BSc Thesis Applied Mathematics

A physics-based cellular automaton model for wildland fire evolution

Robert-Jan Franciscus Gerardus Nijhuis

Supervisors: C.A. Pérez Arancibia
M. Gerhold

July, 2024

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

I want to thank my supervisors Carlos Pérez Arancibia and Marcus Gerhold for their continued support, encouragement and helpful comments. Furthermore I would like to thank Robor Electronics for letting me use their workplace. Finally, I would like to thank anyone not mentioned here, but who played their role to inspire me.

A physics-based cellular automaton model for wildland fire evolution

Robert-Jan. F.G. Nijhuis

July, 2024

Abstract

In this report, a physics-based mathematical model is used to predict the development of a wildfire using a cellular automaton. Physical factors from fire behaviour fuel model data are used as specified by Rothermel in 1972 [9] as an input to the automaton. The automaton was implemented in C for performance reasons and only uses libegeotiff as an external library. Our results are visually compared with a similar model (Jiang et al) that uses a larger set of physical factors. Our results differ from Jiang et al under the same starting location and fire behaviour fuel model, which is likely due to the fact that certain variables such as temperature, humidity, elevation and wind were not considered in the model. However, these variables can be incorporated in future works as per Jiang et al[6] and Andrews [4].

Keywords: cellular automata, Rothermel model, landfire, fire spread, wildfire

1 Introduction

Wildfires are devastating forces of nature. Some of the costliest wildfires in the United States alone occurred in the period from 2017-2021. These fires caused between 2 to 10 billion dollars in damages each, destroyed thousands to ten thousands of structures each and killed more than 100 people in total. [2]

It must be noted that wildfires occur naturally so much that some ecological species have co-evolved with wildfires. However 85% of wildfires are caused by humans [2] and the amount of area burned per year by wildfires is increasing, likely due to human factors [8]. To protect resources and cities, increase air quality, maintain ecological balance and manage future fires, knowing where the fire edge will be in the future is important to effectively fight wildfires and keep firefighters safe [8].

There are software models that attempt to predict wildland fire evolution, such as FARSITE or FlamMap [3] and BehavePlus [1]. These models can take a long time to run or be inaccurate [8]. The goals of this thesis work are to create a cellular automaton with only a fire behaviour fuel model (FBFM) and see visually how similar it is to a model that uses more parameters.

This paper uses Rothermel's approach to fire modelling as described in Andrews [4] to make a cellular automaton as described by Jiang et al [6]. A cellular automaton is used for its reproducibility and simplicity. This is done in the C programming language. Data is used from the United States' Wildland Fire And Natural Resource Management (LANDFIRE) website. Ziel et al shows how to convert this chart to fuel model data [11] . This fuel model is used in an automaton to create the resulting image after a certain amount of iterations.

State	Name	Description
0	NOT BURNED	Not set on fire
1	CATCHING FIRE	Catching fire but not yet spreading
2	BURNING	Burning and able to ignite neighbouring cells
3	EXTINGUISHING	Will go to extinguished next iteration
4	EXTINGUISHED	Not on fire and not spreading
5	INCOMBUSTIBLE	Cannot be set on fire

Table 1: Burning states. Each cell starts in NOT BURNED or INCOMBUSTIBLE. Then the NOT BURNED cells can turn into CATCHING FIRE when there are neighbouring BURNING cells. Cells that are CATCHING FIRE can go to EXTINGUISHING or BURNING, dependent on their rate of spread. Cells that are BURNING turn into EXTINGUISHING when surrounded by cells in state 2 or higher. EXTINGUISHING cells turn to EXTINGUISHED in the next iteration.

Symbol	Parameter	Description
S_B	Burn state	State of the cell as in Table 1
S_C	Combustion status	How close this cell is to burning
R_S	Rate of spread	How fast this cell spreads fire
A_B	Burnt area	How much burnable area this cell has burnt
t_B	Burnt time	For how long this cell has been burning

Table 2: Cell parameters. These Parameters are used for calculations on whether to jump to a different state.

2 Background

The model is a cellular automaton as in Jiang et al[6]. In essence that means that we have a 2 dimensional grid of cells with values that can be iterated upon. For a more detailed view on cellular automata refer to Neumann [10] and for their application in wildfire modelling specifically to Karafyllidis et al [7].

In our case a grid is created with as many cells as pixels in the fire behaviour fuel model (FBFM). Each cell can be set to one of five states as shown in Table 1. Each cell starts in the NOT BURNED state, except for a single cell that starts in the BURNING state. The neighbouring cells can then catch fire, burn and eventually extinguish.

Each cell has the parameters described in Table 2. These were adapted from Jiang et al [6].

To catch fire, each cell is assigned a combustion status (S_C). Let N_i denote each of the cells' up to 8 Moore neighbours, Δt denote the time difference of each iteration and L_C the length of a cell, in our case the length is 30 meters for every cell, since that is the accuracy of the FBFM. Now the time it should take to burn a cell is $\frac{L_C}{R_S}$, since R_S is in distance over time. Therefore let us set R_{MAX} as the maximum R_S of all cells and a constant k, so that

$$\Delta t = k\frac{L_C}{R_{MAX}}$$

as given in Jiang et al [6]. Setting $k = 1.0$ ensures that at least one cell will be burnt every iteration. Setting $k = \frac{1}{R_{MAX}}$ ensures that we burn at most one cell every iteration. Now we can determine the new combustion status S_C every iteration:

$$S_C = \Delta t \frac{\sum_{N_i}^{i \in \{N,E,S,W\}} R_{S_i}}{L_C} + \Delta t \frac{\sum_{i}^{i \in \{NE,NW,SE,SW\}} R_{S_i}}{\sqrt{2} L_C}.$$

This has been taken from Jiang et al [6]. When S_C is above a threshold, i.e. 1.0, then that cell jumps from state NOT BURNING to CATCHING FIRE. Once in state CATCHING FIRE, then the cell will spread fire until the entire cell is burning and it turns to state BURNING. This is done as follows:

$$A_B = A_{B_{prev}} + \Delta t \cdot R_S.$$

Here we have added to the value of A_B from the previous iteration. Then we check whether

$$A_B > \frac{L}{\sqrt{\pi}}.$$

If so, the state will go to BURNING. When BURNING, the cell turns to EXTINGUISHING when fully surrounded by BURNING, EXTINGUISHING, EXTINGUISHED or INCOMBUSTIBLE cells. Additionally, a cell that is BURNING or CATCHING FIRE will turn to EXTINGUISHING when burning for more than 30 minutes, that is when the total iterations in state BURNING or CATCHING FIRE is more than $\frac{30}{\Delta t}$. When in the EXTINGUISHING state, the cell will turn to EXTINGUISHED in the next iteration.

3 Experimental validation

3.1 Setup and data acquisition

A Scott Burghan fire behaviour fuel model (FBFM) map (LC20_F40_200.tif) is downloaded from LANDFIRE with latitude range [34.11,34.07] and longitude range [-118.50,-118.47] as was done in Jiang et al [6]. This map can be loaded into the memory of a C program using functions from the libgeotiff library, however extracting the raw pixels of this image shows that this image is rotated. This rotation occurs from the difference in length of latitude and longitude away from the equator. This image can be reprojected using the GDAL package in Linux or using the QGis application to the projection EPSG:3857 (WGS 84 / Pseudo-Mercator) so that the image's pixels are aligned to the north. The fuel model data is extracted from each pixel and converted to a fuel model as specified by Ziel et al [11]. Since there were some pixels outside of the FBFM that were there as a projection artifact, they have been converted to INCOMBUSTIBLE. They could be cut out to improve the speed of the model, though this was not necessary. Before simulation, the rate of spread must be converted to a map.

We could not find an explanation in Jiang et al [6] on how the spread rate was calculated. We used Rothermel's method [9] as described by Andrews [4] to convert the FBFM to a rate of spread. The FBFM offers the parameters specified in Table 3. A few more parameters are described in Andrews [4] and specified in Table 4. It must be noted that our model has not assimilated wind speed nor slope. Andrews then proceeds to give a fraction as to what the rate of spread will be.

This is given by Andrews [4] and Rothermel [9] as

$$R_S = \frac{I_R \xi (1 + \phi_w + \phi_s)}{\rho_b \epsilon Q_{ig}}$$

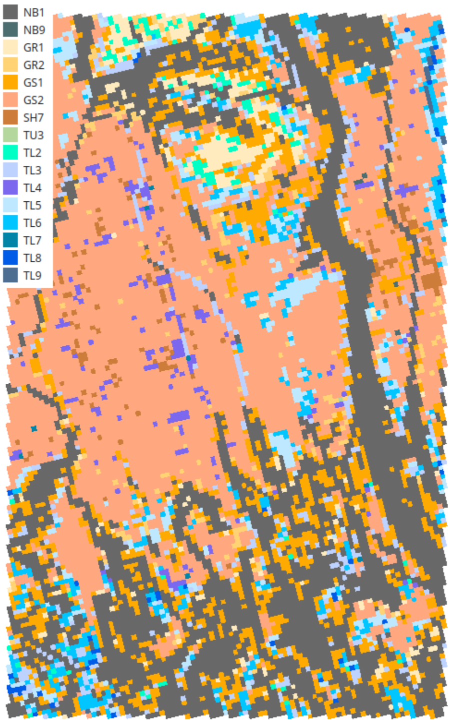

FIGURE 1: FBFM of the area from LANDFIRE. The legend refers to different fuel types as in Ziel et al [11]. For example GR1 refers to short patchy and possibly heavily grazed grass.

Symbol	**Parameter**	**Description**
w_1	load 1hr	Fuel mass per area with 1 hour burn duration
w_{10}	load 10hr	Fuel mass per area with 10 hour burn duration
w_{100}	load 100hr	Fuel mass per area with 100 hour burn duration
w_{herb}	load herb	Fuel mass per area of herbs
w_{woody}	load woody	Fuel mass per area of wood
w_{total}	load total	Total fuel mass per area
σ_1	sav 1hr	Surface area to volume ratio of material with a 1 hour burn duration
σ_{herb}	sav herb	Surface area to volume ratio of herbs
σ_{woody}	sav woody	Surface area to volume ratio of wood
M_x	moist extinct	Weighted average dead fuel moisture content for fire to stop spreading
δ	bed depth	Depth of fuel bed, i.e. average size of surface fuels
h_{dead}	dead heat	Heat content of dead fuel
h_{live}	live heat	Heat content of live fuel

TABLE 3: Fire behaviour fuel model (FBFM) parameters. Ziel at al [11] shows what values these have for different fuel types.

Symbol	Parameter	Description
S_T	Total mineral content	Total mineral mass to fuel mass proportion
S_e	Effective mineral content	Total mineral mass without silica to fuel mass proportion
ρ_p	Oven dry particle density	Particle density without moisture
M_f	Moisture content	Total water mass to fuel mass proportion
U	Wind velocity	Velocity of the wind at midflame height
$tan(\phi)$	Slope steepness	Vertical rise over horizontal distance

TABLE 4: Constant parameters, these parameters do not depend on the fire behaviour fuel model (FBFM), but are used in the model.

Input variable	Value	Parameter
M_f	30	Fuel moisture content
σ	σ_1	Surface area to volume ratio
w_0	w_1	Fuel load
h	$h_{dead} = h_{live}$	Low heat content of the fuel.

TABLE 5: Inputs to calculate the rate of spread. We took fuel moisture content to be 30%, according to Scott et al Table 4-3 [5], then all fuel should be mostly dead. This parameter should vary in reality. We take fuel load and surface area to volume ratio to both be equal to material that burns 1 hour. We therefore assume that the fire propagates over our cell size of 30 meters in 1 hour.

In our case this becomes

$$R_S = \frac{I_R \xi}{\rho_b \epsilon Q_{ig}}$$

Since we do not account for the wind nor slope. For the precise formulation of parameters see Andrews [4], this formula requires the following input parameters given by a FBFM: surface area to volume σ, oven dry fuel load w_0 and low heat content h. Since the σ_{herb} and σ_{wood} were not always defined for every entry of the FBFM, and additionally we are mostly interested in the border of the fire, we set $\sigma = \sigma_1$. For the same reason we set $w_0 = w_1$. That is, we assume the fire propagates our cell size of 30 meters over 1 hour. The table given by Ziel et al. [11] uses the same heat content of dead and live fuel for every fuel model. Therefore $h = h_{dead} = h_{live}$.

3.2 Experiment

Once the rate of spread is determined, the automaton can run. Our model ran for 20 000 iterations. The time coefficient $k = \frac{1}{58.8297}$ ($R_{MAX} = 58.8297$). This means that $\Delta t = 0.028439$ minutes. It took approximately a minute on a laptop with a 2.2GHz Intel i7 CPU and 8GB of RAM to simulate 9 hours.

3.3 Results

For the outputs, refer to figure 4

Our results vary from Jiang et al. This is likely due to the additional parameters Jiang et al. used such as temperature, humidity, wind and elevation. What does stay consistent is that the road and houses in figure 2 seems to stop both Jiang and our model.

FIGURE 2: Geographic Image of the area.

FIGURE 3: Calculated fire spread rate of area.

Iterations	Time coefficient Δt	Device
20 000	0.028439	2.2GHz Intel i7 CPU, 8GB RAM Laptop

TABLE 6: The setup used in the experiment.

Time simulating	Time simulated
1 minute	9 hours

TABLE 7: The real time spent simulating and how much time passed in the simulation.

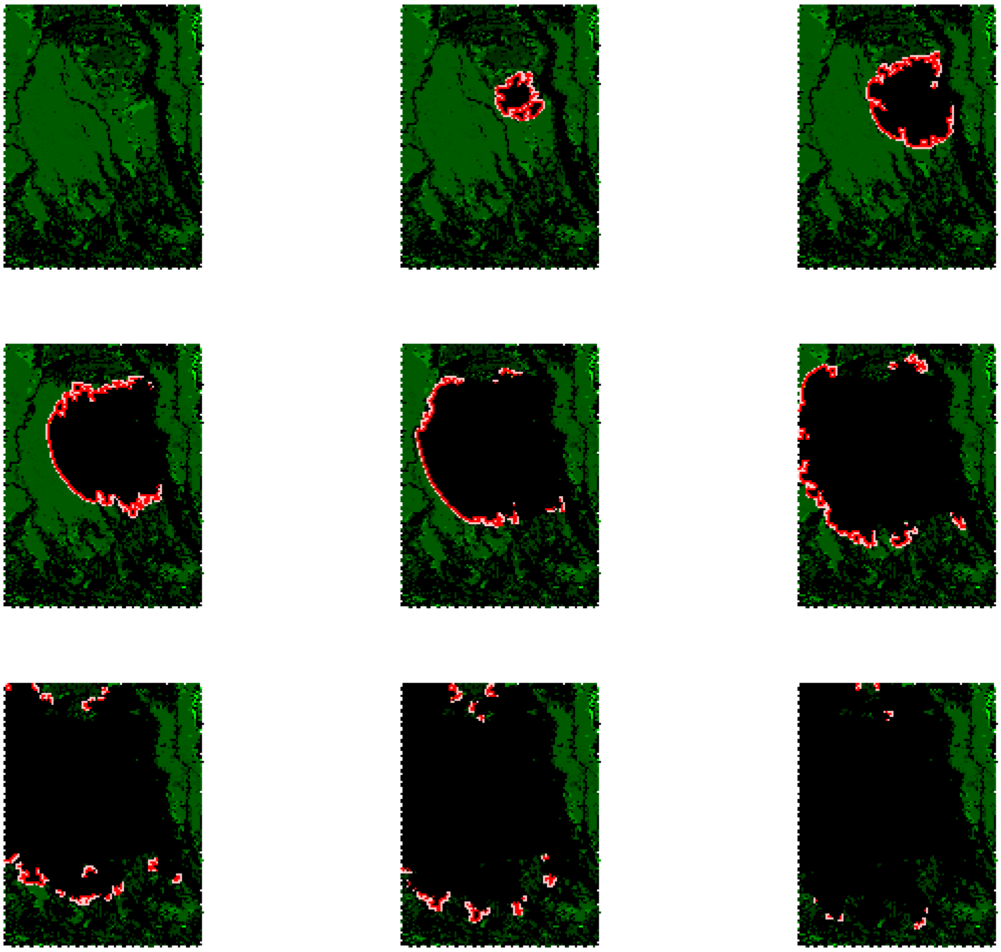

FIGURE 4: Spread of the fire every hour. Starting at 0 minutes in the top left and ending at 480 minutes in the bottom right.

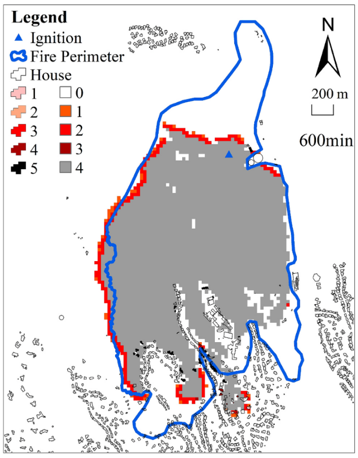

FIGURE 5: Fire perimeter of Jiang et al. at 600 minutes [6]

FIGURE 6: Elevation of the area in meters. Minimal elevation is 149 and maximal is 512. From low to high is from black to white.

3.4 Discussion

As can be seen from the results, the fire perimeter to the north, west and east from Jiang et al after 600 minutes of burning, are already reached after 180 minutes in our model. The border to the south is reached after around 420 minutes. We hypothesize that this is because of the difference in parameters, especially wind or elevation. Since the speed of the fire border towards the west was relatively faster than to the north. While figure 1 shows that the fire behaviour fuel model is relatively the same westwards as it is southwards. Figure 6 shows the elevation of the area. It can be seen that to the north and west of the starting area there is some increase in elevation, possibly leading to a different rate of spread.

4 Conclusions

We have found that a cellular automaton can predict fire spread using only a fire behaviour fuel model, though not as well as Jiang et al [6], who use additional inputs. This suggests that our model needs some modifications before being put into practice. The model takes a timescale of minutes to produce an output.

References

[1] Behave fire modelling software. https://www.frames.gov/behave/home.

[2] Facts + Statistics: Wildfires | III. https://www.iii.org/fact-statistic/facts-statistics-wildfires.

[3] Flammap and farsite fire modelling software. https://www.firelab.org/project/flammap.

[4] Patricia L. Andrews. The rothermel surface fire spread model and associated developments: A comprehensive explanation. *General technical report RMRS-GTR371*, 2018.

[5] Scott Joe H. Introduction to wildfire behavior modeling. national interagency fuels, fire, vegetation technology transfer. www.niftt.gov, 2012.

[6] Wenyu Jiang, Fei Wang, Linghang Fang, Xiaocui Zheng, Xiaohui Qiao, Zhanghua Li, and Qingxiang Meng. Modelling of wildland-urban interface fire spread with the heterogeneous cellular automata model. 135:104895.

[7] Ioannis Karafyllidis and Adonios Thanailakis. A model for predicting forest fire spreading using cellular automata. *Ecological Modelling*, 99(1):87–97, June 1997.

[8] Dani Or, Eden Furtak-Cole, Markus Berli, Rose Shillito, Hamed Ebrahimian, Hamid Vahdat-Aboueshagh, and Sean A. McKenna. Review of wildfire modeling considering effects on land surfaces. *Earth-Science Reviews*, 245:104569, October 2023.

[9] Richard C. Rothermel. *A Mathematical Model for Predicting Fire Spread in Wildland Fuels*. Intermountain Forest & Range Experiment Station, Forest Service, U.S. Department of Agriculture, 1972.

[10] John von Neumann. *Theory of self reproducing automata*. University of Illinois press, 1966.

[11] Robert Ziel and W Matt Jolly. Performance of fire behaviour fuel models developed for the rothermel surface fire spread model.