CA Model Report

Max Woodruff-Madeira

March 2025

1 Introduction

For this report, I will be detailing the progress of a cellular automata model I am developing based on rate of spread data calculated by Zack Holden which gives the predicted growth rate in ha/day of a fire at each point in the domain. The main difficulty with the model is working with rate of spread data which have units of acres/day. In order for this data to be usable, it must be converted to m/hr with some sense of directionality. I am attempting to solve this problem using area calculations and gradient descent.

2 Methods

The model runs on the growth maps, wind velocity, wind direction, slope, and aspect inputs. Before any rate of spread calculations take place, the wind and slope vectors (described by the wind velocity, wind direction, slope, and aspect) are combined to create one vector which describes the effect of wind and slope on fire spread. The calculations for the combined wind slope vector are shown in equations (1), (2), and (3).

$$wm = \frac{ws^{3.6}}{1 + \exp(0.6 * (10 - ws))} \tag{1}$$

$$sm = 0.2e^{0.13s} + 0.6 \tag{2}$$

$$r = (wm^2 + sm^2 + 2 * wm * sm * \cos(\theta_w - \theta_s))^{1/4} \tag{3}$$

$$\theta = \theta_s + \arctan\left(\frac{wm * \sin(\theta_w - \theta_s)}{sm + wm * \cos(\theta_w - \theta_s)}\right) \tag{4}$$

Then as each cell is ignited the rate of spread in each direction (RoS) for that cell is calculated using the combined wind slope vector and the predicted growth rate. The RoS is also updated for every burning cell at each timestep.

This calculation is done by first projecting the wind/slope vector onto each directional vector to get a starting guess for the RoS in each direction. Then an area is calculated from the initial guess by finding and summing the areas for each triangle created from two adjacent directional vectors and the vector spanning their two RoS vectors as shown in Fig. (1), giving us equation (5).

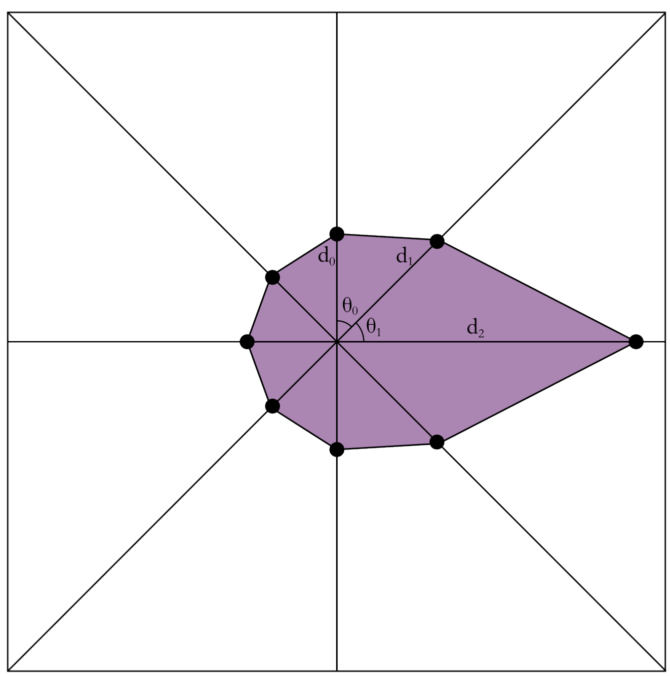

Figure 1: Cell RoS-area calculations

$$a_{total} = \frac{1}{2}(d_n d_0 \sin(|\theta_0 - \theta_n|) + \sum_{i=0}^{n-1} d_i d_{i+1} \sin(|\theta_{i+1} - \theta_i|)) \tag{5}$$

After calculating an initial RoS for each direction, I use gradient descent, or bisection, to ensure the area of our shape is equivalent to the area from the given RoS.

2.1 Gradient Descent

In order to get an area equivalent to the RoS I am looking to minimize the total area minus the given RoS as seen in equation (6). To ensure that there is a minimum at g(x)=0 I use the square of equation (6) giving me equation (7).

$$g(x) = a_{total} - RoS \tag{6}$$

$$f = g(x)^2 \tag{7}$$

To use gradient descent I need the gradient of equation (7) which is given by equation (8) where d is the magnitude of the RoS along the directional vector and theta is the angle of the directional vector.

$$\frac{\delta d_i}{\delta f} = (d_{i-1} \sin|\theta_i - \theta_{i-1}| + d_{i+1} \sin|\theta_{i+1} - \theta_i|) * g(x) \tag{8}$$

Using the gradient I can perform gradient descent to calculate a rate of spread in each direction for each cell that minimizes the difference between the shape of the fire spread and the estimated growth. In order to ensure that the gradient descent is nicely shaped I also added the constraints that the RoS can not have a negative value and that the shape created by the RoS must remain convex.

2.2 Bisection

The other method I implemented to minimize equation (5) is bisection. The bisection method ends up being faster than gradient descent and the RoS along each directional vector is not adjusted individually, so the spread maintains a more circular shape. In order to do bisection I first check if the area created by the vectors is bigger or smaller than the calculated growth and set that as the maximum or minimum respectively. Based on that, I find the counterpart by either subtracting an equal amount from the RoS vectors (with a minimum RoS of 0) or adding an equal amount to the RoS vectors to get a minimum or maximum. Once I have both a maximum and a minimum I can use bisection to find a shape with an area equal to the calculated growth.

3 Fire Propagation

Once I have a RoS for each cell, propagating the fire is fairly simple. The directional vectors in each cell come from an ignition point. All I need to do is calculate the length of each directional vector, using the ignition point and the direction they are pointing to, and then subtract the distance the fire has spread after each time step from that length as seen in equation (8) where phi is the length of the directional vector and s is the size of the time step.

$$\phi = \phi - RoS_{\phi} * s \tag{9}$$

Once phi is equal to zero, the cell in the corresponding direction is ignited.

4 Moving Forward

The biggest issue with the current version of the model is that the fire is spreading too fast. I found that multiplying the growth map by a weight of 0.1 led to good fire growth. Our best guess for what is causing the current problem is that the model is seeing the individual growth rates of each pixel as a kind of sum rather than taking more of a mean of those values. The other thing I would look into are those equations for finding the combined wind slope vector.