https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

An Efficient Method to Simulate Wildfire Propagation Using Irregular Grids

Conor Hackett[1], Rafael de Andrade Moral[2], Gourav Mishra[3], Tim McCarthy[3], Charles Markham[4]

[1]Hamilton Institute, National University of Ireland Maynooth, Kildare, W23 A3HY, Ireland.
[2]Department of Mathematics and Statistics, National University of Ireland Maynooth, Kildare, Ireland.
[3]National Centre for Geocomputation, National University of Ireland Maynooth, Kildare, W23 NPY6, Ireland.
[4]Computer Science Department, National University of Ireland Maynooth, Kildare, W23 A3HY, Ireland.
Correspondence to: Conor Hackett (conor.hackett.2018@mumail.ie)

Abstract. Climate change and land-use changes are projected to make wildfires more frequent and intense, with a global increase of extreme fires of up to 14 % by 2030, 30 % by the end of 2050 and 50 % by the end of the century (Sullivan et al., 2022). This latest information has increased interest of how the large scale, often catastrophic, events can be reduced and more effectively managed. One critical area revolves around real-time fire line prediction and how resources can be better deployed to reduce the propagation of wildfires. This paper explores mathematical models for fire propagation on a fully configurable grid using the Irregular Grid Software (IGS) developed. The configurable grid allows cross comparison of both regular grids such as square, hexagonal, triangular, and irregular grids such as a randomly seeded Voronoi diagram and a flammable resolution grid (FRG). The FRG is adapted to focus attention on areas of higher importance which provides greater precision at the cost of extra computing time. The irregular grid approach and ForeFire, an existing industry standard program were compared. The comparison included simulations of wildfires located in the Wicklow Mountains, in Ireland, a region used by the fire services for exercises. The performance of the grid-based techniques was examined using a set of experiments to characterise the model's response to key factors such as wind, elevation, and fuel type. The results show that the IGS runs on average 34 times quicker than ForeFire while retaining an average result similarity of 80% with ForeFire. In this paper sections 1 and 2 will give an overview on existing research on wildfires and wildfire modelling. Section 3 will describe the resources that were necessary to model wildfire propagation. Section 4 explains how these resources were used to build the IGS. Section 5 compares different gird types produced using the IGS, while section 6 compares the IGS to ForeFire. Sections 7 and 8 discus these results.

Keywords
Wildfire, Forest Fire, Bushfire, Simulation, Modelling, Rothermel Model, Irregular Grid, Voronoi Diagrams, Cellular Automata ForeFire, DecaMap

1 Introduction

All fires, including wildfires, require four components to support combustion (Helene et al., 2019). These are fuel (vegetation), an oxidising agent (oxygen in the air), heat (initiates the process) and a chemical chain

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

reaction (to sustain combustion). Combustion is a chemical process in which fuel and oxidiser combine to form heat greater than the fuel and oxygen mixture's flash point, if there is sufficient heat generated to overcome heat loss, combustion is sustained in a chain reaction. Together they make up the "fire tetrahedron"; a fire will cease to exist if any component is diminished within the tetrahedron (Figure 1).

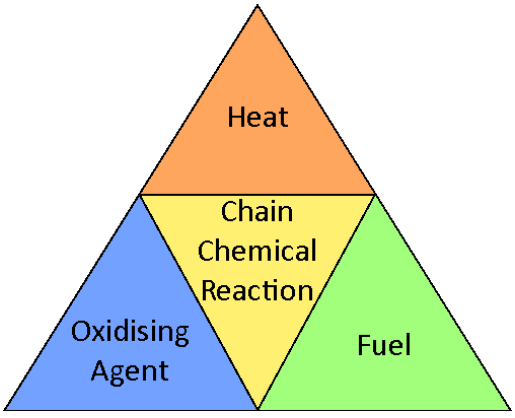


Figure 1: Net of the "fire tetrahedron", showing the four components required to support combustion.

A wildfire is an uncontrolled fire that burns throughout an area, this includes forest fires and bushfires (Haghani et al., 2022). Usually, they begin in rural areas where there is a higher density of combustible vegetation. They can spread quickly and sometimes threaten urban areas (Park et al., 2023). Wildfires can be caused by both natural events such as lighting, and through man-made actions such as deforestation and arson (Jiao et al., 2023; dos Reis et al., 2021). The frequency of wildfires has increased in recent years, this is most likely due to climate change creating drier terrain allowing fires to burn more easily (Halofsky et al., 2020). Wildfires have begun to have more devastating effects due to their increased intensity (Keeley and Syphard, 2021). Countries such as Ireland, where wildfires were previously uncommon due to its wet climate, have seen a sharp increase in the number of wildfires recorded (McElwain and Sweeney, 2003; Hawthorne and Mitchell, 2018).

The devastation of wildfires can be measured in different ways such as fatalities, ecological, environment and economic damage. In California, USA and Australia wildfires tend to be large fast burning fires that are life threatening (Keeley and Syphard, 2021; Blanchi et al., 2014). Wildfires also have an ecological impact; in Ireland, wildfires tend to be smaller slower burning fires occurring mainly in bogs (peatlands), which are home to many rare species of both flora and fauna (Prat-Guitart et al., 2019). The high abundance of gorse (*Ulex europaeus*) in bogs makes them a hotspot for wildfires as the plant is highly flammable [10]. Even though there are very different forms of fire that burn in the USA, Australia, and Ireland, it is still possible to model them all, but the parameters characterising fuel properties need to be changed. When wildfires burn through terrain, they release Carbon dioxide and other environmentally harmful gasses such a methane, further contributing to an increase in greenhouse gas emissions (Xue et al., 2024;Jones et al., 2019). The economic impact of wildfires includes damage to the agricultural and forestry sectors (Meier et al., 2023). When a wildfire spreads to an urban area it can also cause considerable damage to town infrastructure (Park et al., 2023).

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

2 Overview of wildfire propagation models

With the increasing severity and frequency of wildfires, the ability to model and predict wildfire propagation has become an invaluable asset to planners and firefighters (Penney et al., 2019). Planners can use this information to forecast and prevent fires, while firefighters can also use this information to find the optimal locations to apply an intervention. These interventions include construction of firebreaks, controlled burning, or application of water using fire engines or Helicopters equipped with a Bambi-Bucket (SFI-Defence Organisation Innovation Challenge, Challenge 1, 2024).

One of the most cited wildfire models is the Rothermel Model (Rothermel, 1972;Andrews, 2018). It was developed by Richard C. Rothermel in 1972 to help forest managers predict the behaviour of wildfires. The Rothermel model sits as the foundation on which most other fire spread models are built.

The Rothermel model is a physics-based model built around the Rothermel equation. This takes as input multiple environmental factors and produces an estimated rate of spread as an output Eq. (1). In the Rothermel model the numerator measures heat generation while the denominator measures heat loss. The Rothermel model uses R as a measure of how fast a fire is spreading. It does this on a 1-dimensional line, but this can be expanded to more dimensions by measuring R on separate lines along each direction of interest. An easy way to interpret this is with a section of burning terrain acting as a heat source, while the ground beneath this terrain acts as a heat sink. If there is a surplus amount of heat produced it will spread to neighbouring terrain (Figure 2). The Rothermel model is normally presented using the United States customary units. In this paper these values were converted into SI units within the fire simulations developed. It is written as:

$$R = \frac{I_R \xi (1 + \Phi_w + \Phi_s)}{\rho_b \varepsilon Q_{ig}} \tag{1}$$

where $R \geq 0$ is the rate of spread in m/s (ft/min), $I_R \geq 0$ is the reaction intensity in J/m^2/s (Btu/ft^2/min), $0 \leq \xi \leq 1$ is the propagating flux ratio, $\Phi_w \geq 0$ is the wind factor, $\Phi_s \geq 0$ is the slope factor, $\rho_b > 0$ is the bulk density in kg/m^3 (lb/ft^3), $\varepsilon > 0$ is the effective heating number and $Q_{ig} > 0$ is the heat of preignition in J/kg (Btu/lb).

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

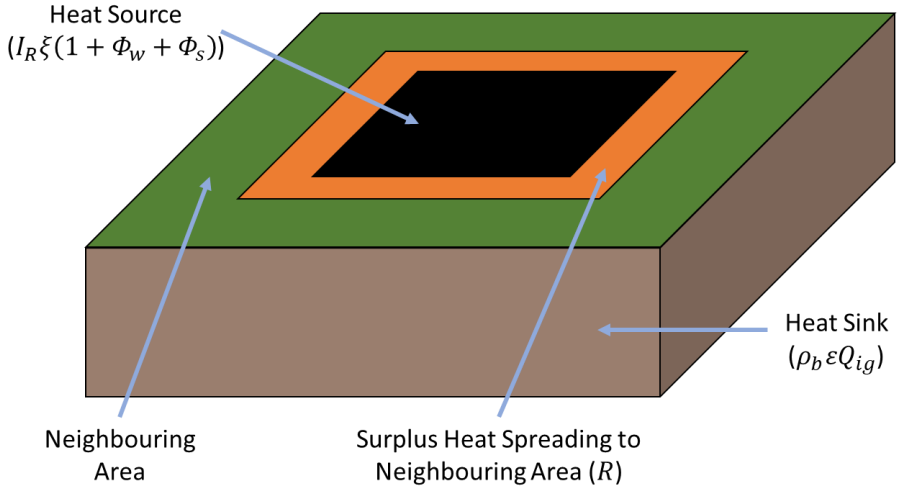


Figure 2: Example of how the Rothermel model functions, where the heat source is the location of the fire (where heat is being generated from combustion), the heat sink absorbs heat from the heat source slowing down propagation, and the surplus heat is used to spread fire to the neighbouring area.

Many novel techniques have been used to optimise the application of the Rothermel model such as the use of genetic algorithms (GA) (Pereira et al., 2022). GAs have been used to refine the input parameters for the Rothermel model to increase the precision. This was done by first having a range of pre-set input values which would be inputted into a GA. The GA would then compare the fire line produced by the different input values to the real rate of spread from datasets obtained through experimental prescribed fires in controlled conditions. The sets of input parameters that produced a rate of spread most closely resembling the real rate of spread were used to generate new sets of input parameters through the process of crossover and mutation. Repeating this method with the new generation of input parameters will continuously improve them, increasing fire prediction quality by 93.66% in the experiments from the paper.

There are other approaches to modelling wildfires than just the Rothermel model. Kalman Filters have been used in conjunction with unmanned aerial vehicles (UAVs) to produce results to a similar level of precision as state-of-the-art methods while requiring a lower computational cost (Lin et al., 2019). This was done by sampling the temperature in multiple areas of a wildfire using UAVs. These samples were then processed by a Kalman Filter to estimate propagation behaviour and rate of spread using the temperature gradients from the samples.

Deep Convolutional Neural Networks (CNNs) have also been utilised to predict wildfire spread in California, USA (Green et al., 2020). The CNN learns propagation patterns of wildfires from training on historic datasets. The historic datasets contain a timeseries of satellite images within the Western Sierra Nevada Mountains which the CNN is trained on. The CNN then predicts the wildfire in 24-hour periods. This method produces realistic results, with accuracies ranging from 78% to 98%.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

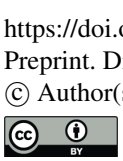

Natural Hazards and Earth System Sciences
Discussions

The Rothermel model is used as an optional fire propagation model in platforms such as Cell2Fire and ForeFire. There are three types of propagation models: cellular automata, cellular and continuous. Cellular automata models are restricted to a grid and a strict set of rules, where each cell is in a finite number of possible states and the state of one cell can influence the state of its neighbouring cells. Cellular models are also restricted to a grid, but the state of a cell is continuous and not confined to only influencing its neighbours (San Martin and Torres, 2023). Continuous models propagate over continuous space and therefore are not restricted to a grid or finite states.

Cell2Fire is a program that utilises cellular automata to simulate wildfire spread (Pais et al., 2021). Traditionally cellular automata have rules that express how the state of one cell interacts with its neighbouring cells. Cell2Fire can be given a fire spread model, a starting fire and information about the environment to begin simulating fire spread. As Cell2Fire is a program built on cellular automata it restricts the spread of fire to a regularly shaped square grid using the Moore neighbourhood where each cell has 8 neighbours (corner neighbouring cells inclusive) (Małecki, 2017). Cell2Fire generates the common ellipse shape of a wildfire for each cell based on the rate of spread. Once any point on the ellipse touches the centroid of a neighbouring cell that cell will also begin generating its own ellipse to calculate rate of spread. The uniform grid presents trade-off between resolution and computation time. Low resolution grids are fast but have the potential to be less precise. High resolution grids are computationally intensive. This will be discussed in further detail later in the paper.

ForeFire is another program that simulates wildfire spread but does this over a continuous space instead of the discrete space used in cellular automata models (Filippi et al., 2014). ForeFire uses a fire model, such as the Rothermel model to compute propagation. It also takes spatial environmental data over a selected region of terrain as an input. The environmental data contains wind data in the form of zonal (west to east) and meridian (south to north) wind speeds, elevation data regarding the terrain and a numerical index representing different land cover types for the terrain. All data inputted into ForeFire is in the form of metric units, if the fire spread model required imperial or United States customary units (such as the Rothermel model), they need to be converted before calculating fire spread and converted back to metric units once fire spread has been calculated. ForeFire is seeded with information about the fire's starting location, the spatial resolution, and the duration of the simulation. During the simulation ForeFire calculates fire spread using the current fire location and environmental inputs. The increased fire spread perimeter is comprised of nodes called markers that ForeFire uses to continue spreading for the next iteration. Each marker has a propagation vector representing its movement direction and speed. The speed at which the marker moves at is determined by the fire spread model. If the distance between markers is greater than a selected spatial resolution, more markers will be created and redistributed along the fire front. If the distance between markers is less than quarter of the spatial resolution, they will be merged into one marker (Figure 3). On completion the software returns, the final set of coordinates (known as markers) describing the location of the fire front. The use of markers allows for increased resolution, limited by the resolution of the input data, as the markers are not restricted to a regular grid. However, the continual placing, removing, and moving of markers add extra computational burden on the simulation.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024



Natural Hazards and Earth System Sciences
Discussions

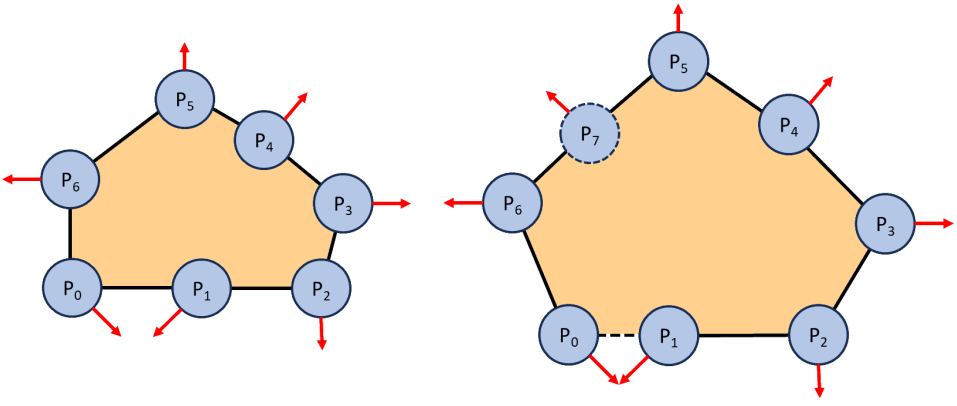


Figure 3: Example of ForeFire markers at timestep 0 (left) and timestep 1 (right). In the example the fire line (black line), markers (blue circles), direction of spread for markers (red arrows) and where the fire has been (orange) are all shown. In this example markers P_0 and P_1 will combine into one marker as the distance between them is quarter the spatial resolution. Marker P_7 will be created as the distance between markers P_5 and P_6 is greater than the spatial resolution.

Fire spread for the software described in section 4 of this paper was mainly based off other research describing how cellular automata can be used to spread fire directly between cells (Zhang et al., 2021). Similarly this paper used the Rothermel model to estimate the rate of spread for the fire. When the fire in one cell's centroid had spread to a neighbouring cell's centroid the fire would also begin propagating from that cell. The distance the fire had spread between cells was also used to estimate the area of the cell that is on fire. These are all techniques that will be discussed later in section 4 of this paper where software is developed to produce predicted wildfire outputs that are of a similar shape to ForeFire but more computationally efficiently.

3 Resources

To compare ForeFire with our novel approach the Irregular Grid Software (IGS), the IGS code was designed to use the same input parametrisation. A satellite image of the region around Lough Dan, County Wicklow, Ireland, was used as input data for both ForeFire and the IGS.

Using the satellite data and a random forest machine learning model, the satellite data was used to produce a land cover map. Each pixel on the landcover map referenced an associated land cover type including pastures, sparsely vegetated areas, mixed forests, moors and heathland, urban fabric, water bodies, clouds, and cloud shadows.

A near cloud free satellite image (product) taken by the Sentinel-2 satellite in April 2022 was downloaded from the European Space Agency (ESA) sentinel-hub portal, a satellite image hosting platform (EO Browser, 2024). Sentinel-2 products contain multiple bands which capture electromagnetic radiation (light) reflected off the Earth's surface. A Level 1-C product (top of atmosphere reflectance values) was used in this study. The data was then atmospherically corrected to bottom of atmosphere values using the Semi-Automatic Classification tool in

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions
Open Access
EGU

the geographic information software QGIS which follows the Dark Object Subtraction (DOS) algorithm (Chavez, 1988; QGIS, 2024; Gilmore et al., 2015). This is due to Level1-C products originally containing information regarding the both the Earth's surface and atmosphere which may not accurately depict the actual conditions on the surface of Earth. Three additional bands for highlighting vegetation, urban areas and soil (i.e. normalised difference vegetation index (NDVI), normalised difference built up index (NDBI) and normalised difference tillage index (NDTI) respectively) were generated from the original Sentinel-2 bands. These indices are simple ratios calculated from the original multispectral bands and are known to help distinguish various land cover classes, i.e. NDVI for vegetation, NDBI for urban areas and NDTI for bare soil. Areas of known landcover were digitized to generate seventy polygons in total, which were further equally split into training and validation sets. A supervised random forest algorithm was then applied on the stack of seven original bands (red, green, blue, near infra-red (NIR), NIR narrow, short wave infra-red (SWIR) 1 and SWIR2) captured by the Sentinel-2 satellite and the three additional indices to predict the land cover classes. The random forest classification algorithm is non-parametric in nature and known to improve estimates by averaging outputs from multiple decision trees randomly subset from the data. It has been successfully used for land cover mapping in previous studies and the algorithm was run using the RStoolbox library in the R programming environment (Sibanda and Ahmed, 2021; Abdi, 2020;Pringle et al., 2018). The cloud covered pixels in the area were classified based on a simple thresholding of the blue band (Baetens et al., 2019). The predicted land cover map was compared against the validation dataset which provided an overall accuracy of 97%. The predicted land cover map for the area under study is shown in (Figure 4).

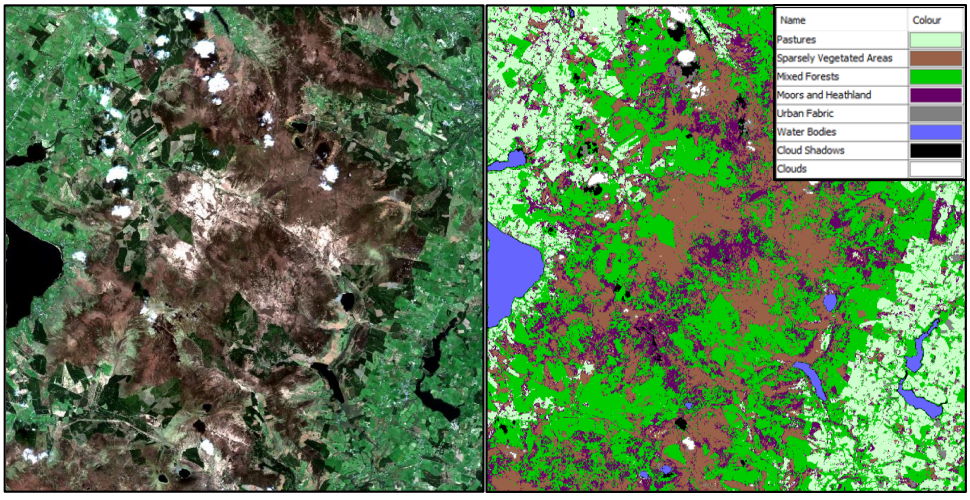


Figure 4: Left: Sentinel-2 based true colour image of the study area; and right: predicted land cover map for April 2022 (Copernicus 2022).

Each type of land cover is mapped to a set of physical attributes. These attributes are contained in a lookup table used by both programs. ForeFire contains one of these files by default and this was also used by the IGS (Fuels.ff Fuel Attribute Table, 2024). ForeFire's fuel file was indexed based on Corine Land Cover classes (Home :: Corine Land Cover classes, 2024). The satellite image of landcover types were then matched to the

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Corine Land Cover classes to get the correct fuel attributes. The fuels file contained values for fuel particle density (kg/m^3), fuel particle moisture content, fuel particle surface area to volume ratio (m^{-1}), fuel height (m), the oven-dry fuel load (kg/m^2) and fuel particle low heat content (J/kg) for the different land cover types. Each of the listed fuel properties are required to run the Rothermel model. The land cover map had two additional fuel types for when it was not possible to identify the land type due to clouds or shadows. These fuel types were not present in the simulation area, as it was cloud free, so they did not affect either program.

The Sentinel-2 satellite data and a geographic information software, Snap Desktop's elevation band generator was used to produce an elevation map of the area (Copernicus Data Space Ecosystem, 2024; Download - STEP, 2024). The elevation map gives the height in meters of elevation above sea level for each pixel in the satellite data.

For experimental simulation, the wind effects were experimentally manipulated to allow a comparison between ForeFire and the IGS under the same conditions. However, an API developed also allows software such as Windy to get live, forecasted, and historical wind data for simulations (Windy, 2024). Wind data is then split into zonal (west to east) and meridian (south to north) wind speeds which are mapped at each pixel across the satellite data.

ForeFire contained a tool that allowed the land cover, elevation, wind data and metadata to be combined into a single NetCDF file (Tools, 2024; Network Common Data Form (NetCDF), 2024). This same file was then used by both ForeFire and the IGS.

4 Methods

The aim of the IGS was to generate a grid-based fire spread model that could be compared to ForeFire. The use of a grid with static points allows a model to compute fire spread without having to continually move and add markers during the simulation to predict fire spread as found in ForeFire.

A Voronoi diagram was used to create an irregular grid. They have been used to simulate the geographic spread of disease in the past (Hackett et al., 2021). A Voronoi diagram takes in a set of points called sites and from these sites it generates edges located equidistant between other sites. This creates tessellating polygons where every point within a polygon's perimeter is closer to that polygon's site than any other polygon's site (Figure 5). This means that each edge separating polygons is equidistant to both polygons' sites. The Voronoi diagram was generated using the efficient Fortune's algorithm implemented in the Foronoi (not spelled Voronoi) Python library (foronoi, 2024; Fortune, 1987). One of the biggest advantages of using a Voronoi diagram is the ability to create irregular simple tessellating shapes which allows for efficient computation of fire spread between polygons.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions
Open Access
EGU

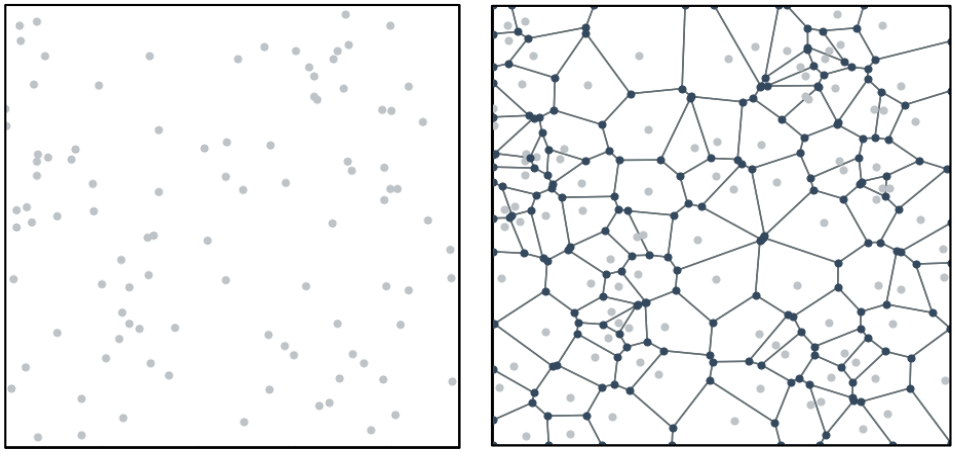


Figure 5: Positions of randomly placed sites (left) and the resulting generated Voronoi diagram of polygons (right).

The NetCDF file containing input data, created using ForeFire, is imported into the Python program using the Snappy Python library (How to use the SNAP API from Python, 2024). Snappy allowed the NetCDF data to be read into Python arrays used for computation. The Voronoi diagram generated was overlayed on top of the NetCDF data. The elevation, wind, and fuel values (per fuel index) were recorded for every pixel in all polygons of the Voronoi diagram. This was done using the scan-line polygon fill algorithm to extract environmental data from the pixels contained within the polygon (Al-Rawi, 2014). The mean elevation, wind and fuel values of each polygon were then saved.

The cellular based model starts with a fire located at the site of a Voronoi polygon. The propagation of this fire towards the site of each bounding polygon is modelled using the Rothermel model. When the fire reaches the site in the bounding cell, the fire then progresses towards the new bounding cells. Fires may propagate towards each other simultaneously. Fire progress is recorded as a ratio of how far it has propagated towards neighbouring polygon sites and the total distance between the two polygon sites. This is denoted as the propagation ratio. When the Rothermel model is ran, the propagation ratio from the ignited polygon to its neighbours will increase if the rate of spread is positive (Figure 6) Eq. (2). When the propagation ratio reaches 1, the neighbouring polygon will become ignited and will begin also spreading fire as this represents the fire spreading the entire distance between both polygon sites. The propagation ratio is calculated as follows:

$$\left(P_{ij}\right)_{t+dT} = \frac{\left(\left(P_{ij}\right)_t d_{ij}\right) + \left(R_{ij} dT\right)}{d_{ij}} \tag{2}$$

where i is an ignited polygon, j is a polygon that neighbours the ignited polygon, t is the current time (s), $dT > 0$ is the timestep (s), $0 \leq (P_{ij})_{t+dT} \leq 1$ is the propagation ratio between polygons i and j at time $t + dT$, $0 \leq \left(P_{ij}\right)_t \leq 1$ is the propagation ratio between polygons i and j at time t, $R_{ij} \geq 0$ is the rate of spread between polygons i and j calculated by the Rothermel model (m/s) and $d_{ij} > 0$ is the distance between the sites of polygons i and j in metres.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

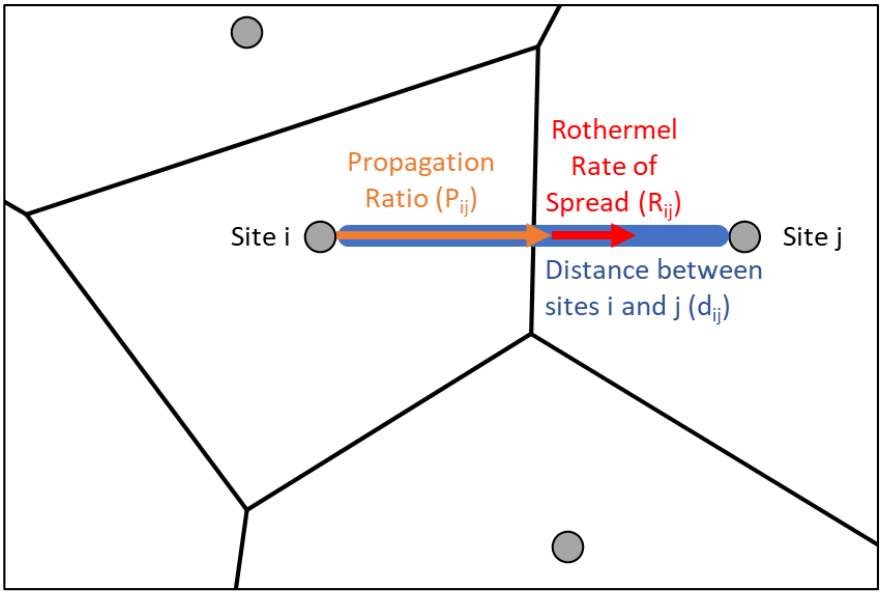


Figure 6: Example of the fire spreading from polygon i to j. Where the blue line represents the total distance (d_{ij}) between sites i and j (m), the orange arrow represents the propagation ratio (P_{ij}) of how far fire has spread from site i to j and the red arrow represents the fire's rate of spread determined by the Rothermel model (R_{ij}) from site i to j (m/s).

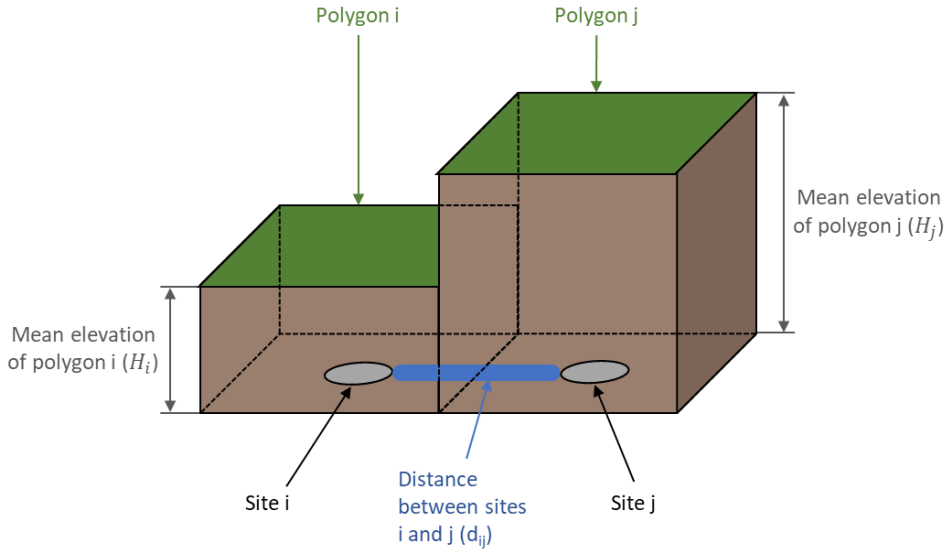
The Rothermel model requires data describing the fuel, the terrain (slope) and wind. Each polygon has fuel data associated with it, during fire spread the fuel data of the polygon that the fire is spreading to is used. The slope is found by getting the difference in elevation between the neighbouring polygon and ignited polygon which is then divided by the distance between the polygon sites Eq. (3) (Figure 7):

$$S_{ij} = \frac{H_j - H_i}{d_{ij}} \quad (3)$$

where i is the ignited polygon, j is the neighbouring polygon, S_{ij} is the slope from i to j, H_j is the mean elevation of polygon j (m), H_i is the mean elevation of polygon i (m) and $d_{ij} > 0$ is the flat distance between the polygon sites i and j (m).

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions
Open Access
EGU



300

Figure 7: Variables used in (3) to determine the slope from polygon i to polygon j.

The normal wind value is found by getting the dot product between the wind vector and the unit vector describing the ignited polygon and neighbouring polygon sites' coordinates Eq. (4) (Figure 8):

$$(W_{ij})_n = \begin{bmatrix} (\boldsymbol{W_j})_z \\ (\boldsymbol{W_j})_m \end{bmatrix} \cdot \frac{\vec{ij}}{\|\vec{ij}\|} \tag{4}$$

305

where i is the ignited polygon's site, j is the neighbouring polygon's site, $(W_{ij})_n$ is the normal wind from i to j (m/s), $(W_j)_z$ is the mean zonal wind speed of j (m/s), $(W_j)_m$ is the mean meridian wind speed of j (m/s) and $\vec{ij}$ is the vector from polygon site i to polygon site j.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

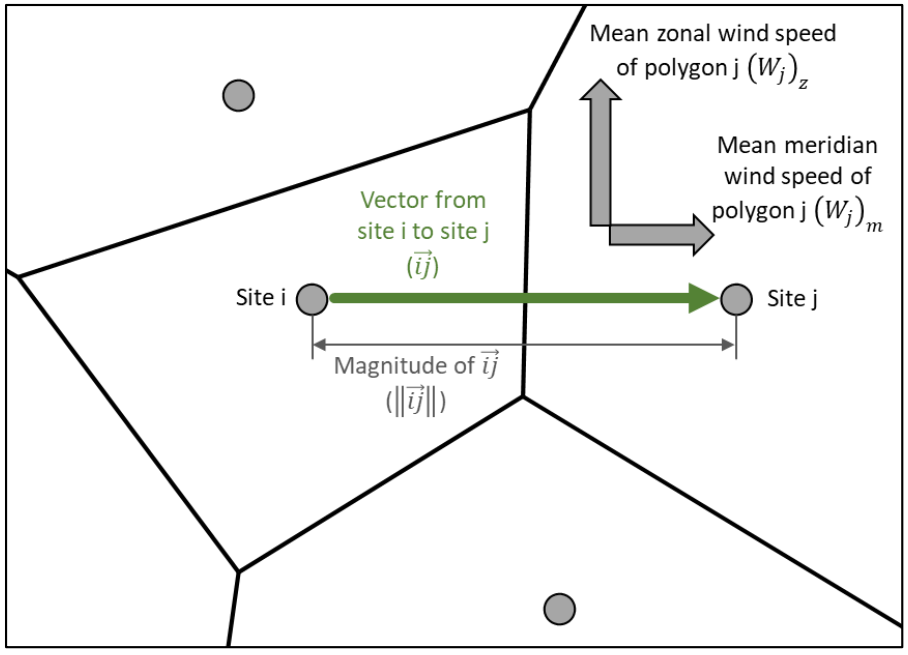


Figure 8: Finding the normal wind for fire spreading from site i to j by getting the dot product of both the mean zonal and meridian wind speed properties of polygon j and the unit vector of spread from site i to j.

A visualisation was created to show the spatial distribution of the polygons and evolution of the fire line. The polygons were rendered in distinct colours to represent their states. Polygons rendered red represented polygons that were currently on fire, black represented completely burnt, clear without colour represent polygons that had not caught fire yet and green represented the polygon(s) where the fire started. A 50% transparency was used to allow the underlying map data to be seen.

The produced Voronoi Diagram was then scaled and overlayed on top of the satellite image so each pixel coordinate in the Voronoi Diagram would correspond to a pixel on the satellite image. The area of the polygon was then found in pixel coordinates and converted to an area in m^2. For fire line calculations, the oven dry fuel load per m^2 was sourced from the fuel file and therefore the total fuel per polygon (kg) was found. The total fuel in the polygon could then be split into flammable vegetation, currently ignited vegetation and vegetation that had been completely burnt Eq. (5):

$$V_i = N_i - (I_i + B_i) \tag{5}$$

where i is a polygon, $N_i \geq 0$ is the total amount of fuel in polygon i (kg), $V_i \geq 0$ is the amount of remaining flammable vegetation in polygon i (kg), $I_i \geq 0$ is the amount of fuel in polygon i currently ignited (kg) and $B_i \geq 0$ is the amount of fuel in polygon i that has burnt (kg). To estimate the mass of a polygon that was currently on fire, the propagation ratios from the ignited polygon and its neighbours were found. The total propagation ratios of fire that had spread within the ignited polygon only, and the total half distances between

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

the ignited and neighbouring polygons were calculated. This value was then multiplied by the total amount of fuel in the ignited polygon, the amount of completely burnt fuel was then subtracted to find the amount of fuel that was currently on fire in that ignited polygon Eq. (6) Eq. (7):

$$q_{ij} = \left(P_{ij} d_{ij}\right) + \left(d_{ij}\left(P_{ji} - 0.5\right)\right) \tag{6}$$

$$I_i = N_i \frac{\sum_{j=1}^{n_i} \min\left(q_{ij}, \frac{d_{ij}}{2}\right)}{\sum_{j=1}^{n_i} \frac{d_{ij}}{2}} - B_i \tag{7}$$

where i is a polygon , j is a polygon neighbouring i, $q_{ij} \geq 0$ is the total distance fire has spread within polygon i between it and polygon j (m), $N_i \geq 0$ is the total amount of fuel in polygon i (kg), $I_i \geq 0$ is the amount of fuel in polygon i currently ignited (kg), $B_i \geq 0$ is the amount of fuel in polygon i that has burnt (kg), $0 \leq P_{ij} \leq 1$ is the propagation ratio between polygons i and j, $n_i \geq 0$ is the number of neighbours polygon i has and $d_{ij} > 0$ is the distance between polygons i and j (m). This was to ensure only fire that had spread within the cell was measured as any other spread in neighbouring cells was used to find the mass of their cell that was on fire instead.

The Reaction intensity was used as measure of the quantity of fuel in each polygon that has burnt, this measures the energy produced per unit area per time step. Using the equation for the Reaction Intensity, the energy produced per unit mass per unit time step $\left(E_{ij}\right)$ can be evaluated. This done by removing the net fuel load $(w_n)_j$ from the equation Eq. (8) Eq. (9) which as previously stated use United States customary units. $E_{ij}$was found using the fuel properties from neighbouring polygons and was converted to metric units:

$$(I_R)_{ij} = \Gamma'_j \, (w_n)_j h_j \eta_{M_j} \eta_{S_j} \tag{8}$$

$$E_{ij} = \Gamma'_j \, h_j \eta_{M_j} \eta_{S_j} \tag{9}$$

where i is a polygon, j is a polygon neighbouring i, $(I_R)_{ij} \geq 0$ is the reaction intensity of polygon j in J/m^2/s (Btu/ft^2/min), $(w_n)_j \geq 0$ is the net fuel load of polygon j in kg/m^2 (lb/ft^2), $E_{ij} \geq 0$ is the energy produced spreading from polygon i to polygon j per mass per time increment in J/kg/s (Btu/lb/min), $\Gamma'_j \geq 0$ is the optimum reaction intensity of polygon j in 1/s (1/min), $h_j \geq 0$ is the low heat content of polygon j in J/kg (Btu/lb), $0 \leq \eta_{M_j} \leq 1$ is the moisture damping coefficient of polygon j and $0 \leq \eta_{S_j} \leq 1$ is the mineral damping coefficient of polygon j. The mean energy for fuel burnt spreading to neighbouring polygons was calculated by multiplying the mean E_{ij} from all neighbouring polygons by the mass of ignited fuel in the polygon and the time step. The energy was then divided by the low heat content of the polygon to get the amount of ignited mass lost to burning Eq. (10):

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

$$dB_i = \frac{\frac{\sum_{j=1}^{n_i} E_{ij}}{n_i} I_i dT}{h_i} \quad (10)$$

where i is a polygon, j is a polygon neighbouring i, $E_{ij} \geq 0$ is the energy produced spreading from polygon i to polygon j per mass per time increment in J/kg/s (Btu/lb/min), $dB_i \geq 0$ is the rate at which ignited fuel in polygon i becomes burnt fuel in kg/dT, $n_i > 0$ is the number of polygons that neighbour i, $I_i \geq 0$ is the amount of fuel that is currently ignited in polygon i in kg, $dT > 0$ is the timestep in s and $h_i \geq 0$ is the low heat content of polygon i in J/kg (Btu/lb). This value was then added to the total amount of fuel that has been burnt in that polygon Eq. (11):

$$(B_i)_{t+dT} = (B_i)_t + dB_i \quad (11)$$

where i is a polygon, $dT > 0$ is the timestep (s), $dB_i \geq 0$ is the rate at which ignited fuel in polygon i becomes burnt fuel (kg/dT), t is time in the simulation (s), and $(B_i)_t \geq 0$ is the amount of fuel that is burnt in polygon i at t time (kg). Tracking both ignited and burnt fuel amounts paints a clear image of what areas are still on fire and which have ceased burning (Figure 9).

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions
Open Access
EGU

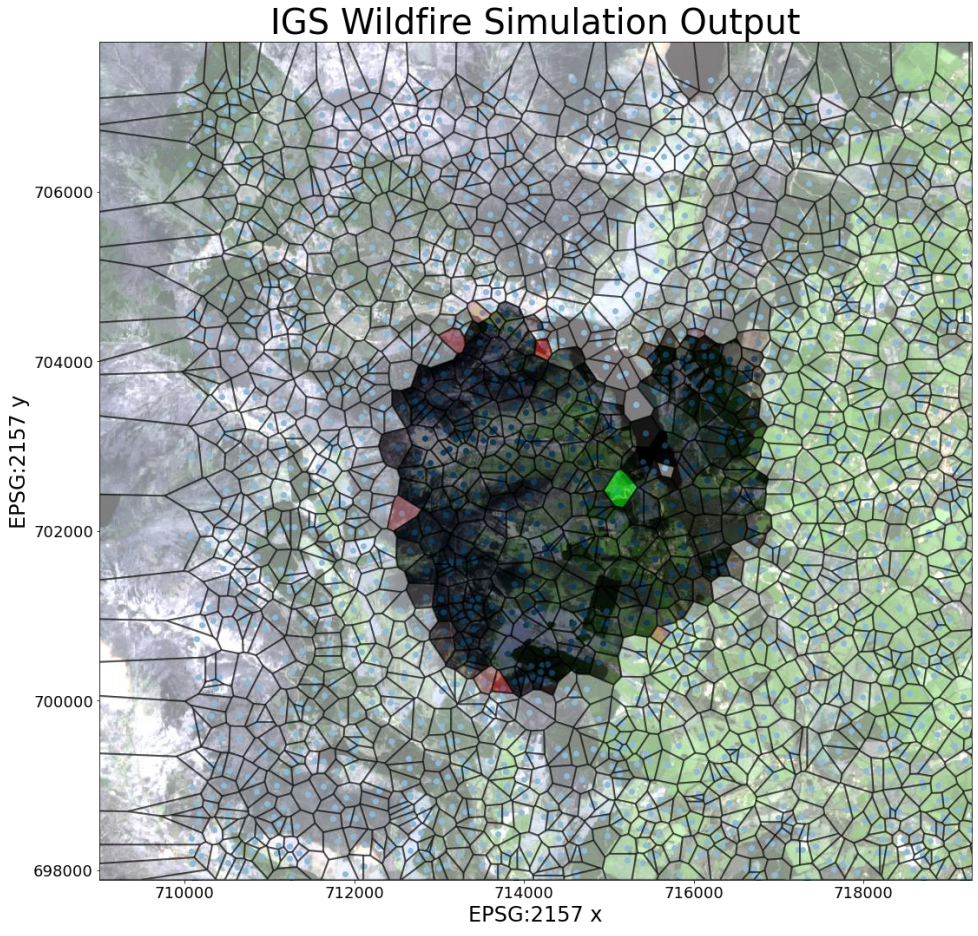


Figure 9: Sample simulation output after 100,000 seconds showing not burnt (transparent), fire source (green), burning (red) and burnt (black) polygons with coloured satellite image of the terrain in the background (Copernicus 2022).

The boundary of ignited and burnt polygons can then be converted into a fire line, representing the outer perimeter of the fire. This allows the IGS to be compared to other continuous programs such as ForeFire. The boundary polygons need to be found to produce a fire line. A straightforward algorithm was used to find the boundary polygons where each polygon was checked to see if it can spread the fire (a polygon is only able to spread fire if the fire spread model has reached that polygon's site). Once a list of polygons capable of spreading fire is found, the list can be shortened by checking if these polygons have one or more neighbours that cannot spread fire. This gives a list of boundary polygons that are on the fire line perimeter of the fire spread model. This list of boundary polygons was then converted to a list of boundary edges by checking the edges of each boundary polygon and recording any edges between a boundary polygon and a polygon that cannot spread fire. Getting the boundary edges of the boundary polygons helps reduce underpredicting of how far the fire has spread within a given boundary polygon. Edges each consist of two vertices. To begin a random edge is selected from the list of boundary polygons and a recursive algorithm finds the next edge to share vertices with the randomly selected

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

edge, removing it from the list and storing it in a stack. This process is repeated until no more shared vertices can be found and the stack then becomes an ordered boundary. The process of selecting a random edge continues until all edges have been removed from the list (Figure 10). Sometimes the vertices of edges don't line up perfectly, so a rough margin of error was allowed to ensure a completed shape. All edges shorter than the margin of error were removed to prevent other problems generating the completed boundary.

The ordered boundary can them be smoothed to appear more like real fire line, using a cubic spline. A Python library SciPy has a subpackage called interpolation (Interpolation (scipy.interpolate), 2024). This package can be used to generate cubic splines given separate lists of x and y coordinates. Traditionally cubic splines are performed where the values of x coordinates are ordered in increasing value. This will not work for the IGS's output as reordering the x coordinates would make the boundary unordered and ruin the boundary shape. To circumvent this, a parameter (c) was defined as the cumulative distance between all previous points in the ordered boundary. For example, the first point would have c equal to 0, while the second point a would be equal to the distance between the first and second point. The third point would have c equal to the distance between the first and second point summed with the distance between the second and third point. Due to c being a list of increasing values it was possible to perform a cubic spline on both the x and y coordinates separately using c (Cubic spline for non-monotonic data (not a 1d function), 2024). The x and y coordinates can them be re-combined to produce a cubic spline. With the cubic spline, the fire line becomes a smooth curve, more closely resembling how a real fire would appear (Figure 10).

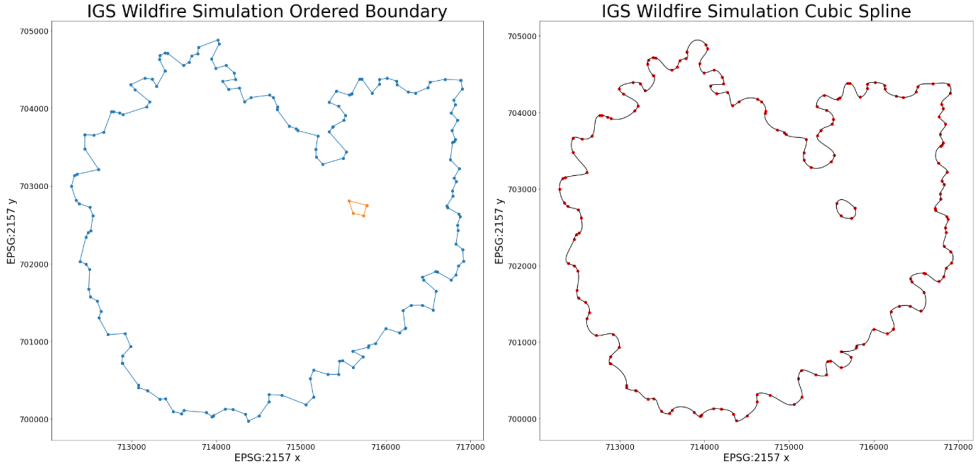


Figure 10: Left: Ordered boundary consisting of polygon edges from the IGS output that create an outer perimeter of the simulated wildfire; right: the ordered boundary smoothened by a cubic spline.

5 Comparison of Grid Types

Five different grid types were compared. All five grids were built using Voronoi diagrams. The five grid types consisted of: randomly plotted sites, triangular tessellating grid, square tessellating grid, hexagonal tessellating grid and a grid where more sites would be plotted closer to the fire source and where more flammable fuel is

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024


Natural Hazards and Earth System Sciences
Discussions

present; this grid is referred to as the flammable resolution grid (FRG). Each grid uses the Von Neuman neighbourhood for determining which cells neighbour other cells (corner neighbouring cells exclusive) (Małecki, 2017).

The grid with randomly plotted sites takes as an input the number of sites it must plot and a bounding box to plot the sites inside. It randomly plots that number of sites within the bounding box. If a site is plotted on top of an existing site, then that site will be re-plotted. Each polygon in the randomly plotted grid will have at least one neighbour with no maximum number of neighbours (Figure 12).

The triangular grid consists of tessellating equilateral triangles of equal size. It uses the number of sites it must plot and a bounding box to plot the sites inside. It is one of the more complicated grid types to plot using a Voronoi diagram. The program iterated through possible different numbers of sites to place on each row until it finds the optimal number. Using these values, it was possible to find the horizontal distance between sites and therefore the required vertical height of triangles on each column using the properties of equilateral and right-angled triangles (Figure 12) Eq. (12):

$$y = \sqrt{3x^2} = Y_l + Y_s \tag{12}$$

where $Y_s > 0$ is the short vertical offset, $x > 0$ is the horizontal distance between sites, $y > 0$ is the height of the equilateral triangles and $Y_l > 0$ is the long vertical offset. The program continues iterating until it finds the smallest number of sites to place in a row, where once the entire grid is filled vertically there will still be approximately the same number of sites as originally stated. A short vertical offset is found to generate the triangular shape. The short vertical offset is calculated as the site for each equilateral triangle should be equidistant from all corners of the triangle. Therefore, using Pythagoras's theorem, the horizontal distance between sites and vertical height of the triangles, the location of these sites can be found Eq. (12) Eq. (13) Eq. (14):

$$Y_l = x^2 + Y_s^2 \tag{13}$$

$$Y_s = -\frac{x^2 - y^2}{2y} \tag{14}$$

where $Y_s > 0$ is the short vertical offset, $x > 0$ is the horizontal distance between sites, $y > 0$ is the height of the equilateral triangles and $Y_l > 0$ is the long vertical offset. In triangular tessellation on a singular row the triangles appear upright (generated from the short vertical offset) followed by an upside-down triangle in a repeating pattern. The upside-down triangles require the long vertical offset to be placed correctly. To find the long vertical offset the short vertical offset is taken away from the vertical height of the triangles (Figure 11) Eq. (12). Where both rows and columns are indexed even or odd, the short vertical offset is applied. While on odd indexed rows and even indexed columns, or even indexed rows and odd indexed columns, the long vertical offset is applied. This placement of sites creates a triangular gird. Due to rounding errors in Python and the

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

Foronoi Python library, the grid creates tiny edges between polygons where there should not be any. To fix this, if the length of an edge was under a certain threshold it was ignored and therefore correctly does not treat the two sites as neighbours (Figure 12).

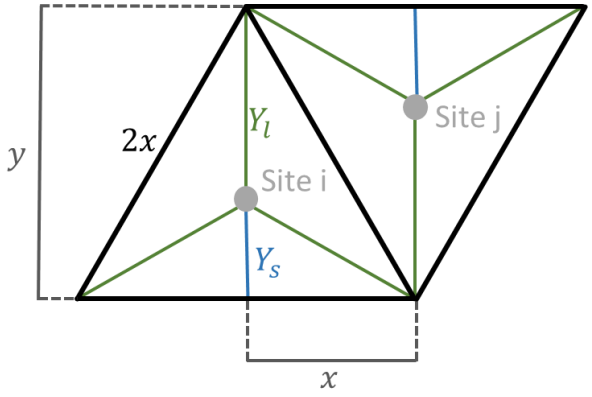


Figure 11: Calculation of Voronoi diagram site placement for equilateral triangular tessellation. Given the distances of x and y, Y_s and Y_l can be derived using the Pythagorean theorem.

The square grid consists of tessellating squares in a chess board pattern where the squares are all equal size and aligned both horizontally and vertically. It uses the number of sites it must plot and a bounding box to plot the sites inside. The floored square root of the number of sites to plot is found. This value is then used to equally space sites within the bounding box into equidistant columns and rows. Each polygon in the square grid (not on the edge) will have four neighbours (Figure 12).

The hexagonal grid consists of tessellating regular hexagons of equal size defined by the number of sites and a bounding box. Like the triangular grid, the program iterated through possible different numbers of sites to place on each column until it finds the optimal number. The optimal number of points to place was found by getting a number as close to what was originally stated within the bounding box, as the horizontal distance between sites could be found given the vertical distance Eq. (15):

$$x = \sqrt{y^2 - \left(\frac{y}{2}\right)^2} \tag{15}$$

where $x > 0$ is the horizontal distance between sites and $y > 0$ is the vertical distance between sites. Every even column is then offset vertically by half the vertical distance between sites. Each polygon in the hexagonal grid (not on the edge) will have six neighbours (Figure 12).

A random grid with increased focus on a region of interest was created. As stated previously this grid was called the flammable resolution grid (FRG). The FRG was designed to have a greater density of sites in regions most likely to be affected by fire. The regions that are most likely to be affected by the fire tend to be situated closer

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

to the ignition point of the fire and in areas of preselected important fuel types (e.g., urban, and forested areas). To do this the program has a base probability to select each pixel within a bounding box (this ensures not too many sites are generated). Once a pixel has been selected a probability is generated based on the distance from the ignition point and the fuel type of that pixel Eq. (16):

$$G = F(1+m)^{-k} \tag{16}$$

where $0 \leq G \leq 1$ is the probability of a site being placed, $0 \leq F \leq 1$ is the coefficient affecting the probability of site placement based on fuel, $m \geq 0$ is the distance between the current pixel and the mean ignition point, divided by the distance between the ignition point and furthest point within the bounding box and $k > 0$ is a coefficient that weighs site placement probability based on $(1+m)$. This allows most of the computing to be performed where it is most likely to be needed instead of wasting resources in areas the fire is unlikely to reach or to spread fast in. Each polygon in the FRG will have at least one neighbour with no maximum number of neighbours (Figure 13).

A fire was set at the coordinates EPSG:2157 (715122, 702388) at 0 seconds in the simulation. The environment had no wind, and the fire was simulated for 100,000 seconds with a timestep of 1,000 seconds. The fire was simulated 10 times for each grid type where the time it took for the IGS to simulate the fire was recorded (Table 1).

	Random Grid Time (s) (1998 sites)	**Triangular Grid Time (s) (1952 sites)**	**Square Grid Time (s) (1936 sites)**	**Hexagonal Grid Time (s) (1974 sites)**	**FRG Time (s)**
Fire 1	46.71	32.8	31.65	45	48.2
Fire 2	49.99	33.27	31.56	43.43	46.4
Fire 3	44.35	32.05	31.82	43.76	49.74
Fire 4	48.83	31.86	32.14	44.03	48.86
Fire 5	43.85	32.26	32.08	44.41	48.39
Fire 6	44.7	32.19	32.14	43.4	47.96
Fire 7	45.52	32.02	31.81	44.3	51.99
Fire 8	42.06	33.86	32.63	44.1	50
Fire 9	40.8	31.89	31.41	44.13	49.18
Fire 10	40.9	32.02	31.93	44.29	49.91
Mean	44.77	32.42	31.92	44.08	49.06
Standard Deviation	3.11	0.67	0.35	0.47	1.5
Standard Error	0.98	0.21	0.11	0.15	0.47

Table 1: Execution time for simulation on the different grid types. For both the random grids and FRG the number of sites is the mean from the 10 samples.

The regular grids tend to have quicker execution times than the random grid and FRG, where the FRG has the longest simulation time. The random grid and FRG also have significantly higher standard deviations and standard errors, this is due to both grids generating a completely new type of grid each time the simulation is run while the regular girds always have the exact same grid layout.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions
Open Access
EGU

The triangular, square, and hexagonal grids can roughly estimate the shape of the fire line, but they are confined to their regular shape and may struggle to represent some of the more irregular shapes of actual fire lines. As the Von Neumann neighbourhood was used to define neighbours. The regular grids all share the property of having an equal number of neighbours (not on the bounding box edges) that are all equidistant from each other which gives all polygons an equal spreading opportunity (Figure 12). The regular girds had an equal resolution across the entire grid. This leaves room for further optimisation where areas less likely to burn according to the model could have a lower resolution.

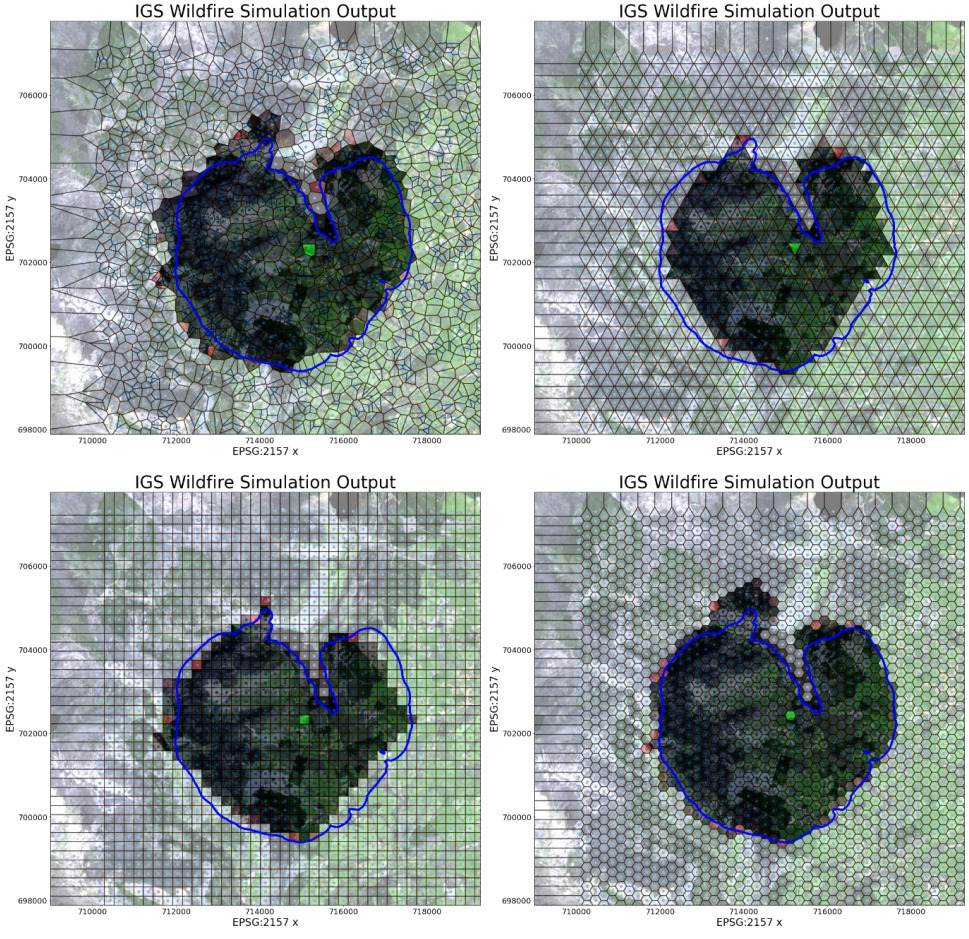


Figure 12: Outputs of different IGS grid types (coloured polygons) compared to ForeFire's output (blue). Top left: randomly plotted sites, top right: triangular grid, bottom left: square grid, bottom right: hexagonal grid (Copernicus 2022).

The randomised Voronoi grid can produce some of the more irregular shapes which can allow it to follow the irregular pattern of real terrain. Unlike the regular grids each polygon does not have an equal spreading opportunity due to the distance between neighbouring sites varying. The randomised Voronoi gird has a similar resolution throughout the grid which wasn't an optimal use of computation power. The FRG can follow the

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

shape of the fire line with greater precision due to its high resolution in areas of importance. It also decreases computational waste due to it rendering smaller numbers of polygons in areas of little importance. It does not have an equal spreading opportunity for each polygon and takes longer than all other types of grids to simulate the spread of fire (Figure 13).

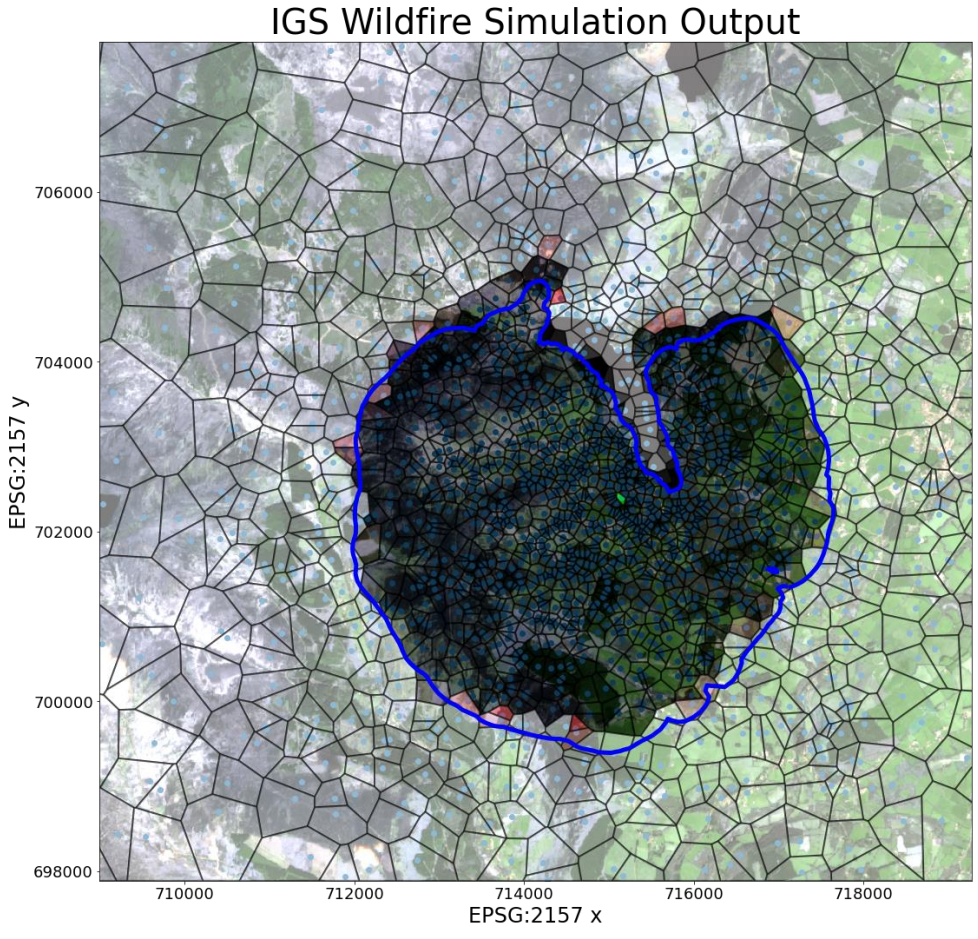


Figure 13: Output of the flammable resolution grid (coloured polygons) compared to ForeFire's output (blue) (Copernicus 2022).

Due to the greater resolution of the FRG, it will be used in the remaining comparisons in this paper. To find how effective IGS was at simulating wildfires it was then compared to the industry standard wildfire modelling program ForeFire.

6 Comparison of IGS and ForeFire

Outputs of ForeFire and the cubic spline of the IGS were graphed and then their similarity was found (Figure 14). To calculate how precise IGS was to ForeFire's output, a new method of comparing polygons was

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions
Open Access
EGU

developed. This method took inspiration from both set theory and receiver operating characteristics (ROC). When comparing the polygons produced by IGS and ForeFire, the area of predicted fire that overlapped was treated as true positives. Areas where IGS predicted fire and ForeFire didn't were false positives. Areas where ForeFire predicted fire and IGS didn't were false negatives and areas where both IGS and ForeFire predicted there wouldn't be fire were true negatives (Figure 15). Unfortunately, the area of true negatives could be considered infinite, dependant on the size of the plane where the fire is being simulated. Therefore, true negatives were not counted and any confusion matrix calculations involving them could not be found. This however still left a variety of confusion matrix derivations that could be found and used.

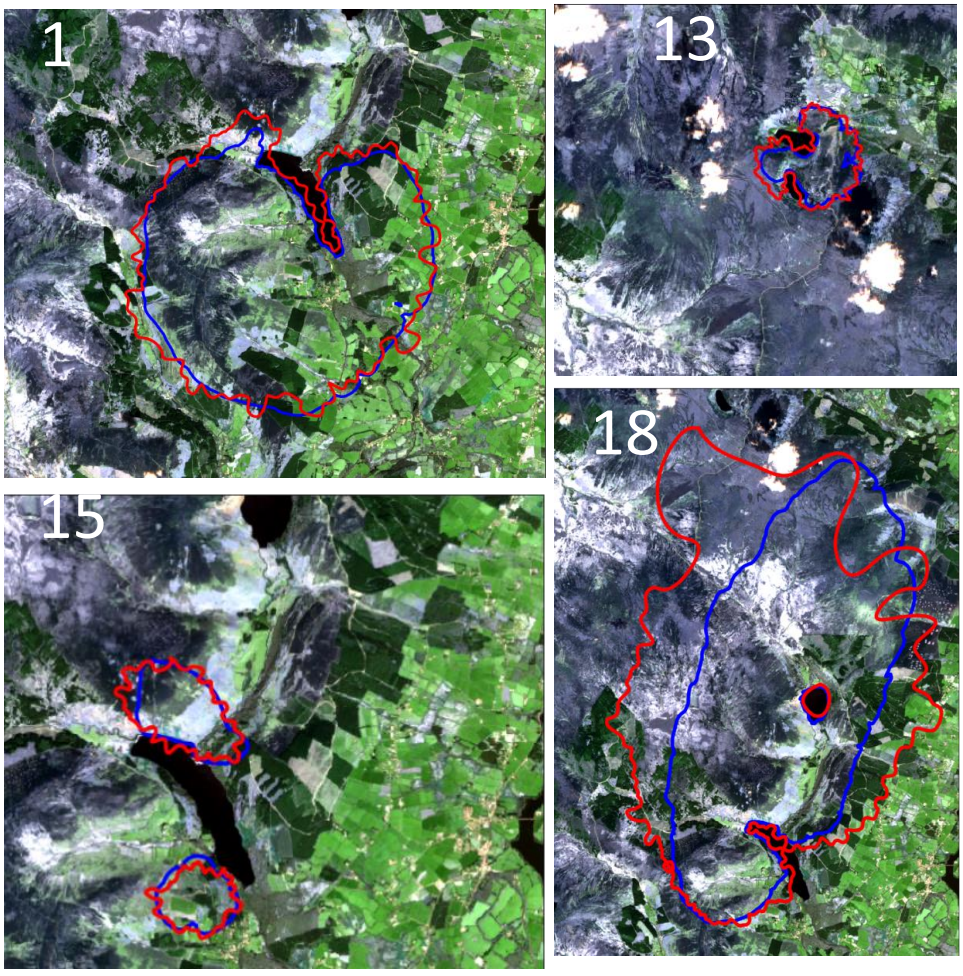


Figure 14: IGS outputs (Red) compared to ForeFire output (Blue). (Top left: Fire 1, Top Right: Fire 13, Bottom left: Fire 15 and Bottom Right: Fire 18 (Table 2)). (Copernicus 2022)

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

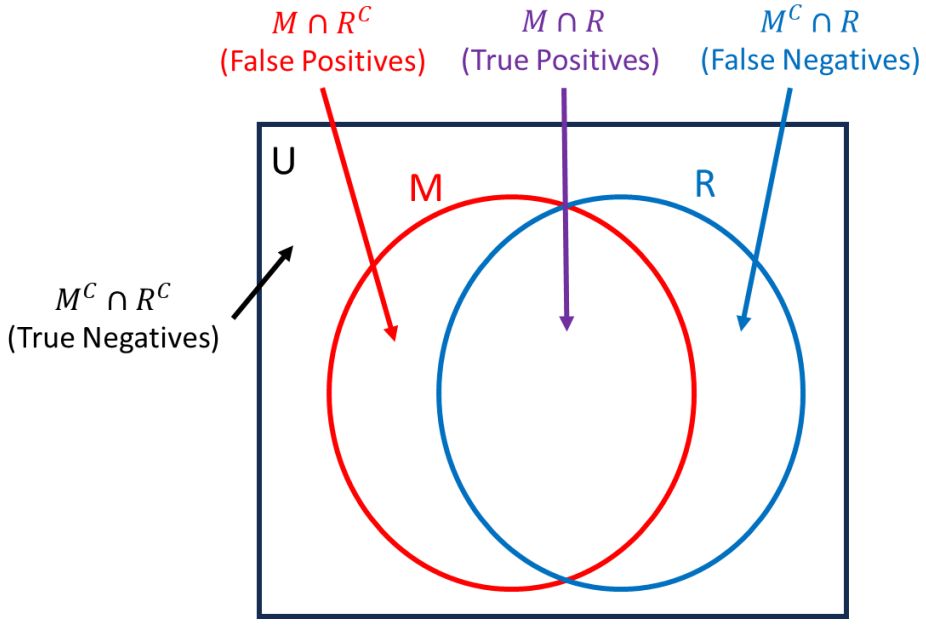


Figure 15: Calculating similarities of IGS (M) and ForeFire (R) wildfire shape outputs. Where the area of $M \cap R$ are the true positives, $M \cap R^C$ are the false positives, $M^C \cap R$ are the false negatives and $M^C \cap R^C$ are the true negatives.

Of the possible confusion matrix derivations available an altered form of the threat score was chosen as the measure of accuracy due to it measuring true positives, false positives, and false negatives simultaneously. The default threat score equation was used but two weighting factors were added to allow the scaling of how important false positives and false negatives are relative to the true positives Eq. (17):

$$TS = \frac{TP}{TP + w_1 FN + w_2 FP} \tag{17}$$

where TS is the threat score, TP is the area of true positives, FN is the area of false negatives, FP is the area of false positives, $w_1 \geq 0$ is the weighting coefficient for FN and $w_2 \geq 0$ is the weighting coefficient for FP.

If TS is 1 both the IGS an ForeFire polygons line up perfectly, while if TS is 0 then there is no overlap between the IGS and ForeFire polygons. Increasing w_1 or w_2 will make FN and FP respectively, have a greater impact on lowering TS while decreasing w_1 or w_2 will have the opposite effect. For the comparison of results in this paper we set $w_1 = w_2 = 1$.

While simulating these fires the time it took to run the simulation of the IGS usually took longer than ForeFire even though it required less computing. This was due to ForeFire being built in the more computationally efficient language C++ (Zehra et al., 2020). To make the comparison between the IGS and ForeFire fair, the

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024


Natural Hazards and Earth System Sciences
Discussions

main simulation loop of the IGS was ported to C++ from Python to provide an equal comparison. The remaining code for IGS was still in Python. The comparison doesn't include the setup times for ForeFire or the IGS.

7 Results

The 20 simulated wildfires were compared using both IGS and ForeFire. The wildfires were placed in different locations within the Wicklow Mountains, Ireland with varying wind speeds and simulated for different durations. Some simulations included multiple separate wildfires. The time it took to run the main simulation loops of ForeFire and IGS in both Python and C++ were all recorded and compared. The area of true positives, false positives, and false negatives for the IGS when compared to ForeFire were recorded for each fire and the corresponding threat score was also calculated. This data is presented in (Table 2).

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

Fire	Start Points and Time (x, y, t) (EPSG:2157, s)	Wind (Zonal, Meridian) (m/s)	Simulation Run Time (s)	ForeFire Time (s)	IGS Python Time (s)	IGS C++ Time (s)	ForeFire - IGS C++ Time (s)	ForeFire / IGS C++ Time (s)	True Positives (m^2)	False Positives (m^2)	False Negatives (m^2)	Threat Score
1	(715122, 702388, 0)	(0,0)	100,000	5.3	46	0.193	5.105	27	21,162,353	2,597,119	594,021	0.87
2	(713350, 704946, 0)	(0,0)	100,000	7.1	58	0.321	6.773	22	31,641,462	6,709,658	393,613	0.82
3	(715651, 706392, 0)	(0,0)	100,000	6.9	68	0.267	6.613	26	29,336,486	6,015,675	477,550	0.82
4	(710773, 707941, 0)	(0,0)	100,000	5.9	72	0.294	5.584	20	28,808,953	4,668,653	492,470	0.85
5	(713513, 708268, 0)	(0,0)	100,000	6.6	61	0.256	6.342	26	28,944,866	6,508,166	543,992	0.80
6	(712496, 708990, 0)	(0,0)	100,000	5.9	65	0.314	5.561	19	28,969,212	4,046,505	273,223	0.87
7	(715823, 708190, 0)	(0,0)	100,000	7.1	61	0.289	6.791	24	31,247,060	5,138,147	315,410	0.85
8	(713467, 711230, 0)	(0,0)	100,000	6.3	63	0.291	6.018	22	29,105,672	3,749,638	351,839	0.88
9	(714378, 701907, 0)	(0,0)	100,000	5.5	55	0.238	5.263	23	25,224,112	2,781,007	1,005,543	0.87
10	(704110, 703419, 0)	(0,0)	100,000	7.2	55	0.257	6.896	28	30,443,340	3,596,271	892,268	0.87
11	(709231, 705966, 0)	(0,0)	50,000	1.8	27	0.072	1.717	25	5,684,170	418,904	771,298	0.83
12	(714122, 706378, 0)	(0,0)	50,000	2.5	26	0.081	2.451	31	6,941,426	482,771	772,200	0.85
13	(714027, 716143, 0)	(0,0)	30,000	1.4	15	0.028	1.394	51	1,701,109	497,978	84,778	0.74
14	(713369, 707072, 0)	(0,0)	120,000	9.9	75	0.371	9.509	27	43,051,154	8,672,817	792,791	0.82
15	(715122, 702388, 0) (714868, 704531, 2000)	(0,0)	20,000	1.7	11	0.043	1.621	39	1,833,936	293,936	237,246	0.78
16	(715147.861932,702436.882089,0) (715930.304689,703048.530844,1200) (715803.717801,702092.322882,2700)	(0,0)	30,000	2.2	14	0.034	2.142	64	3,856,323	398,370	671,296	0.78
17	(714633, 704695, 0) (715297, 707714, 0)	(0,0)	50,000	3.7	27	0.106	3.609	35	13,747,087	3,459,420	326,943	0.78
18	(713945, 703907, 0)	(1,3)	50,000	10.6	31	0.158	10.404	67	42,742,800	25,938,603	3,724,274	0.59
19	(712649, 715686, 0)	(1, -3)	20,000	2.1	11	0.040	2.068	53	4,779,324	1,406,943	1,845,529	0.60
20	(713925, 707395, 0)	(-0.5, -0.4)	30,000	2.0	17	0.043	1.997	47	4,085,839	283,454	714,927	0.80
Mean				5.1	43	0.185	4.893	34	20,665,334	4,383,202	764,061	0.80
Standard Deviation				2.8	23	0.118	2.664	15	13,902,996	5,658,324	792,456	0.08
Standard Error				0.9	7	0.037	0.842	5	4,396,513	1,789,319	250,597	0.03
Sum									413,306,686	87,664,034	15,281,210	0.80

590

Table 2: Comparison of 20 simulated fires in ForeFire and IGS under the metrics of computing time (ForeFire/IGS C++ Time) and similarity of resulting fire line (Threat Score, where $w_1, w_2 = 1$).

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024


Natural Hazards and Earth System Sciences
Discussions

8 Conclusion

595 A novel software platform that allows the change of gird type using simple parameterisation for wildfire modelling was developed. This allowed grids to be compared within the same framework. The software included irregular grids constructed from Voronoi diagrams. The approach of using irregular girds to predict the spread of wildfires allows for extremely efficient computing while retaining a reasonable level of similarity in results defined by the threat score. Existing software such as ForeFire can produce more precise simulations, but
600 they require more computing time.

The software has a few issues that could be improved in future work. A better form of site placement than FRG could be developed. This is very evident in Fire 18 (Figure 14) where there is a large drop in site resolution at distances far away from the fire-starting location. The large distance between the IGS and ForeFire's northern
605 and western fire lines highlights this issue.

Due to the resolution of sites within the IGS the cubic spline produces a wavey pattern on the fire line while ForeFire has more straight edges. A higher resolution could be a possible solution for this issue.

610 Future work includes porting the rest of IGS from Python to a more computationally efficient programming language such as C++. This would speed up the setup process of the IGS model making it more efficient. Another method to determine site placement in the IGS would also rectify issues with lower resolutions on the outskirts of extremely large, simulated fires when using the FRG. The IGS could also be ran at a higher resolution to fix the problems with the fire line appearance produced by the cubic spline, this would however
615 affect performance.

Code availability: Code is freely available upon request.

Data availability: Data used in this study are freely available online from many sources which have been
620 referenced within the text.

Author contributions: Conor Hackett, Charles Markham and Rafael de Andrade Moral conceived and implemented the research. Gourav Misra and Tim McCarty provided the landcover map used by the software. All authors contributed to the editing and have read and agreed to the published version of the manuscript.

625

Competing interests: The authors declare no competing interests.

Acknowledgements

'This publication has emanated from research conducted with the financial support of Science Foundation
630 Ireland under Grant number 18/CRT/6049.'

References

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

Abdi, A. M.: Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data, GIsci Remote Sens, 57, 1–20, https://doi.org/10.1080/15481603.2019.1650447, 2020.

Al-Rawi, I.: Implementation of an Efficient Scan-Line Polygon Fill Algorithm , Computer Engineering and Intelligent Systems, 5, 22–28, 2014.

Andrews, P. L.: The Rothermel surface fire spread model and associated developments: A comprehensive explanation, https://doi.org/10.2737/RMRS-GTR-371, 2018.

Baetens, L., Desjardins, C., and Hagolle, O.: Validation of Copernicus Sentinel-2 Cloud Masks Obtained from MAJA, Sen2Cor, and FMask Processors Using Reference Cloud Masks Generated with a Supervised Active Learning Procedure, Remote Sens (Basel), 11, 433, https://doi.org/10.3390/rs11040433, 2019.

Blanchi, R., Leonard, J., Haynes, K., Opie, K., James, M., and Oliveira, F. D. de: Environmental circumstances surrounding bushfire fatalities in Australia 1901–2011, Environ Sci Policy, 37, 192–203, https://doi.org/10.1016/j.envsci.2013.09.013, 2014.

Chavez, P. S.: An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data, Remote Sens Environ, 24, 459–479, https://doi.org/10.1016/0034-4257(88)90019-3, 1988.

Copernicus Data Space Ecosystem: https://dataspace.copernicus.eu/, last access: 25 January 2024.

Home :: Corine Land Cover classes: https://land.copernicus.eu/content/corine-land-cover-nomenclature-guidelines/html/, last access: 25 January 2024.

Download - STEP: https://step.esa.int/main/download/, last access: 25 January 2024.

Filippi, J.-B., Bosseur, F., and Grandi, D.: ForeFire: open-source code for wildland fire spread models, in: Advances in forest fire research, Imprensa da Universidade de Coimbra, 275–282, https://doi.org/10.14195/978-989-26-0884-6_29, 2014.

Tools: https://github.com/forefireAPI/firefront/tree/master/tools, last access: 25 January 2024.

Fuels.ff Fuel Attribute Table: https://github.com/forefireAPI/firefront/blob/master/examples/aullene/fuels.ff, last access: 25 January 2024.

Fortune, S.: A sweepline algorithm for Voronoi diagrams, Algorithmica, 2, 153–174, https://doi.org/10.1007/BF01840357, 1987.

Gilmore, S., Nalband, A., and Dewan, A.: Effectiveness of DOS (Dark-Object Subtraction) method and water index techniques to map wetlands in a rapidly urbanising megacity with Landsat 8 data, CEUR Workshop Proc, 1323, 100–108, 2015.

Cubic spline for non-monotonic data (not a 1d function): https://stackoverflow.com/questions/67460967/cubic-spline-for-non-monotonic-data-not-a-1d-function, last access: 25 January 2024.

Green, M. E., Kaiser, K., and Shenton, N.: Modeling Wildfire Perimeter Evolution using Deep Neural Networks, 2020.

Hackett, C., Moral, R. D. A., and Markham, C.: Simulating Disease in Periods of Low Mobility Using a Hybrid Diffusion and Compartmental Model Built on Geographic Data, in: 2021 32nd Irish Signals and Systems Conference, ISSC 2021, https://doi.org/10.1109/ISSC52156.2021.9467871, 2021.

Haghani, M., Kuligowski, E., Rajabifard, A., and Kolden, C. A.: The state of wildfire and bushfire science: Temporal trends, research divisions and knowledge gaps, Saf Sci, 153, 105797, https://doi.org/10.1016/j.ssci.2022.105797, 2022.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Halofsky, J. E., Peterson, D. L., and Harvey, B. J.: Changing wildfire, changing forests: the effects of climate change on fire regimes and vegetation in the Pacific Northwest, USA, Fire Ecology, 16, 4, https://doi.org/10.1186/s42408-019-0062-8, 2020.

Hawthorne, D. and Mitchell, F. J. G.: Investigating patterns of wildfire in Ireland and their correlation with regional and global trends in fire history, Quaternary International, 488, 58–66, https://doi.org/10.1016/j.quaint.2017.06.067, 2018.

Helene, P., Britez, C., and Carvalho, M.: Fire impacts on concrete structures. A brief review, Revista ALCONPAT, 10, 1–21, https://doi.org/10.21041/ra.v10i1.421, 2019.

Jiao, Q., Fan, M., Tao, J., Wang, W., Liu, D., and Wang, P.: Forest Fire Patterns and Lightning-Caused Forest Fire Detection in Heilongjiang Province of China Using Satellite Data, Fire, 6, 166, https://doi.org/10.3390/fire6040166, 2023.

Jones, M. W., Santín, C., van der Werf, G. R., and Doerr, S. H.: Global fire emissions buffered by the production of pyrogenic carbon, Nat Geosci, 12, 742–747, https://doi.org/10.1038/s41561-019-0403-x, 2019.

Keeley, J. E. and Syphard, A. D.: Large California wildfires: 2020 fires in historical context, Fire Ecology, 17, 22, https://doi.org/10.1186/s42408-021-00110-7, 2021.

Lin, Z., Liu, H. H. T., and Wotton, M.: Kalman Filter-Based Large-Scale Wildfire Monitoring With a System of UAVs, IEEE Transactions on Industrial Electronics, 66, 606–615, https://doi.org/10.1109/TIE.2018.2823658, 2019.

Małecki, K.: Graph Cellular Automata with Relation-Based Neighbourhoods of Cells for Complex Systems Modelling: A Case of Traffic Simulation, Symmetry (Basel), 9, 322, https://doi.org/10.3390/sym9120322, 2017.

McElwain, L. and Sweeney, J.: Climate change in Ireland- recent trends in temperature and precipitation, Irish Geography, 36, 97–111, https://doi.org/10.1080/00750770309555815, 2003.

Meier, S., Elliott, R. J. R., and Strobl, E.: The regional economic impact of wildfires: Evidence from Southern Europe, J Environ Econ Manage, 118, 102787, https://doi.org/10.1016/j.jeem.2023.102787, 2023.

Pais, C., Carrasco, J., Martell, D. L., Weintraub, A., and Woodruff, D. L.: Cell2Fire: A Cell-Based Forest Fire Growth Model to Support Strategic Landscape Management Planning, Frontiers in Forests and Global Change, 4, https://doi.org/10.3389/ffgc.2021.692706, 2021.

Park, H., Nam, K., and Lim, H.: Is critical infrastructure safe from wildfires? A case study of wildland-industrial and -urban interface areas in South Korea, International Journal of Disaster Risk Reduction, 95, 103849, https://doi.org/10.1016/j.ijdrr.2023.103849, 2023.

Penney, G., Habibi, D., and Cattani, M.: Firefighter tenability and its influence on wildfire suppression, Fire Saf J, 106, 38–51, https://doi.org/10.1016/j.firesaf.2019.03.012, 2019.

Pereira, J., Mendes, J., Júnior, J. S. S., Viegas, C., and Paulo, J. R.: A Review of Genetic Algorithm Approaches for Wildfire Spread Prediction Calibration, Mathematics, 10, 300, https://doi.org/10.3390/math10030300, 2022.

Prat-Guitart, N., Nugent, C., Mullen, E., Mitchell, F. J. G., Hawthorne, D., Belcher, C. M., and Yearsley, J. M.: Peat Fires in Ireland, in: Coal and Peat Fires: A Global Perspective, Elsevier, 451–482, https://doi.org/10.1016/B978-0-12-849885-9.00020-2, 2019.

https://doi.org/10.5194/nhess-2024-27
Preprint. Discussion started: 19 February 2024

Natural Hazards and Earth System Sciences
Discussions

Pringle, M. J., Schmidt, M., and Tindall, D. R.: Multi-decade, multi-sensor time-series modelling—based on geostatistical concepts—to predict broad groups of crops, Remote Sens Environ, 216, 183–200, https://doi.org/10.1016/j.rse.2018.06.046, 2018.

QGIS: https://qgis.org/en/site/, last access: 25 January 2024.

dos Reis, M., Graça, P. M. L. de A., Yanai, A. M., Ramos, C. J. P., and Fearnside, P. M.: Forest fires and deforestation in the central Amazon: Effects of landscape and climate on spatial and temporal dynamics, J Environ Manage, 288, 112310, https://doi.org/10.1016/j.jenvman.2021.112310, 2021.

Rothermel, R. C.: A mathematical model for predicting fire spread in wildland fuels, Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Intermountain Forest and Range Experiment Station. 40 p., 1972.

San Martin, D. and Torres, C.: 2D Simplified Wildfire Spreading Model in Python: From NumPy to CuPy, CLEI Electronic Journal, 26, https://doi.org/10.19153/cleiej.26.1.5, 2023.

Interpolation (scipy.interpolate): https://docs.scipy.org/doc/scipy/reference/interpolate.html, last access: 25 January 2024.

EO Browser: https://www.sentinel-hub.com/explore/eobrowser/, last access: 25 January 2024.

SFI-Defence Organisation Innovation Challenge, Challenge 1: https://www.sfi.ie/funding/funding-calls/future-innovator-defence/#row-collapse-item-1, last access: 25 January 2024.

Sibanda, S. and Ahmed, F.: Modelling historic and future land use/land cover changes and their impact on wetland area in Shashe sub-catchment, Zimbabwe, Model Earth Syst Environ, 7, 57–70, https://doi.org/10.1007/s40808-020-00963-y, 2021.

How to use the SNAP API from Python: https://senbox.atlassian.net/wiki/spaces/SNAP/pages/19300362/How+to+use+the+SNAP+API+from+Python, last access: 25 January 2024.

Sullivan, A., Baker, E., and Kurvits, T. (Eds.): Spreading like Wildfire: The Rising Threat of Extraordinary Landscape Fires, 2022.

Network Common Data Form (NetCDF): https://www.unidata.ucar.edu/software/netcdf/, last access: 25 January 2024.

Windy: https://windy.app/, last access: 25 January 2024.

Xue, C., Krysztofiak, G., Ren, Y., Cai, M., Mercier, P., Fur, F. Le, Robin, C., Grosselin, B., Daële, V., McGillen, M. R., Mu, Y., Catoire, V., and Mellouki, A.: A study on wildfire impacts on greenhouse gas emissions and regional air quality in South of Orléans, France, Journal of Environmental Sciences, 135, 521–533, https://doi.org/10.1016/j.jes.2022.08.032, 2024.

foronoi: https://github.com/Yatoom/foronoi, last access: 25 January 2024.

Zehra, F., Javed, M., Khan, D., and Pasha, M.: Comparative Analysis of C++ and Python in Terms of Memory and Time, Preprints (Basel), 2020.

Zhang, S., Liu, J., Gao, H., Chen, X., Li, X., and Hua, J.: Study on Forest Fire spread Model of Multi-dimensional Cellular Automata based on Rothermel Speed Formula, CERNE, 27, https://doi.org/10.1590/01047760202127012932, 2021.