alpha = 1.8793 \times 10^2 K s^{-1}$, $\beta = 5.5849 \times 10^2 K$, $\gamma = 4.8372 \times 10^{-5} K^{-1}$, and $\gamma_S = 1.6250 \times 10^{-1} s^{-1}$). Further, for this paper we assume $v = 0 ms^{-s}$ and $T_0 = 293K$. To simplify computation, we consider the related model where $\tilde{T} = T - T_0$. Substituting into [2]-[3], we obtain the following equivalent wildland fire spread model

$$\frac{\partial \tilde{T}(x, t)}{\partial t} = \kappa \frac{\partial^2 \tilde{T}(x, t)}{\partial x^2} - v \frac{\partial \tilde{T}(x, t)}{\partial x} + \alpha \left(S(x, t) e^{-\beta/\tilde{T}(x, t)} - \gamma(\tilde{T}(x, t)) \right), \quad [4]$$

$$\frac{\partial S(x, t)}{\partial t} = -\gamma_S S(x, t) e^{-\beta/\tilde{T}(x, t)}, \quad [5]$$

where the actual temperature can be recovered from the solution to [4] by $T(x, t) = \tilde{T}(x, t) + T_0$. Assuming a constant velocity v , the system was discretized in the spatial domain from $0 - 1000m$ in $n = 6000$ equal increments to obtain the $2n$ -dimensional dynamical system

$$\begin{bmatrix} \dot{\tilde{\mathbf{T}}}(t) \\ \dot{\mathbf{S}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{T}}(t) \\ \mathbf{S}(t) \end{bmatrix} + \begin{bmatrix} \alpha \mathbf{f}(\tilde{\mathbf{T}}(t), \mathbf{S}(t)) \\ -\gamma_S \mathbf{f}(\tilde{\mathbf{T}}(t), \mathbf{S}(t)) \end{bmatrix}, \quad [6]$$

with $\mathbf{f}(\tilde{\mathbf{T}}(t), \mathbf{S}(t)) = \mathbf{S}(t) e^{-\beta/\tilde{\mathbf{T}}(t)}$. We can write this system more compactly as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{F}(\mathbf{x}(t)), \quad [7]$$

where

$$\mathbf{x}(t) = \begin{bmatrix} \tilde{\mathbf{T}}(t) \\ \mathbf{S}(t) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \text{ and} \quad \mathbf{F}(\mathbf{x}(t)) = \begin{bmatrix} \alpha \mathbf{S}(t) e^{-\beta/\tilde{\mathbf{T}}(t)} \\ -\gamma_S \mathbf{S}(t) e^{-\beta/\tilde{\mathbf{T}}(t)} \end{bmatrix},$$

with $\mathbf{x}(t) \in \mathbb{R}^{2n}$, $\mathbf{A} \in \mathbb{R}^{2n \times 2n}$, $\tilde{\mathbf{A}} \in \mathbb{R}^n$, and $\mathbf{F} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$. For this paper, we reduce the wildland fire-spread model by projecting this dynamical system to a lower dimensional dynamical system of the same form. This lower dimensional dynamical system is solved using standard differential-equation solvers and lifting the solution back to the original system dimension.

REDUCED-ORDER MODELING

To create a reduced-order model of the dynamical system given by Equation [7], we project the full-order dynamical system to a lower-dimensional dynamical system. In particular, suppose that $\mathbf{V} \in \mathbb{R}^{2n \times r}$ with $\mathbf{V}^T \mathbf{V} = \mathbf{I}_r$ is a projection matrix. We approximate the full state by $\mathbf{x} \approx \mathbf{V} \mathbf{x}_r$ enforce the Galerkin condition where

$$\mathbf{V}^T (\mathbf{V} \dot{\mathbf{x}}_r(t) - \mathbf{A} \mathbf{V} \mathbf{x}_r(t) - \mathbf{F}(\mathbf{V} \mathbf{x}_r(t))) = 0. \quad [8]$$

Defining $\mathbf{A}_r = \mathbf{V}^T \mathbf{A} \mathbf{V} \in \mathbb{R}^{r \times r}$, we obtain the following r -dimensional reduced-order model (ROM)

$$\dot{\mathbf{x}}_r(t) = \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{V}^T \mathbf{F}(\mathbf{V} \mathbf{x}_r(t)). \quad [9]$$

While there are many ways to construct the projection matrix, \mathbf{V} , we focus on the proper orthogonal decomposition (POD) technique in this paper. For this method, we seek to minimize the error between the full-order and reduced-order systems in the least-squares sense. POD provides an optimal solution to this minimization problem. We cover a very basic description here on how to create the POD projection matrix. For a more thorough coverage of the topic, we refer the reader to Berkooz et al.¹⁵, Kunisch and Volkwein^{1, 16}, Hinze and Volkwein², Borggaard et al.¹⁷ and the references therein.

Suppose we desire a reduced-order model size of r , then $k > r$ full-order data snapshots, ϕ_i for $i = 1, \dots, k$, are captured and assembled into a solution matrix $\Phi = [\phi_1 \dots \phi_k]$. We then perform a singular value decomposition (SVD) on the solution matrix as

$$\Phi = \mathbf{U} \Sigma \mathbf{W}^T \quad [10]$$

where $\mathbf{U}, \mathbf{W} \in \mathbb{R}^{n \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, see Golub and Van Loan¹⁸ for a full description of the SVD. The projection matrix \mathbf{V} is then formed by truncating \mathbf{U} to the first r columns, as $\mathbf{V} = [u_1 u_2 \dots u_r]$. It is important to note that the columns of \mathbf{U} are orthonormal; therefore, $\mathbf{V}^T \mathbf{V} = \mathbf{I}_r$.

For the system in [6], we must separately create the POD projection matrices for temperature $\tilde{\mathbf{T}}$ and mass fraction of the species \mathbf{S} . We create solution matrices Φ_T and Φ_S and compute the SVD of each of these to determine \mathbf{U}_T and \mathbf{U}_S . Setting the reduced-order sizes for temperature and mass fraction, r_T and r_S respectively, we create the projection matrices $\mathbf{V}_T \in \mathbb{R}^{n \times r_T}$ and $\mathbf{V}_S \in \mathbb{R}^{n \times r_S}$ where $r = r_T + r_S$. We can then build the full projection matrix $\mathbf{V} \in \mathbb{R}^{2n \times r}$ by the block matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_S \end{bmatrix}, \quad [11]$$

where we see that $\mathbf{V}^T \mathbf{V} = \mathbf{I}_r$. Using this matrix, we can project the full-order dynamical system [7] to obtain the ROM [9]. It is straight-forward to show that by expanding \mathbf{V} and Equation [7], we obtain the following coupled ROM for temperature and mass fraction

$$\underbrace{\begin{bmatrix} \dot{\tilde{\mathbf{T}}}_r(t) \\ \dot{\mathbf{S}}_r(t) \end{bmatrix}}_{\dot{\mathbf{x}}_r(t)} = \underbrace{\begin{bmatrix} \mathbf{A}_{T_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_r} \underbrace{\begin{bmatrix} \tilde{\mathbf{T}}_r(t) \\ \mathbf{S}_r(t) \end{bmatrix}}_{\mathbf{x}_r(t)} + \underbrace{\begin{bmatrix} \alpha \mathbf{V}_T^T \mathbf{f}(\mathbf{V}_T \tilde{\mathbf{T}}_r(t), \mathbf{V}_S \mathbf{S}_r(t)) \\ -\gamma_S \mathbf{V}_S^T \mathbf{f}(\mathbf{V}_T \tilde{\mathbf{T}}_r(t), \mathbf{V}_S \mathbf{S}_r(t)) \end{bmatrix}}_{\mathbf{V}^T \mathbf{F}(\mathbf{V} \mathbf{x}_r)}, \quad [12]$$

Unfortunately, to evaluate the nonlinear function \mathbf{F} in Equation [12], we must first project $\mathbf{x}_r(t)$ back to the full order. After we have applied \mathbf{F} , we must then project the result back to the lower order by

pre-multiplying it by \mathbf{V}^T . This process is known as the *lifting bottleneck*. Therefore, for POD, the computational gains compared to the original model are limited since the nonlinear terms are computed at the full-order dimension. It is the primary reason that the performance improvements are limited when applying POD to a nonlinear system. In the next section, we discuss one method for overcoming this bottleneck.

Discrete Empirical Interpolation Method (DEIM)

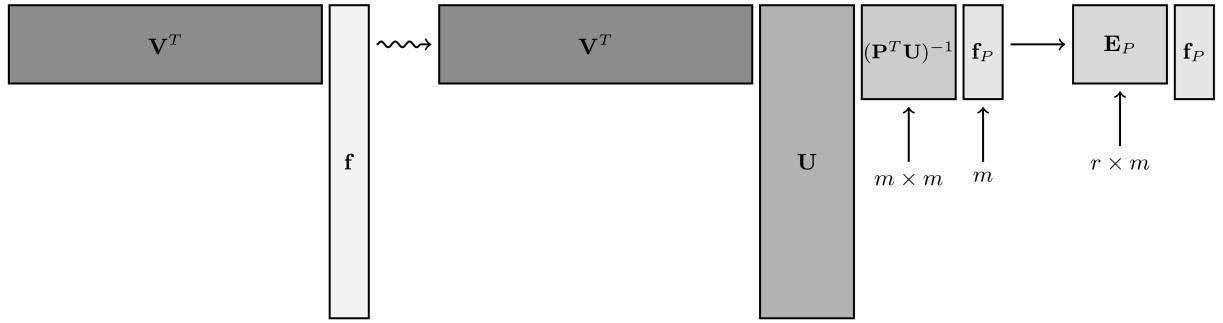
For this discussion, suppose we have a nonlinear system given by $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t))$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^n$, and $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$. As shown in the last section, the Galerkin projection yields the following ROM

$$\dot{\mathbf{x}}_r(t) = \underbrace{\mathbf{V}^T \mathbf{A} \mathbf{V}}_{\mathbf{A}_r: r \times r} \mathbf{x}_r(t) + \underbrace{\mathbf{V}^T \mathbf{f}(\mathbf{V} \mathbf{x}_r(t))}_{r \times n \quad n \times 1}, \quad [13]$$

where $\mathbf{x}_r : \mathbb{R} \rightarrow \mathbb{R}^r$.

For the nonlinear term, $\mathbf{x}_r(t)$ must be lifted back to the original size of the state space, i.e. $\mathbf{V} \mathbf{x}_r(t) \in \mathbb{R}^n$, before evaluating \mathbf{f} . This implies that the computational complexity of calculating the nonlinear term is order n . Using a discrete version of the Empirical Interpolation Method introduced in Barrault

Figure 1: Visual depiction of the DEIM approximation $\mathbf{V}^T \mathbf{f}(t) \approx \mathbf{E}_P \mathbf{f}_P(t)$.¹⁹



et al.²⁰ known as DEIM⁸, we can address the lifting bottleneck associated with reducing nonlinear equations. Given the nonlinear dynamics given in Equation [13] with the nonlinearity $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, when we capture the data snapshots of the state vector, we also compute snapshots associated with the nonlinearity. We then calculate the $n \times m$ DEIM-projection matrix \mathbf{U} by selecting the first m left-singular vectors associated with the nonlinear snapshots. Using this DEIM-projection matrix, we approximate the nonlinear function \mathbf{f} by

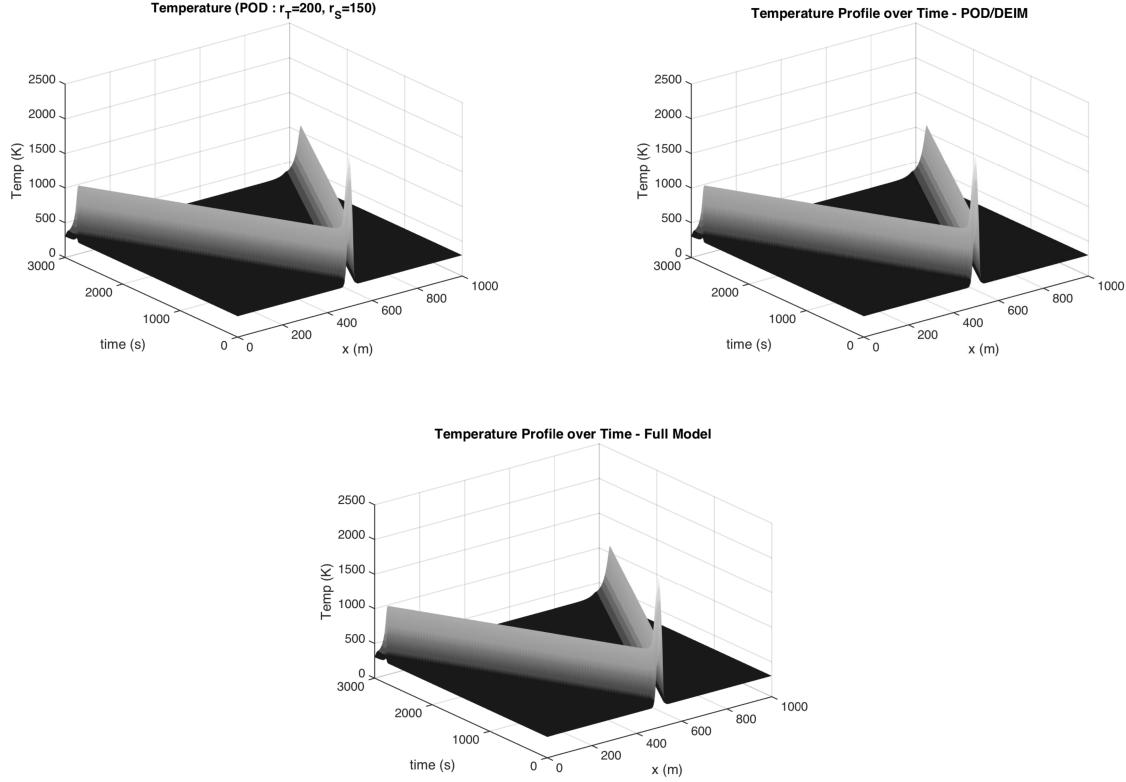
$$\hat{\mathbf{f}}(t) = \mathbf{U}(\mathbf{P}^T \mathbf{U})^{-1} \mathbf{P}^T \mathbf{f}(t). \quad [14]$$

Here $\mathbf{P} \in \mathbb{R}^{n \times m}$ is the DEIM-selection operator that is produced by selecting columns of the n -dimensional identity matrix \mathbf{I} . This selection operator \mathbf{P} forces interpolation at the DEIM indices of \mathbf{f} . In Chaturantabut and Sorensen⁸, the indices are determined by a greedy search process. However, for this paper, we implemented a new variant of DEIM recently developed by Drmac and Gugercin⁹ where the greedy search is performed via a pivoted QR decomposition. Now the reduced nonlinear term becomes

$$\mathbf{f}_r(\mathbf{x}_r(t)) \approx \mathbf{V}^T \mathbf{U}(\mathbf{P}^T \mathbf{U})^{-1} \mathbf{P}^T \mathbf{f}(\mathbf{V} \mathbf{x}_r(t)). \quad [15]$$

We emphasize that we only evaluate the selected rows of $\mathbf{V} \mathbf{x}_r(t)$. Further, when we have a component-wise nonlinearity, as we do here, we can move \mathbf{P} into the nonlinear function. Then we define $\mathbf{E}_P := \mathbf{V}^T \mathbf{U}(\mathbf{P}^T \mathbf{U})^{-1}$ and $\mathbf{F}_P(t) := \mathbf{f}((\mathbf{P}^T \mathbf{V}) \mathbf{x}_r(t))$, where $\mathbf{E}_P \in \mathbb{R}^{r \times m}$ and $(\mathbf{P}^T \mathbf{V}) \in \mathbb{R}^{m \times r}$ only have to be computed once. For more details on DEIM, we refer the reader to Chaturantabut and Sorensen⁸ and references therein.

Figure 2: FOM versus POD and POD/DEIM ROM where $r_T = 250$, $r_S = 150$, and $r_{DEIM} = 250$.



NUMERICAL RESULTS

Starting with the wildland fire-spread model given by Equations [2]-[3], we set the initial conditions as follows

$$T(x, 0) = T_c \exp \left[-\frac{(x - x_c)^2}{\sigma^2} \right], \quad [16]$$

$$S(x, 0) = 1, \quad [17]$$

which sets an initial fire at the 500 meter point of the domain and sets the entire domain to have a maximum fuel loading. The fire then propagates across the domain towards both boundaries based on Equations (2-3) as seen in Figure 2.

The system is then discretized using finite differences to obtain coupled ODE model given by Equation [6]. This system is solved from $t = [0, 3000]$ to create a full-order solution. Using the full-order solutions, separate POD bases for temperature and fuel mass fraction were built and used to project the full-order system to the reduced-order model given by Equation [12]. We created several ROMs using various sizes to reduce the temperature r_T and mass fraction r_S . It is important to note that the mass fraction of fuel was easier to reduce and thus required fewer POD to create an accurate model. Therefore, to reduce the computation time of the ROM, fewer mass fraction POD vectors were used. We focus on three representative size reduced-order models in Table 1. Each of the ROMs was solved using the same ODE solver that was used for the FOM.

To measure the effectiveness of the POD model, the relative error between the temperature of the full-order and reduced-order solutions was calculated. Specifically, let \mathbf{T}_k be the full-order solutions of the

Figure 3: Fire spread for FOM and POD/DEIM.

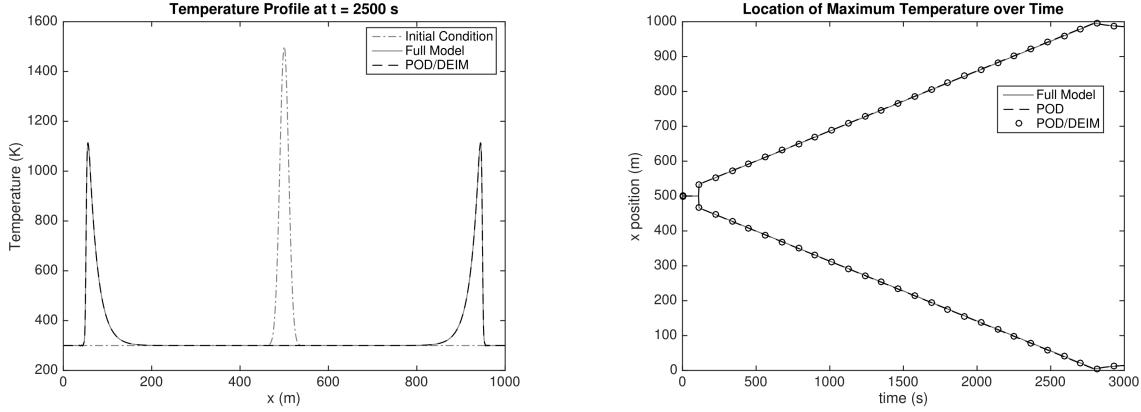


Table 1: Results for the ROM. Solution time for the FOM was 99.1 s

# POD Vectors $r_T/r_S/r_{DEIM}$	POD			POD/DEIM		
	Time (s)	Speed up	Rel Error	Time (s)	Speed up	Rel Error
70/35/250	1.33	74.6	1.7471e-02	0.13	760.1	1.7594e-02
200/150/250	15.4	6.38	4.3329e-03	0.72	137.6	5.2877e-04
200/200/250	18.5	5.31	1.8478e-02	0.84	118.2	1.3873e-02

temperature at $t = 0 = t_0, t_1, \dots, t_f = 3000$. Similarly, we define the reduced-order temperature solutions at the same time steps by $\widehat{\mathbf{T}}_k$. We can assemble these into the matrices

$$\mathbf{T} = [\mathbf{T}_0 \cdots \mathbf{T}_f] \quad \widehat{\mathbf{T}} = [\widehat{\mathbf{T}}_0 \cdots \widehat{\mathbf{T}}_f]. \quad [18]$$

With these matrices defined, we can now use the matrix 2-norm to compute the relative error between the FOM and the ROM given as

$$T_{rel\ err} = \frac{\|\mathbf{T} - \widehat{\mathbf{T}}\|_2}{\|\mathbf{T}\|_2}. \quad [19]$$

From Table 1, we see that there was some reduction of computation time when using POD. While the improvement in computation time is very good for the smallest ROM ($r_T = 70$, $r_S = 35$), the error in the system is too high. As is often the case with numerical computations, there is a trade off between accuracy and speed. While larger models are more accurate, they are only 5-7 times faster than the FOM.

Next we applied DEIM to each of the POD models by projecting the nonlinearity using 250 DEIM vectors. Because we have a component-wise nonlinearity, we can move the selection operator into the nonlinear function. Additionally, since the nonlinearity is the same (except for a constant) for both the temperature and fuel mass fraction, we only need to compute the DEIM projection for one of them. This results in the POD/DEIM ROM given by

$$\begin{bmatrix} \dot{\widehat{\mathbf{T}}}_r(t) \\ \dot{\mathbf{S}}_r(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{T_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{T}}_r(t) \\ \mathbf{S}_r(t) \end{bmatrix} + \begin{bmatrix} \alpha \mathbf{E}_T \mathbf{f}(\mathbf{P}_T \widetilde{\mathbf{T}}_r(t), \mathbf{P}_S \mathbf{S}_r(t)) \\ -\gamma_S \mathbf{E}_S \mathbf{f}(\mathbf{P}_T \widetilde{\mathbf{T}}_r(t), \mathbf{P}_S \mathbf{S}_r(t)) \end{bmatrix}, \quad [20]$$

where $\mathbf{E}_T = \mathbf{V}_T^T \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1} \in \mathbb{R}^{r_T \times m}$, $\mathbf{E}_S = \mathbf{V}_S^T \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1} \in \mathbb{R}^{r_S \times m}$, $\mathbf{P}_T = \mathbf{P}^T \mathbf{V}_T \in \mathbb{R}^{m \times r_T}$, and $\mathbf{P}_S = \mathbf{P}^T \mathbf{V}_S \in \mathbb{R}^{m \times r_S}$ call all be pre-computed. This means that the nonlinear computational complexity is now order $2m$. Because of this, all of the DEIM models were faster than the fastest POD model as shown in Table 1. Further, the relative error for the DEIM ROMs were essentially the

same as the POD model of the same size. The most accurate DEIM model, shown in Figure 2, was almost 140 times faster to solve than the FOM. Figure 4 shows that the POD/DEIM ROM provides an excellent approximation of the full-order model (FOM) matching both the flame front location and temperature profile quite well. When using POD with DEIM, the solution times were significantly better than POD alone while maintaining essentially the same error. The results demonstrate that using POD with DEIM can reduce the computational time by 2-3 orders of magnitude while retaining the model accuracy. Importantly, this is accomplished without removing the physics associated with the problem.

Varying the Input Fuel Loading

While these results are good, they represent how well we can approximate a known solution with a reduced-order model. Ultimately for this technique to be useful, we would like to create a reduced-order model from one input condition and then use it as a surrogate model for various other input conditions. In particular, we built a reduced-order model for the condition where the fuel mass fraction was one over the entire domain. We then ran the reduced-order model with a modified initial fuel loading. The result was then compared to full-order model run using the modified fuel loading. To be an effective surrogate model, the reduced-order model should do a good job of approximating the full-order model while significantly reducing the computation time over a wide range of input conditions. For our tests, we compared four initial loadings as shown in Figure 4. The first two initial conditions are symmetric over the domain while the last two are not.

Figure 4: Fuel mass fraction (S) initial conditions.

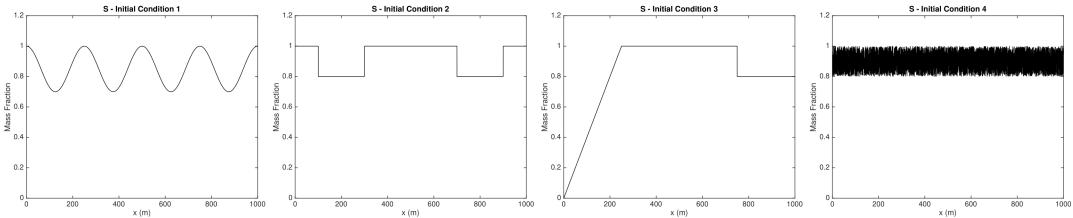


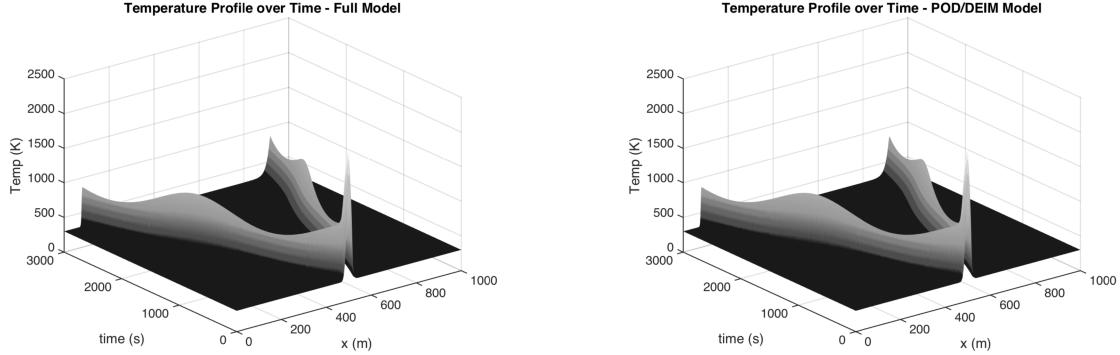
Table 2: Results for the ROM given different initial conditions and $r_T = 250$, $r_S = 150$, and $r_{DEIM} = 250$.

Initial Condition	Full-order		POD			POD/DEIM		
	Time (s)	Time (s)	Speed up	Rel Error	Time (s)	Speed up	Rel Error	
1	99.7	15.0	6.6	1.0358e-03	0.69	144.9	6.8308e-04	
2	99.6	12.4	8.1	2.4817e-03	0.55	181.3	6.5510e-04	
3	99.2	15.1	6.6	5.6108e-02	0.73	144.9	5.6250e-02	
4	98.8	15.3	6.3	4.2332e-03	0.68	145.6	5.5969e-03	

To see how reduced-order models can be utilized to provide cost savings, we look at the total costs of building and running these models. POD reduced-order models require that a full-order simulation first be computed. This could be a computer simulation or an actual experiment. For our case, the full-order model used to create the ROM took approximately 99 seconds to compute. It then took about one second to build the POD and DEIM bases and assemble the ROM. Once assembled, the ROM takes less than a second to compute. Therefore, it takes about two additional seconds to compute a ROM for an existing full-order model (FOM). However, if we use the same ROM for a different initial fuel loading, it only takes one second, whereas a new FOM would require 99 seconds. The computational savings only become more pronounced as more ROMs are run. For example, suppose there are ten initial fuel loadings

that must be tested. For the FOM, it would require 990 seconds to complete the testing. However, it only takes 110 seconds to test the same number of input conditions for a ROM (101 seconds for the first initial condition and one second for each subsequent condition). While the computation time-savings are evident, one must ensure that the ROM will provide an accurate surrogate model over the range of initial conditions being tested.

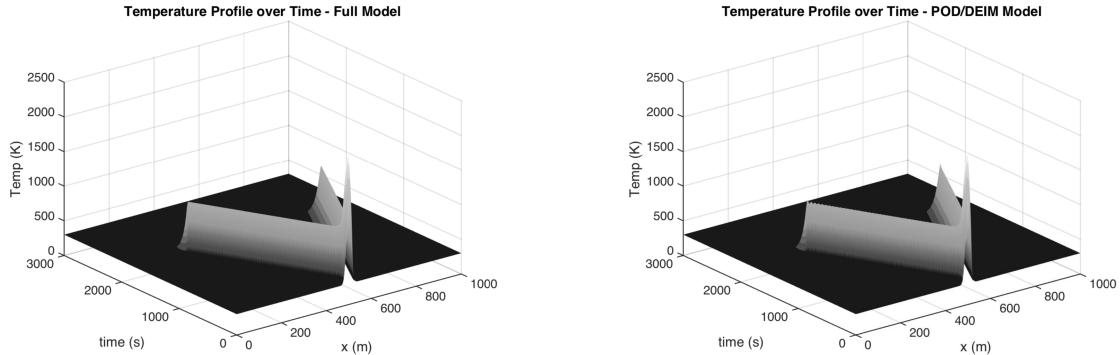
Figure 5: FOM versus POD/DEIM ROM for initial condition 1 where $r_T = 250$, $r_S = 150$, and $r_{DEIM} = 250$.



For these tests the reduced-order model size was set to the best match shown in Table 1 where the size was $r_T = 200$, $r_S = 150$, and $r_{DEIM} = 250$. Using the new initial conditions for fuel loading, both the FOM and ROM were simulated over 3000 seconds. In practice, one would only simulate the system using the ROM. The FOM is simulated here only to provide a comparison.

For the first fuel loading, we chose a smooth oscillating distribution of the fuel over the domain. As expected the flame height and spread oscillated in the same frequency as the fuel loading. Figure 5 shows that the ROM provides a good approximation to the FOM matching both the maximum temperature and spread rate of the FOM. In Table 2 shows that ROM is over two orders of magnitude faster than the FOM. Further, the error between the FOM and ROM is on the order of 10^{-4} which is very good.

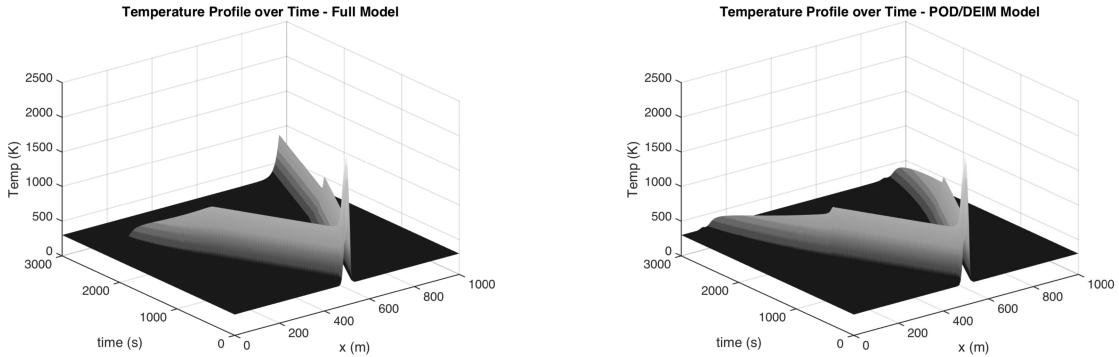
Figure 6: FOM versus POD/DEIM ROM for initial condition 2 where $r_T = 250$, $r_S = 150$, and $r_{DEIM} = 250$.



The second fuel loading simulates building a 10 meter fire break 300 meters away from the original fire. As can be seen in Figure 6, the fire, for both the ROM and the FOM, is extinguished at the fire break. There are some minor oscillations of the maximum temperature in the ROM near the transition point. These are typical of POD solutions and can often be addressed by increasing the number of POD

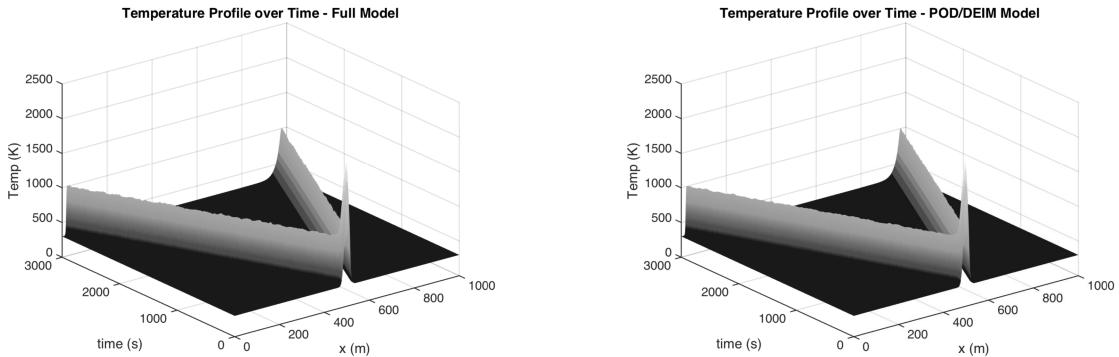
basis vectors. We emphasize that, again, the ROM provides a good approximation of the FOM for a substantially lower simulation time, as shown in Table 2. We can see that the ROM provides a good surrogate model for the FOM when the fuel distribution is symmetrical over the domain.

Figure 7: FOM versus POD/DEIM ROM for initial condition 3 where $r_T = 250$, $r_S = 150$, and $r_{DEIM} = 250$.



Next, we look at an asymmetric fuel loading over the domain. For the FOM, we see that the fire behaves differently on each side of the fire. On the side where the fuel loading linearly decreases the fire eventually goes out. On the other side, the temperature and flame-spread rate decrease when there is a discontinuous drop in fuel loading. However, for the ROM, the fire behaves the same on both sides of the initial fire. In particular, Figure 7 shows that it approximately averages both behaviors, capturing neither side well. The ROM errors shown in Table 2 also show that the error for both the POD and POD/DEIM models are much higher than for either symmetric case. Even though the solution times are much quicker than the FOM, the ROM does not accurately capture the behavior of the FOM. We see here that, for this model, caution must be exercised when trying to simulate asymmetric input conditions when the POD basis is built from a symmetric input condition. There are strategies that include building the ROM with a more closely distributed fuel loading to the asymmetric cases being investigated, but defining those classes was beyond the scope of this paper.

Figure 8: FOM versus POD/DEIM ROM for initial condition 4 where $r_T = 250$, $r_S = 150$, and $r_{DEIM} = 250$.



Finally, we looked at a Gaussian distribution for the fuel mass fraction between 1.0 and 0.8. Here we are trying to simulate that fact that in real conditions the fuel concentration is not strictly uniform and will vary slightly across the domain, even from meter to meter. Strictly, this fuel loading is not symmetric,

however the mean of the fuel distribution is 0.9 across the domain. From Figure 8 we can see that while there are minor local variations in the temperature, the large-scale behavior of the fire is the same as the first symmetric fuel loading in Figure 5. With this behavior, we would expect the ROM to be a good surrogate model, and the ROM shown in Figure 8 indicates that it does. Table 2 shows that the ROM does provide a good approximation of the FOM at a fraction of the computational cost.

CONCLUSIONS

Nonlinear wildland fire-spread models such as the one given by Mandel et al.⁴ can be challenging to reduce to the lifting bottleneck associated with the nonlinear term. By addressing this bottleneck using DEIM, we were able to reduce the computational time by another order of magnitude over POD alone without sacrificing accuracy. Additionally, we are able to create a ROM from a single full-order simulation, and then use that same ROM to simulate several other input conditions. While our surrogate model is a good approximation when looking symmetric input fuel loadings, it does not perform well on asymmetric loadings. As was shown, the symmetric nature of the first model is embedded in the POD basis and thus asymmetric fuel loadings are not well-represented by that POD basis. One must exercise caution when utilizing a ROM as a surrogate model for prediction or decision making when the input conditions vary too far from the original model. Further, underlying structures in the original model must be considered when determining the efficacy of the ROM for simulating alternate input conditions.

ACKNOWLEDGEMENTS

This research was partially developed under Grant No. 200-2014-59669, awarded by NIOSH. The findings and conclusions in this report are those of the authors and do not reflect the official policies of the Department of Health and Human Services; nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

References

- ¹ K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical analysis*, 40(2):492–515, 2002.
- ² M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In *Dimension Reduction of Large-Scale Systems*, pages 261–306. Springer, 2005.
- ³ S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 4:4, 2013.
- ⁴ J. Mandel, L. S. Bennethum, J. D. Beezley, J. L. Coen, C. C. Douglas, M. Kim, and A. Vodacek. A wildland fire model with data assimilation. *Mathematics and Computers in Simulation*, 79(3):584–606, 2008.
- ⁵ B. R. Sharma, M. Kumar, and K. Cohen. Spatio-Temporal Estimation of Wildfire Growth. In *ASME 2013 Dynamic Systems and Control Conference*, pages V002T25A005–V002T25A005. American Society of Mechanical Engineers, 2013.
- ⁶ E. Guelpa, A. Sciacovelli, V. Verda, and D. Ascoli. Model reduction approach for wildfire multi-scenario analysis. In *Advances in Forest Fire Research*. Imprensa da Universidade de Coimbra, 2014.

- ⁷ S. Chaturantabut and D. C. Sorensen. Discrete empirical interpolation for nonlinear model reduction. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 4316–4321. IEEE, 2009.
- ⁸ S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- ⁹ Z. Drmac and S. Gugercin. A new selection operator for the Discrete Empirical Interpolation Method—improved a priori error bound and extensions. *arXiv preprint arXiv:1505.00370*, 2015.
- ¹⁰ T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2):880–896, 2007.
- ¹¹ K. Willcox, O. Ghattas, B. van Bloemen Waanders, and B. Bader. An optimization frame work for goal-oriented, model-based reduction of large-scale systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 2265–2271. IEEE, 2005.
- ¹² K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA journal*, 40(11):2323–2330, 2002.
- ¹³ K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.
- ¹⁴ P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.
- ¹⁵ G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.
- ¹⁶ K. Kunisch and S. Volkwein. *Crank-Nicolson Galerkin proper orthogonal decomposition approximations for a general equation in fluid dynamics*. Universität Graz/Technische Universität Graz. SFB F003-Optimierung und Kontrolle, 2002.
- ¹⁷ J. Borggaard, A. Duggleby, A. Hay, T. Iliescu, and Z. Wang. Reduced-order modeling of turbulent flows. In *Proceedings of MTNS*, volume 2008, 2008.
- ¹⁸ G. H. Golub and C. F. Van Loan. *Matrix computations. third*. The John Hopkins University Press, 1996.
- ¹⁹ A. M. Lattimer. *Model Reduction of Nonlinear Fire Dynamics Models*. PhD thesis, April 2016.
- ²⁰ M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.