1 Introduction

The goal is to create an operational wildland fire model that captures fireline behaviors that have been observed in other models and experiments at a much smaller computational cost.

Wildfires are multiscale systems with complex dynamics and feedback mechanisms, making it difficult to capture all of their characteristics. Physics based models, such as HIGRAD/FIRETEC [3] (H/F), model the fluid dynamics, the fire-atmosphere interaction, and the chemical processes (pyrolysis) of fires. These models can simulate a wide range of conditions but come with large computational costs and highly complex initialization. This means these models do not lend themselves to be used for exploring the development of a fire under slightly varying conditions without a considerable investment of time. Current operational fire models, used to forecast development of active fires, are mostly based off the Rothermel spread model, an empirical model originally developed in 1972. The Rothermel model has The aim of this research is to develop models that are

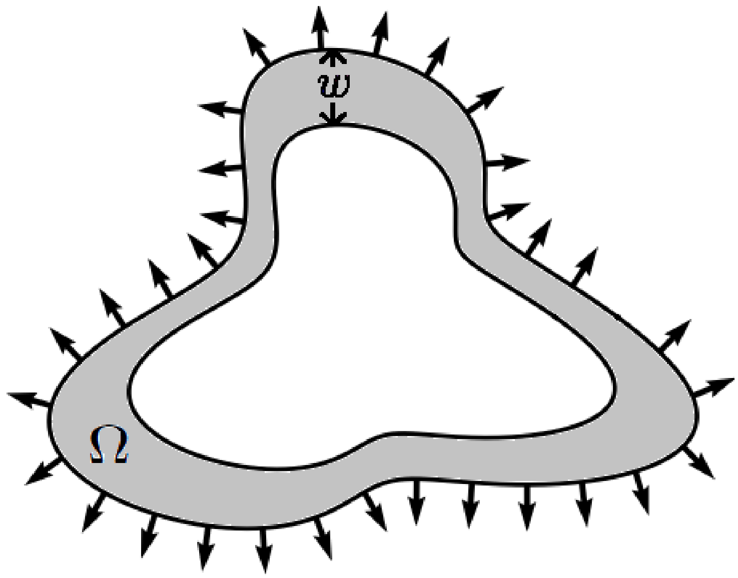
2 Interface Tracking Model

We cast the modeling of fire as an interface tracking problem where a fireline is represented as either a single closed curve (entire interior burning) ,Γ_0, or as an outer fireline with one or multiple inner firelines (regions of exhausted or unburnt fuel), Γ_i for $i \geq 1$. We assume that any burning section of fuel can be encapsulated with a somewhat smooth continuous curve. The firelines, Γ_i, represent the isotherm encapsulating the area where steady combustion of fuel is taking place, Ω.

We considered three drivers for determing spread rate. The first being the heat flux of combusting fuel. As fuel combusts it releases radiation that heats surrounding fuel to ignition temperature, which varies and ranges from (NEED CITATION AND VALUES). This radiation dries out surrounding fuel and allows for combustion to take place, allowing the fire to spread. The heat from wildfires raises the temperature of the surrounding air causing it to rise(NEED CITATION AND VALUES). These buoyant updrafts pull in surrounding air at the ground level that result in a vacuum effect, restraining the spread of the fireline. The radiation received by fuels is unimpeded by the wind so we considered these factors separately. Lastly, we account for the background wind speed that is typically always present in the event of a wildfire.

The fireline model represents the fire dynamics in terms of a normal velociy, $\frac{\partial \vec{x}}{\partial t} = V_n \hat{n}$, where $\vec{x}$ is a parameterization of a fireline in $\mathbb{R}^2$, and $\hat{n}$ is the outward unit normal of this interface. The normal velocityV_n is represented as the weighted sum from three of the dominating sources of fire spread:

$$V_n(\vec{x}) = \lambda_1 v_{bg}(\vec{x}) + \lambda_2 v_F(\vec{x}) + \lambda_3 v_{Sink}(\vec{x}).$$



The spread due to the heat flux of burning fuel is denoted by v_F, the convective sinks caused by updrafts are represented by v_{Sink}, and the ambient background wind near the ground is represented by v_{bg}.

2.1 Dynamics

2.1.1 Heat Flux

For the transfer of heat from a source we follow the inverse square law. To determine the total heat flux at a single point on the interface we integrate over the enclosed area of the interface.

$$v_F(\vec{x}) = \int_\Omega \frac{1}{\sqrt{\|\vec{x} - \vec{y}\|^2 + \epsilon^2}} dA$$

For this project we originally focused on fires with one interior fireline. With only one interior fireline the integral can be approximated by determining the width ,w_i, of the burning region between the inner and outer fireline at each discretization point. The width is defined as the distance between a point on the outer fireline and the closest point on the inner fireline. The closest point is determined by doing a few Newtons method iterations in Fourier space[1] to find the α_i where $\partial\left(\Gamma_0^i - \Gamma_1\right)/\partial\alpha = 0$ that is nearest to $i\frac{2\pi}{N}$ where N is the number of discretization points. With the corresponding alpha the FFT weights are used to return back to physical space by summing the corresponding exponentials. With the nearest interior points determined we also define the midpoints, y_i, as the halfway point between the outer point and its associated nearest interior point. These points will be used as the source locations for calculating the effect of the convective sinks.

INSERT GRAPHIC OF NEAREST NEIGHBOR RESULTS AND POINTS

The heat flux can now be calculated as a line integral along the outer fireline using the width to calculate the area of the enclosed burning region.

[1]Should i explain this further?

$$v_F(\vec{x}) = \oint_{\Gamma_0} \frac{w(\vec{y})}{\sqrt{\|\vec{x}-\vec{y}\|^2+\epsilon^2}} d\vec{s} \text{ for } \vec{y} \in \Gamma_0.$$

This integral is calculated by first determing ds by taking the spectral derivative of the outer interface. The guaranteed periodicity of the interface means we can calculate its derivatives in Fourier space making the calculation trivial and spectrally accurate[2]. Given we can express a function as the sum of complex exponentials using the discrete Fourier transform (DFT)

$$f(x) \simeq \sum_{-N/2}^{N/2} \hat{f}_k e^{ikx}$$

where $\hat{f}_k$ are the weights given by the DFT. The derivative can be easily approximated by scaling the weights by ik and taking the inverse Fourier Transform (IFT) since

$$g(x) = \frac{df}{dx} \simeq \sum_{-N/2}^{N/2} ik\hat{f}_k e^{ikx} \Longrightarrow \hat{g}_k = ik\hat{f}_k.$$

Using the spectral derivative we can easily approximate $d\vec{s} = \sqrt{\left(\frac{dx}{d\alpha}\right)^2 + \left(\frac{dy}{d\alpha}\right)^2}\hat{\tau}$, where $\Gamma_0 = (x(\alpha), y(\alpha))$, and sum to approximate the integral

$$v_F(\vec{x}_j) = \sum_{i=0}^{N-1} \frac{w(\alpha_i) * ds_i * (2\pi/N)}{\sqrt{\|\vec{x}_j - \vec{y}_i\|^2 + \epsilon^2}}.$$

2.1.2 Fire-Atmosphere Interactions

To model the effects of buoyant updrafts we considered a DONT KNOW WHAT TO CALL IT.

$$v_{Sink}(\vec{x}) = -\sum_{i=1}^{N} w(\vec{y}_i) \frac{\vec{x}-\vec{y}_i}{\|\vec{x}-\vec{y}_i\|^2} \cdot \hat{n}$$

The larger a burning region the greater the heat released from combustion, resulting in a stonger convective plume. For a fire with only a single interior fireline the width is used as a way to scale the convective sinks. The sum of all of these sinks with their strength, also scaled by distance, forms the convective sink contribution to the interface velocity. For firelines with multiple interior lines a Delauney triangulation was used to mesh the interior burning region. The centers were used as the sink locations and their strength scaled up by the corresponding triangles area. The sum over these triangles and the same r^{-1} distance scaling was used to form the convective sink velocity contribution in these cases.

2.2 Stability

[2]Explain this in more detail?

A numerical issue with modeling a fireline as a moving interface is the distortion of the mesh as the interface moves. This leads to a varying spatial resolution and can cause tangling and instabilities. By using methods for curvature driven flow[2] a tangential velocity component is introduced that guarantees points remain equispaced in arclength without affecting the shape of the interface (Figure). With this formulation, a well-behaved mesh is attained and a larger number of stable timesteps can be taken.

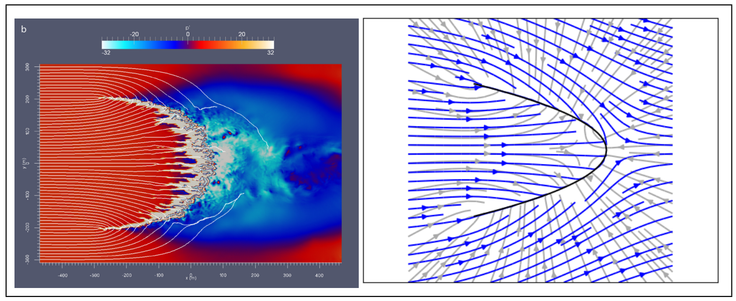


Figure 1: A comparison of streamlines from a HIGRAD/FIRETEC [3] high-fidelity simulation (left) and streamlines from the interface model (right). The streamlines of the convective sinks are plotted in gray and the streamlines of $\lambda_1\vec{v}_{bg} + \lambda_3\vec{v}_{sink}$ ($\lambda_1 = 26.6; \lambda_3 = 1.0$) are plotted in blue. This figure demonstrates that with proper scaling of the convective sink strength a similar velocity field to a high-fidelity model can be attained.

2.2.1 L-θ Formulation

With Γ given by $X = (x(\alpha,t), y(\alpha,t))$ a tangential velocity, v_s, is introduced to keep the points along the intereface equispaced in arclength; $X_t = v_n\hat{n} \rightarrow X_t = v_n\hat{n} + v_s\hat{s}$, where $\hat{s}$ is the unit tangent vector. To do this a switch of parameterization of Γ from x and y to the tangent angle, θ, and arclength s, more specifically $\frac{\partial s}{\partial \alpha}$. With curvature driven flow $v_n = \kappa = \theta_\alpha/s_\alpha$ we can express the evolution of the curve in terms of θ and $s_\alpha$via

$$s_{\alpha t} = (v_s)_\alpha - \frac{1}{s_\alpha}\theta_\alpha^2 \tag{1}$$

$$\theta_t = \frac{1}{s_\alpha}\left(\frac{1}{s_\alpha}\theta_\alpha\right)_\alpha + \frac{v_s}{s_\alpha}\theta_\alpha. \tag{2}$$

We can see that Eq. 2 is an advection-diffusion equation with stability condition $\Delta t < C \cdot (\bar{s_\alpha} h)^2$where $\bar{s_\alpha} = \min_\alpha s_\alpha$ and h is the grid spacing in α. The stiffness of the problem comes through Eq. (2) which can be dealt with by ensuring s_α doesn't depend on α. To achieve this enforce that $\frac{ds}{d\alpha}$ be everywhere equal to its mean

$$s_\alpha(\alpha,t) = \frac{1}{2\pi}\int_0^{2\pi} s_{\alpha'}\left(\alpha', t\right) d\alpha' = \frac{1}{2\pi}L(t)$$

where $L(t)$ is the total length of the interface Γ. If s_α initially satisfies this constraint then

$$v_s(\alpha,t) = v_s(0,t) + \frac{2\pi}{L}\int_0^\alpha \theta_{\alpha'}^2 d\alpha' - \frac{\alpha}{L}\int_0^{2\pi} \theta_{\alpha'}^2 d\alpha',$$

resulting from IDONTKNOW, maintains the constraint in time. The tangential velocity $v_s(0,t)$ is an arbitrary change of frame and can simply be set to zero. Now s_α does not depend on α and evolves with the length of the curve L. Eqn. 1 now becomes an ODE for L and we now have

$$L_t = -\frac{1}{L}\int_0^{2\pi} \theta_{\alpha'}^2 d\alpha' \tag{3}$$

$$\theta_t = \left(\frac{2\pi}{L}\right)^2 \theta_{\alpha\alpha} + \frac{2\pi}{L}v_s\theta_\alpha. \tag{4}$$

Substituting the curvature equation and it's derivative with respect to α into Eqns. 3 and 4 and using the restriction $\int_0^{2\pi} s_{\alpha'}\left(\alpha', t\right) d\alpha' = L(t)$ we can express our interface evolution in terms of the normal velocity CITE MOORE

$$L_t = -\int_0^{2\pi} v_n \theta_{\alpha'} d\alpha' \tag{5}$$

$$\theta_t = \frac{2\pi}{L}\left(v_n\right)_\alpha + \frac{2\pi}{L} v_s \theta_\alpha . \tag{6}$$

Now v_n can be formed using the methods described above section and with Eqns. 5 and 6 the discretization points can be kept equispaced in arclength. Since we are now keeping track of opening angle and length we need to track a reference point in cartesian space to account for translations of the interface. For this we track the surface mean $\langle X \rangle$ which moves according to

$$\langle X \rangle_t = \langle v_n \hat{n} + v_s \hat{s} \rangle$$

which is tracked using forward Euler[3].

3 Model Fitting

The proposed model has unknown parameters that DRAM will identify. The objective of DRAM is to start with a prescribed prior distribution and then sample the parameter space to determine the posterior probability density for each parameter. This is done by updating the likelihood based on the error between the model and data. The DRAM algorithm will show correlations between parameters, shortcomings of the model, and quantify uncertainty. DRAM requires analysis of results and reinitialization to attain good stationary posterior distributions. Depending on the number of inferred parameters, the DRAM algorithm can require a large amount of simulation runs to estimate these parameters, and my previous experiences with HPC will prove useful. In addition, the computational efficiency of this model makes it a natural fit for DRAM.

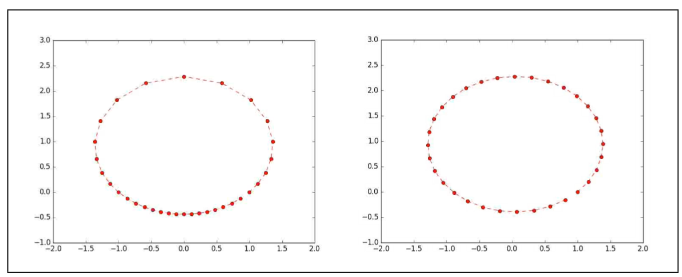


Figure 2: Snapshots of two moving interfaces initialized with the same shape and forcing. The left interface has no tangential velocity component and the right interface has the tangential velocity implemented [2]. Notice that the right interface has the same shape as the left and keeps points equispaced in arclength.

The error that was used for driving the model took three points from a head fire high fidelity simulation from HIGRAD/FIRETEC shown in [1] which had the canonical parabolic fireline shape I was hoping to capture. The initial fireline had a width (x domain) of 20m and a height (y domain) of 100m, which was similar to initial conditions in [1] with a backing wind of 5 m/s in the positive x-direction. The points used for the error were

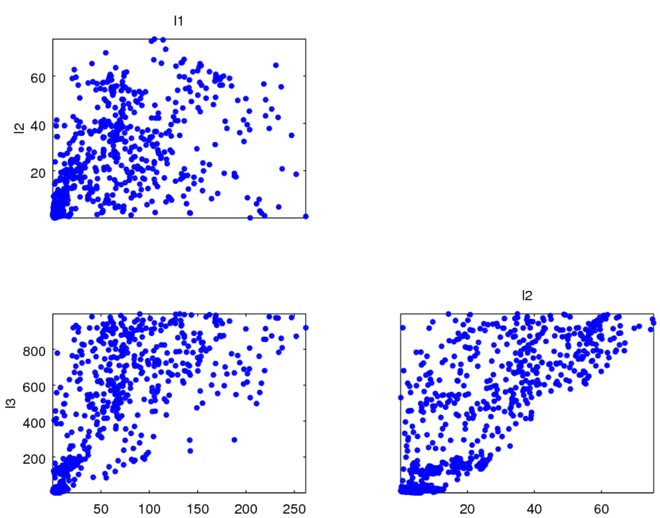
[3] Explain this in more detail?

x	y
-30	0
544	0
*	55

with the * being a free x value. Once a simulation was completed the min and max x values and max y values were taken and compared with the three points above. Since the initial fire line was initialized with its center being on the x-axis with a background wind only in the x-direction the developing fireline would be symmetric with respect to the x-axis as well as not drift upward. This means that these points were a consistent error measurement.

There have been many improvements but there seems to be, for lack of a better term, forbidden regions. There are combinations of λ_1, λ_2, and λ_3 that cause high frequency modes developing and the interface breaks. This is an obvious obstacle when using this model with DRAM since if a model returns nonsensical results how does DRAM update the proposal distribution? Another layer of difficulty is the fact that this code is written in Python. My solution for this was to put catch points in my code to detect when the interface had become unstable. When this happened the simulation was terminated and a number code was returned instead of a result. In DRAM whenever this code was received an error of ∞ was returned, which the code is able to handle. My initial fix for this was making modifications to a significant amount of the DRAM code to have it resample until it received an actual result. This caused the time to get samples to increase dramatically. I was able to do 200 runs over a 2 day period which is a completely inadequate amount of samples to be able to do any sort of analysis. By making the above change I was able to do 1,000 runs in a 1 day period. All of these runs returned dissappointing results, although, there is some knowledge gained from using DRAM with this model.

RESULTS



Luckily this is the one that produced some possible information about the model. From the bottom right plot we can see that there seems to be an area where DRAM cannot sample. Taking a points from this graph and fitting a line along the bottom region I arrived at an expression of

$$13.846\lambda_2 - \lambda_3 \geq 207.6$$

for the bottom region. Running my simulation with parameters satisfying this did in fact cause the simulation to break. Taking λ_1 and λ_2 factors from the upperleft plot and taking the satisfying the negative of the above expression resulted in simulations that ran the full time and didn't break. Sadly the simulation runs resulting from the sampling means are not realistic. The final shape of the resulting fireline does in fact have its width around the right scale of the data points but it's height is much too large, ~200m. At least a rough constraint for stability for the model was able to be realized. This exercise definitely has shown that much more foresight needs to be used in order to utilize an MCMC sampling algorithm efficiently.

References

[1] JM Canfield, RR Linn, JA Sauer, M Finney, and Jason Forthofer. A numerical investigation of the interplay between fireline length, geometry, and rate of spread. *Agricultural and forest meteorology*, 189:48–59, 2014.

[2] Thomas Y Hou, John S Lowengrub, and Michael J Shelley. Removing the stiffness from interfacial flows with surface tension. *Journal of Computational Physics*, 114(2):312–338, 1994.

[3] RR Linn and FH Harlow. FIRETEC: a transport description of wildfire behavior. Technical report, Los Alamos National Lab., NM (United States), 1997.

4 SARndbox

5 Conclusion