Title

Fire Spread Simulation and Probabilistic Regional Fire Loss Assessment at Wildland-Urban Interface

Permalink

https://escholarship.org/uc/item/4s73g79d

Author

Zhou, Xitong

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Fire Spread Simulation and Probabilistic Regional Fire Loss Assessment at Wildland-Urban Interface

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Civil Engineering

by

Xitong Zhou

2024

ABSTRACT OF THE DISSERTATION

Fire Spread Simulation and Probabilistic Regional Fire Loss Assessment at Wildland-Urban Interface

by

Xitong Zhou
Doctor of Philosophy in Civil Engineering
University of California, Los Angeles, 2024
Professor Jian Zhang, Chair

Fire spread simulation plays a significant role in mitigating the devastating impacts of fires in both wildland and urban areas. Over the years, the analogy of fire spread simulation has undergone a remarkable transformation from wildland area to urban area. Especially the structure damage assessment after the fire becomes more important these days. This research provides a fire spread model that is designed to simulate the affected area and assess the fire damage to concrete, steel, and wooden structures in both wildland and urban areas. There are three main parts in this research, the fire spread simulation, structure damage assessment, and the regional cost estimation.

First, the fire spread simulation integrates Rothermel's surface fire spread model, fire elliptical growth model, and Huygens' principle. Rothermel's surface fire spread model takes into account factors such as weather conditions, terrain, and fuel types to predict the fire spread rate. The elliptical growth model combines not only a mathematical approach but also empirical

results that come from past research. Finally, Huygens' Principle of Fire Front Expansion is a pivotal concept in fire growth modeling, particularly in sophisticated vector or wave-type models. In addition, this model accounts for the impact of firebrands, which can lead to spot fires and are more commonly found in urban areas.

Second, following the fire spread simulation, the nodes associated with a specific type of structure are equipped with time data that indicates the duration for which they have been exposed to the fire. Through the analysis of compartment time and temperature data, the temperature condition within the structure can be derived. Furthermore, a probabilistic model and the corresponding fragility curve have been developed to assist in the assessment of damage levels.

Third, Monte Carlo simulation is adapted to account for the simplicity and uncertainty in environmental conditions while estimating the probabilistic economic losses within the affected area. This methodology aids in pinpointing areas of potential vulnerability and determining the likelihood of structure ignition. The assessment of economic loss consists of two parts, the damage to wildlands, and the damage to structures. The latter is further divided into the damage to structural and non-structural components. The aggregate repair costs for all nodes will represent the total economic loss in the area.

In conclusion, this study developed a fire spread simulation model for the wildland-urban interface, incorporating inhomogeneous fuels and firebrand impacts, and created probabilistic models for structural loss assessment using fragility functions and Monte Carlo simulations. The model's accuracy and reliability were demonstrated through a case study, where the simulated fire spread and predicted regional losses closely matched real-world estimates. This model is essential for urban planners, civil engineers, and fire management professionals, providing a valuable tool to better predict, mitigate, and manage the adverse effects of fires.

The dissertation of Xitong Zhou is approved.

Ertugrul Taciroglu

Henry J. Burton

Sriram Narasimhan

Jian Zhang, Committee Chair

University of California, Los Angeles

2024

To my parents, Jun Zhou and Guifen Zhuang

my twin brother, Xixie Zhou

my fiancee, Wenyan Zhao

for their love, support, and patience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I extend my deepest gratitude to my thesis advisor, Jian Zhang, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate her willingness to give her time so generously and offer me valuable insights into the research process.

I am immensely grateful to the members of my thesis committee, Ertugrul Taciroglu, Henry Burton, and Sriram Narasimhan, for their splendid education on each subject and their critical insights and feedback which were instrumental in refining my research methodologies and ultimately made this work much more robust.

My sincere thanks also go out to the CEE Department at UCLA, for providing me with the necessary financial support and resources throughout my research. Special thanks to Mimi Baik and Stacey Fong, whose assistance was invaluable. My appreciation also goes to my research group member, Dr. Dong Wang, and Dr. Yi Peng who have offered support and help to me in past years.

I am also thankful to my family for their wisdom, encouragement, and unconditional support. I am especially grateful to my parents, Jun Zhou, Guifen Zhuang, and my twin brother, Xixie Zhou whose love and guidance are with me in whatever I pursue. A heartfelt thank you to my fiancee, Wenyan Zhao, for her unwavering support, patience, and encouragement throughout this challenging journey. Your love and understanding have been a constant source of strength and comfort to me.

This journey would not have been possible without the collective support and encouragement of each one of you mentioned above, and many others unnamed. Thank you.

VITA

Education

M.S. in Structural and Earthquake Engineering, University of California, Los Angeles, Los Angeles, U.S. 2018-2019

B.E. in Civil Engineering, Purdue University, West Lafayette, Indiana, U.S. 2014-2018

ACADEMIC EXPERIENCE

Teaching Fellow, University of California, Los Angeles 2019-2024

Graduate Student Researcher 2018-2024

PUBLICATIONS

Zhou, XT, Zhang J. Fire Spread Simulation and Probabilistic Regional Fire Loss Assessment at Wildland Urban Interface. 2024, in preparation

1. INTRODUCTION

1.1 Motivation

Before delving into research regarding the fires in urban areas, the primary focus of fire research was centered on wildfires. The exploration of wildfire research took root in the 1970s, driven by an increasing recognition of their significance (Alexander and Cruz, 2013). Uncontrolled wildfires can have far-reaching impacts on local ecosystems, climate patterns, air quality due to smoke, and in some cases, it may take decades for the affected environment to fully recover. (Maselli et al., 2000; Alloza et al., 2006; Fernandez et al., 2007). In recent years, wildfires occurring at the Wildland-Urban Interface (WUI), where these fires encroach upon communities, have garnered significant attention.

One distinguishing factor between wildfires and WUI fires is the consistent assumption of fuel type in the former. For wildfires, soil properties, environmental factors, and geological conditions are often treated as relatively uniform across the area just ahead of the fire line. However, when it comes to WUI fires, a critical question arises: how does fire transition from a wildland area to an urban one? Typically, there exists a physical gap between the wildland and urban zones, and research indicates that embers or firebrands are the primary sources for transporting fire into urban areas (Gollner et al., 2015).

Simulating fires in urban areas presents a much more complex challenge compared to wildfires. Despite the initiation of wildfire research in the 1970s, achieving high accuracy in these simulations remains an ongoing pursuit. Uncertainty remains a pervasive obstacle in the quest for a more accurate simulation. Consequently, the initial step involves comprehending, characterizing, and quantifying the influence of environmental uncertainty.

Figure 1.1 provides visual representations of what wildfires and WUI fires typically look like. Satellite images offer a clear depiction of the fire front and the areas both currently affected and under threat. Additionally, by analyzing successive images, it becomes possible to estimate the extent of the affected area, the prevailing wind direction, and the rate of fire spread.

Figure 1.1 - California wildfires: Flames seen from space. (from BBC News, 2017)

Numerous instances demonstrate that wildfires can lead to not only environmental issues but also catastrophic consequences for humanity, including their communities, and substantial economic losses. For example, Figure 1.2 shows the Camp Fire (2018) that happened in California, this was considered as the deadliest and most destructive wildfire in California history. It destroyed the town of Paradise and led to 85 fatalities. Approximately 18,800 structures were destroyed and about 153336 acres of the area was covered. The economic losses were estimated at over $16 billion (Cal Fire, retrieved on 6/1/2024). Fort McMurray Fire (2016) in Canada, this wildfire resulted in the evacuation of nearly 90,000 residents from Fort McMurray in Alberta. It destroyed approximately 2,400 homes and buildings, with an estimated economic impact of

CAD 9 billion (Public Safety Canada, 2016). These examples underline the devastating impact wildland fires can have on communities, economies, and ecosystems.

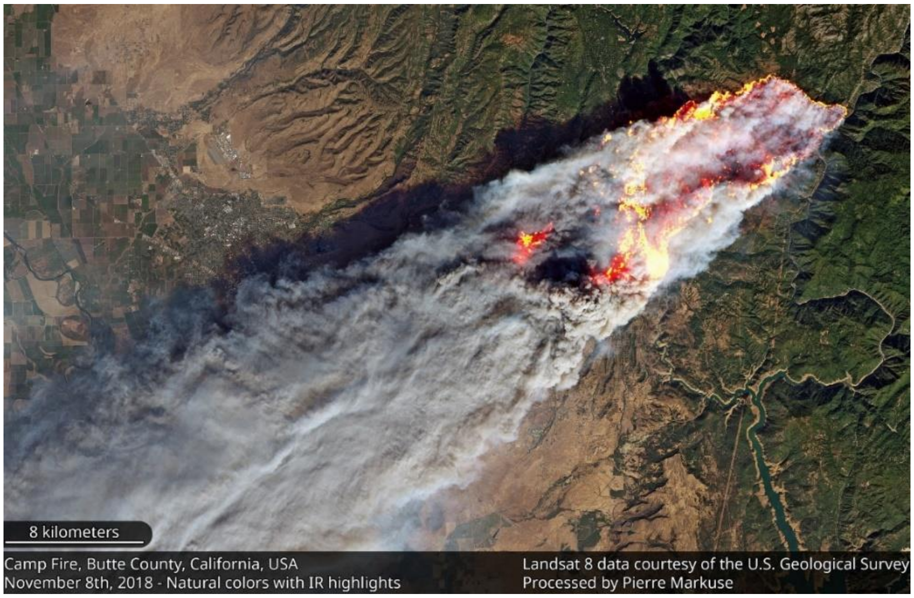


Figure 1.2 – Satellite Image of Camp Fire. (from Cal Fire, 2018)

As a result, fire simulation research has become an indispensable tool in understanding fire dynamics, predicting fire spread, and mitigating potential risks. in wildland areas, fire spread simulation is essential for predicting and managing wildfires. With the increasing frequency and severity of wildfires driven by climate change, accurate simulation models are vital for early warning systems, evacuation planning, and resource allocation. By providing timely information to firefighters and decision-makers, these simulations enhance the effectiveness of fire suppression efforts, protect ecosystems, and save lives. On the other hand, in urban areas, fire spread simulation is equally critical for assessing the vulnerability of structures and improving urban planning. Urban fires pose unique challenges due to densely populated areas and complex infrastructure networks (Mahmoud & Chulahwat, 2018; Alexander & Cruz, 2013). Fire spread simulations help urban planners and emergency responders understand how fires can propagate within the built environment, guiding the placement of firebreaks, evacuation routes, and the design of fire-resistant buildings. Moreover, these simulations could aid in evaluating the effectiveness of building codes and firefighting strategies, ultimately enhancing community resilience and safety. Furthermore, the integration of technology, including Geographic Information Systems (GIS) and advanced modeling techniques, has revolutionized fire spread simulations, making them more accurate and accessible. These simulations could also empower decision-makers with real-time data, allowing for proactive and informed responses to fire incidents. In addition, they facilitate the development of innovative strategies for fire prevention, including controlled burns and vegetation management in wildland areas, and the adoption of fire-resistant construction materials and urban planning practices in cities.

1.2 Background

Many simulation models were developed after the introduction of Rothermel’s surface fire spread model, which is the most commonly used in the U.S. fire management systems and also a significant use outside the U.S. There are two primary types of fire spread simulation models: vector-based models, such as FARISITE, which discretize the fire front and calculate the next position for each point, and raster models, like cellular automata, which discretize the terrain into cells and calculate the spread rate based on these cells. The latter approach resembles a raster model and provides insights into the extent to which each cell has ignited (Finney, 2004; Mahmoud & Chulahwat, 2016).

It's important to note that there is no perfect model for simulating fire spread, primarily due to the necessity of making significant assumptions and dealing with inherent inaccuracies. These assumptions are required to simplify the complex relationships and account for minor factors related to fuel properties, terrain, fuel type consistency, and weather conditions.

Numerous software programs and systems have been developed to facilitate the simulation of fire spread (Sousa, et all., 2012). One such example is CudaFGM, created by LASEF-IST, which leverages GPU technology and the CUDA framework developed by NVIDIA. CudaFGM employs cellular automata to simulate fire spread, enabling the determination of the rate of spread in the absence of wind and terrain effects, as well as accounting for these factors through the use of propagation ellipses (Sousa, et all., 2012).

Nonetheless, the primary challenge faced by this simulation model lies in the availability and quality of input data, such as terrain information, the accuracy of the theoretical foundation for the fire behavior model, and the fire growth algorithm itself (Kochanski, et al, 2013). An alternative approach is the utilization of WRF-Sfire, a coupled atmosphere-fire model developed by the Open Wild Fire Modeling Community. WRF-Sfire excels in providing a realistic assessment of the rate of fire front spread and its associated vertical temperature profiles. However, this model does come with certain limitations. It tends to overestimate vertical velocities while underestimating horizontal wind speeds at altitudes exceeding 10 meters. Moreover, it does not account for fire-induced horizontal winds, fire-induced updrafts, or the spatial intricacies of fire-induced airflow (Kochanski, et al, 2013). On the other hand, in 2023, WRF-Fire was integrated with the fire-atmosphere wildland fire simulation platform to simulate the 2018 Camp Fire. The results revealed significant discrepancies between the simulated fire and the real-world event. These limitations are primarily due to the complexity of the terrain and uncertainties in wildfire modeling (Shamsaei, et al., 2023).

FARSITE stands out as the most widely adopted fire growth simulation modeling system in the United States. It encompasses a comprehensive suite of five distinct models, which include surface fire spread model (Rothermel, 1972), crown fire ignition model (Wagner, 1977), spotting model (Albini, 1979), crown fire spread model (Rothermel, 1991), and dead fuel moisture model (Nelson, 2000). To generate simulations, FARSITE necessitates a minimum of five raster layers, encompassing elevation, aspect, slope, fuel model, and canopy cover. Its notable advantage lies in its integration of spatial information on topography, fuels, weather conditions, and wind patterns (Finney, 2004).

However, there is still room for improvement in FARSITE's fuel model to enhance simulation accuracy. One of the primary challenges associated with the FARSITE model is its assumption of a homogenous fuel load in front of the firefront (Finney, 2004). This assumption can lead to reduced efficiency and accuracy, particularly when dealing with smaller, more complex areas. On the other hand, FARSITE primarily concentrates on modeling fire spread in wildland areas. However, it operates under the assumption of simplified fuel conditions, which may not fully encompass the complexities of real-world fire behavior, particularly in urban environments (Finney, 2004). These limitations underscore the need for continued refinement and improvement in fire spread modeling to better account for real-world variations and provide more informative output.

In summary, every fire spread model has its limitations to varying degrees, as they all make simplifications to achieve their specific objectives. These simplifications often result in the neglect of minor factors, and the models are typically based on a single set of idealized conditions. For instance, many models assume that the fuel complex is continuous, uniform, and homogeneous, even though it is more intricate in reality. Some models also simplify the fuel bed by assuming it is a single layer contiguous to the ground, whereas the actual situation is more complex (Andrew, 2018). Additionally, the potential for fire spread through spotting, such as flying embers or firebrands, is often not adequately accounted for in terms of ignition possibilities. These inherent limitations highlight the ongoing need for refinement and development in fire spread modeling to better align with the complexities of real-world fire behavior. The model developed in this dissertation aims to provide a feasible approach to simulate fire spread not only in wildlands but also in urban areas.

On the other hand, there is also a lack of loss assessment for wildfires. In 2022, Khorasani proposed a probabilistic risk and loss assessment framework for wildfires (Khorasani, et al., 2022). This framework depicts dynamic risk through spatial probability density functions of loss, including physical, social, economic, environmental, and health impacts. However, limitations remain due to the complexity and large scope of the problem domain. There are no integrated solutions that account for all aspects (Khorasani, et al., 2022). This dissertation will focus on the economic loss resulting from structural and wildland damage. An assessment of structural damage and the development of a probabilistic regional cost estimation post-fire will be demonstrated. This dissertation initially combines fire spread simulation with post-fire consequences, thereby providing a deeper understanding of damage assessment for fire hazards.

1.3 Organization

The objective of this dissertation is to develop a fire spread simulation in the wildland-urban interface, create probabilistic strength loss models to assess structural damage and estimate regional losses after a fire. To achieve this, this dissertation will begin with the basic theories and methods for fire spread simulation.

Chapter 2 focuses on the main theory and methodology for fire spread. This study utilizes Rothermel’s surface fire spread model to calculate the rate of fire spread at each node. Section 2.1 discusses the main input parameters and the correlations of Rothermel’s model. Additionally, fuel models developed after Rothermel's innovation will be examined. Section 2.2 covers the methodology used for firefront propagation, including the elliptical growth model, the definition of ellipse dimensions, Huygen's principle, and the integration of the fire elliptical growth model with Huygen’s Principle in a grid-based simulation. Finally, this dissertation will discuss the implementation of the Monte Carlo Simulation used in this study.

Chapter 3 discusses fire spread in urban areas and the behavior of structures under fire conditions. Section 3.1 covers firebrand transportation in urban areas and the implementation of the firebrand model in this study. Section 3.2 examines the behavior of structures under fire, including the investigation of the time-temperature relationship for structural components and the relationship between strength loss and temperature. Section 3.3 details the development of probabilistic models and fragility curves for structural components.

Chapter 4 focuses on the development of the Matlab model. It begins with the input parameters from Excel and then discusses the model structure, including its definition, simulation process, and post-analysis for cost estimation. At the end of Chapter 4, this dissertation will also address the capabilities of the model, its limitations, and suggestions for future work.

Chapter 5 discusses parametric modeling and presents a case study of the Lahaina fire that occurred in August 2023 in Hawaii. The parametric modeling section will show the correlation between the spread area and the variation of input parameters. The case study will begin by discussing the selection of the main variables for this fire scenario. It will then cover the fire spread simulation, structural damage assessment under this scenario, and loss assessment. Finally, this model allows the user to manually add fire barriers and observe how these barriers mitigate the fire spread.

Chapter 6 presents the conclusion and future work. This will include improvements to the model itself and the broadening of the research scope.

2. THEORY AND SPREAD MODEL BASIS FOR FIRE SPREAD SIMULATION

2.1 Rothermel's Surface Fire Spread Model

Rothermel's Surface Fire Spread Model has been regarded as a cornerstone in wildfire science and management (Rothermel, 1972). It has been used to estimate the fire spread rate since 1972. Most fire spread simulation software is created based on Rothermel’s model like FARSITE (Finney, 2004). Sullivan (2009a,b,c) conducted an extensive review of wildland surface fire spread models developed between 1990 and 2007. His findings addressed that there are six types of models including physical and quasi-physical models, empirical and quasi-empirical models, as well as simulation and mathematical analogue models.

In his research, Sullivan provided definitions for each model category. He characterized physical models as those relying on a physical framework as the foundation for representing both the physics and chemistry governing fire spread. In contrast, quasi-physical models primarily concentrate on the physical aspects. Empirical models, on the other hand, lack a physical foundation and tend to be inherently statistical in nature. Quasi-empirical models incorporate some form of a physical framework as the basis for their statistical modeling. Mathematical analog models employ mathematical principles instead of physical equations to replicate the dynamics of wildland fire spread (Sullivan, 2009a). Additionally, simulation models were distinguished as those models that integrate fire behavior principles into landscape spread applications. The central focus of empirical models revolves around the correlation between variables such as wind speed and fuel moisture content with the rate of forward spread.

Many simulation models involve the implementations of pre-existing empirical or quasi-empirical models. Their primary purpose lies in the conversion of these generally one-dimensional models into two dimensions, enabling the simulation of fire perimeter progression within a modelled landscape. Mathematical analogue models, on the other hand, are built upon mathematical concepts, rather than attempting to physically represent the spread of fire. Remarkably, they still manage to replicate the dynamics of fire propagation (Sullivan, 2009b; Sullivan, 2009c). Rothermel’s model is categorized as quasi-empirical as it is based on a physical basis, heat balance model developed by Frandsen (1971). It's noteworthy that this model appealed to be the primary spread model for nine of the 14 simulation models discussed in Sullivan’s work (Sullivan, 2009c). Rothermel’s fire spread model has can play a significant role in the prediction and understanding of the behavior of surface fires in wildland and forested environments. The model itself has significant assumptions and limitation, however, many researchers and systems revised it and made it usable. It considers the variabilities from weather, fuel, and many other aspects. This theory makes the prediction of fire spread simpler and makes it possible to calculate (Andrews, 2018). In conclusion, it uses the fuel property, weather, and terrain property prior to ignition to estimate the following spread rate. Equation 1.1 shows the final equation for the spread rate using Rothermel’s model.

$$R = \frac{I_R \xi (1 + \phi_w + \phi_s)}{\rho_b \varepsilon Q_{ig}} \tag{1.1}$$

This fire spread model is based on the conservation of energy principle and is calculated by a heat source divided by a heat sink. The numerator is the heat source, which is also considered to be the propagating flux. The denominator is the heat sink, which is the heat required to ignite the fuel (Frandsen, 1971). Table 2.1 describes the components of the final fire spread equation.

Table 2.1 -Definition of components in the final equation for Rothermel's surface fire spread model. (from Andrews, 2018)

Term	**Name**	**Explanation**
I_R	Reaction intensity (Btu/ft^2/min)	Energy release rate perunit areaof firefront
x	Propagating flux ratio	Proportion of the reaction intensity that heats adjacent fuelparticles to ignition (no wind)
ϕ_w	wind factor	Dimensionless multiplierthat accounts for the effect of wind inincreasing the propagating fluxratio
ϕ_s	Slope factor	Dimensionless multiplierthat accounts for the effect of slope in increasing the propagating flux ratio
ρ_b	Bulk density (lb/ft^3)	Amount of oven-dry fuel per cubic foot of fuel bed
e	Effective heating number	Proportion of a fuel particle that is heated to igniton temperature at the time flaming combusion starts
Q_{ig}	Heat of preignition (Btu/lb)	Amount of heat required to ignite one pound of fuel
$I_R\xi$	No-wind, no-slope propagating flux (Btu/ft^2/min)	Heat releaseratefrom a fire to the fuel ahead of the fire, without wind or slope
$(1+\phi_w+\phi_s)$	wind and slope factor multiplier	increase to the no-wind, no-slope propagating flux dueto windand slope
$I_R\xi(1+\phi_w+\phi_s)$	Heat source (Btu/ft^2/min)	Propagating flux
$\rho_b\varepsilon Q_{ig}$	Heat sink (Btu/ft^3)	Heat required to ignite the fuel
$\frac{I_R\xi}{\rho_b\varepsilon Q_{ig}}$	No-wind, no-slope rate ofspread (ft/min)	Rate of spread without wind and slope factors
R	Rate of spread	Speed of firefront of a surface fire

2.1.1 Input parameters

The input parameters for this quasi-empirical model can be grouped into three categories including fuel particle properties, fuel array arrangements, and environmental values (Rothermel, 1972). Table 2.2 provides the details and explanation for each category. Table 2.3 gives all the equations in the basic fire spread model and also includes modifications by Albini (1976).

Table 2.2 - Input parameters for the basic fire spread model. (from Andrews, 2018)

Type	Symbol	Parameter	Notes
Fuel Particle	h	Low heat content (Btu/lb)	Often 8000 Btu/lb
	S_T	Total mineralcontent(Fraction)	Generally 0.0555 lb mineral/lb for wood
	S_e	Effective mineral content (Fraction)	Generally 0.01 (lb minerals - lb silica)/lb for wood
	ρ_p	Oven-dry particle density (lb/ft^3)	Generally 32 lb/ft^3
Fuel array	σ	Surface-area-to volumeratio (ft^2/ft^3)	
	w_0	Oven-dry fuel load (lb/ft^2)	
	δ	Fuel bed depth (ft)	Mean fuel array value
	M_x	Dead fuel moisture of extinction (fraction)	Live fuel is not included in the basic model. Live fuel moisture of extinction is calculated in the final model
Environmental	M_f	Moisture content (fraction)	Dry weight basis lb moisture / lb wood
	U	Wind velocity at midflame height (ft/min)	
	$\tan\phi$	Slope steepness, maximum (fraction)	Vertical rise / horizontal distance

Table 2.3 - Equations for the basic fire spread model with modified terms. (from Andrews, 2018)

Element	**Equation**
Rate of spread (ft/min)	$R = \dfrac{I_R \xi (1 + \phi_w + \phi_s)}{\rho_b \varepsilon Q_{ig}}$
Reaction intensity (Btu/ft^2-min)	$I_R = \Gamma' w_n h \eta_M \eta_s$
Optimum reaction velocity (min^-1)	$\Gamma' = \Gamma'_{\max} (\beta / \beta_{op})^A \exp[A(1 - \beta / \beta_{op})]$
	$A = 133\sigma^{-0.7913}$
Maximum reaction velocity (min^-1)	$\Gamma'_{\max} = \sigma^{1.5}(495 + 0.0594\sigma^{1.5})^{-1}$
Optimum packing ratio	$\beta_{op} = 3.348\sigma^{-0.8189}$
Packing ratio	$\beta = \rho_b / \rho_p$
Oven-dry bulk density (lb/ft^3)	$\rho_b = w_0 / \delta$
Net fuel load (lb/ft^2)	$w_n = w_0(1 - S_T)$
Moisture damping coefficient	$\eta_M = 1 - 2.59 r_M + 5.11(r_M)^2 - 3.52(r_M)^3$
	$r_M = M_f / M_x (\max = 1.0)$
Mineral damping coefficient	$\eta_s = 0.174 S_e^{-0.19} (\max = 1.0)$
Propagating flux ratio	$\xi = (192 + 0.2595\sigma)^{-1} \exp[(0.792 + 0.681\sigma^{0.5})(\beta + 1)]$
Wind factor	$\phi_w = C U^B (\beta / \beta_{op})^{-E}$
	$C = 7.47 \exp(-0.133\sigma^{0.55})$
	$B = 0.02526\sigma^{0.54}$
	$E = 0.715 \exp(-3.59 \times 10^{-4} \sigma)$
Slope factor	$\phi_s = 5.275 \beta^{-0.3} (\tan \phi)^2$
Effective heating number	$\varepsilon = \exp(-138 / \sigma)$
Heat of preignition (Btu/lb)	$Q_{ig} = 250 + 1116 M_f$

2.1.1.1 Fuel Particle

According to Rothermel, fuel particle properties are parameters that are intrinsic to the fuel particles including heat content, mineral content, and particle density (Rothermel, 1972). Unlike other fire spread models that treat these fuel properties as constant, this research model allows these parameters to vary in a reasonably small range. Heat content, also known as enthalpy, is a thermodynamic property of a substance that represents the total internal energy of the substance, including its internal kinetic energy and potential energy. It is a measure of the energy stored within a substance that can be transferred as heat during a chemical reaction or a physical process. In Rothermel's model heat content behaves as a factor in the numerator which is represented by heat source. The most commonly used value for heat content is 8000 Btu/lb, and in this model the heat content is varied between 7000 Btu/lb and 9000 Btu/lb (Andrews, 2018). At the same time, this model allows the user to modify these numbers since this model is used not only for wildland fire, but also for urban areas.

Mineral content refers to the presence and concentration of various inorganic elements or minerals in a substance, typically foods, beverages, soil, water, or other natural materials. Regarding the combustion of fuel, the mineral composition plays a role in influencing the combustion rate. Mineral composition is quantified as the weight of minerals per unit weight of wood and is employed to determine the net fuel load based on the oven-dry fuel load. According to Andrews (2018), at present, all standard fire behavior fuel models use 0.0555 for total mineral content and 0.01 for effective mineral content. However, the exact mineral content for different fuel type needs to be further investigated to achieve a more accurate result. In this research, the mineral content is initialized with a range between 0.03 and 0.055 (Andrews, 2018; Hough and Albini, 1978).

Particle density refers to the mass of solid particles per unit volume of a material. It is a measure of how tightly packed the solid particles are within a substance or material. Observations reveal that as particle density rises, both the propagating flux ratio and reaction intensity tend to decline. Conversely, the wind factor and slope factor exhibit an upward trend as particle density increases. In the presence of wind or slopes, the rate of spread typically rises due to the substantial impact of these factors. Conversely, in the absence of wind or slopes, an increase in particle density results in a decrease in the rate of spread (Rothermel, 1972). At the same time, it's worth mentioning that the effect of particle density has not been experimentally validated (Andrews, 2018). Given that the impact of particle density on the estimated rate of fire spread is minimal, it's common practice to utilize a standard value of 32 pounds per cubic foot (lb/ft^3) for fire behavior models. While alternative values have been employed in the past, like the Hough and Albini (1978) approach, which uses 30 lb/ft^3 for dead fuel and 46 lb/ft^3 for live fuel, this research employs a consistent input parameter of 32 lb/ft^3 (Andrews, 2018; Hough and Albini, 1978). However, it's important to note that this model still allows for potential adjustments in particle density in the future.

2.1.1.2 Fuel Array

Fuel array, also known as the fuel bed properties, includes four parameters: surface-area-to-volume ratio (SAV), fuel bed depth, fuel load, and dead fuel moisture of extinction. Fuel particle size is a critical factor that impacts both the rate of fire spread and its intensity. It is quantified using the surface-area-to-volume ratio. Another crucial factor is the ratio of fuel load to depth, which provides insights into the packing density of the fuel bed. The concept of the dead fuel moisture of extinction is equally important. This term refers to the moisture level at which dead fuel ceases to support the spread of a surface fire (Andrews, 2018). In other words, it's the point at which the fuel becomes too damp to sustain combustion and fire spread effectively. In Rothermel's model, the sizes of fuel are characterized by SAV and Table 2.4 shows fuel particle diameter in inches associated with selected SAV values.

Table 2.4 - Relationship between fuel particle diameter and surface-area-to-volume ratio. (Adapted from Andrews, 2018)

Diameter	inches	0.014	0.024	0.04	0.06	0.25	0.44	0.50	1.00	1.60	3.00
SAV	feet^2/feet^3	3500	2000	1200	750	192	109	96	48	30	16

In the past wildland fuel models, different fuel types were categorized into 1-h, 10-h, and 100-h fuels. 1-h fuels mean the smallest and finest fuels. They typically include grasses, needles, leaves, twigs, and other very small and easily ignitable materials. They are called "1-hour fuels" because they can dry out and become highly flammable within approximately one hour of exposure to drying conditions. They contribute to the initial, rapid spread of a wildfire. 10-hour fuels are slightly larger and include small branches and twigs, as well as small dead vegetation. They are called "10-hour fuels" because they take longer, around 10 hours, to dry out and become readily flammable after being exposed to drying conditions. They contribute to the sustained spread of a wildfire. Lastly, 100-hour fuels are larger dead woody materials, such as branches and small logs, which take even longer, around 100 hours, to dry out and become highly flammable. They are typically slower to ignite and contribute to the long-term or smoldering phases of a wildfire (Rothermel, 1972; Deeming et al. 1972; Andrews, 2018).

When the Surface Area-to-Volume ratio (SAV) increases, indicating the presence of finer fuels, several fire-related factors tend to increase as well. These include reaction intensity, propagating flux ratio, wind and slope coefficients, and the effective heating number. Consequently, both the heat source and the heat sink in the equation experience an increase as

SAV increases. Based on this equation and relationship, the numerator (the heat source) tends to dominate the denominator (the heat sink) as SAV increases. As a result, the overall effect is that the spread rate of the fire tends to increase. In simpler terms, when you have finer fuels with a higher SAV, the fire is more likely to spread at a faster rate due to the increased availability of fuel and enhanced heat transfer characteristics. In Rothermel's model, fuels are divided into three categories based on their diameter. Fuels with a diameter between 0 and 0.25 inches are classified as 1-hour fuels, those with a diameter of 1.25 inches are considered 10-hour fuels, and any fuels with a diameter greater than 1.25 inches are categorized as 100-hour fuels (Rothermel, 1972; Andrews, 2018).

In summary, Rothermel's fuel model classifies fuels into these three classes. 1-hour fuels are smaller than 0.25 inches and have a standard fuel model surface area-to-volume ratio (SAV) ranging from 750 to 3500 ft^2/ft^3. 10-hour fuels fall within the range of 0.25 to 1 inch in diameter and have a standard fuel model SAV of 109 ft^2/ft^3. 100-hour fuels have a diameter between 1 and 3 inches and possess a standard fuel model SAV of 30 ft^2/ft^3. In this research, the surface-to-volume ratio is initialized between 30 ft^2/ft^3 and 1200 ft^2/ft^3 in the conservative aspect (Rothermel, 1972; Albini, 1978). Further research on the surface-to-volume ratio of construction material should be conducted to achieve a more accurate result for the fire spread simulation in urban areas.

Fuel bed depth refers to the thickness or vertical measurement of a layer of combustible materials, such as vegetation, leaves, branches, or other flammable substances, that can contribute to a fire. It is an important parameter in fire science and wildfire modeling because it influences the behavior and intensity of a fire (Rothermel, 1972). Measuring fuel bed depth can be challenging, particularly due to its variability in real-world conditions. Nonetheless, fuel bed depth significantly influences fire models through its impact on bulk density (the ratio of fuel load to fuel bed depth) and packing ratio (the ratio of bulk density to particle density). As fuel bed depth increases, it tends to decrease the packing ratio, leading to higher wind and slope factors. Additionally, greater fuel bed depth reduces the heat sink effect, collectively resulting in an increased rate of fire spread. Andrews (2018) established the correlation between fuel bed depth and several fire-related factors, including rate of spread, reaction intensity, heat source, and heat sink (Rothermel, 1972; Andrews, 2018). The research considered a fuel bed depth range spanning from 0.5 to 6 feet. However, for this study, fuel bed depths ranging from 1 to 5 feet were selected as a conservative approach. It's worth noting that this simulation model permits the adjustment of this depth range in future analyses.

Fuel load refers to the total amount of combustible material or biomass present in a given area or ecosystem that can potentially burn in a fire. The fuel load is typically expressed in terms of weight per unit area, such as pounds per square foot (lb/ft^2) or kilograms per square meter (kg/m^2). Fuel load affects every component of the fire spread model through net load, bulk density, and packing ratio, and a value for oven-dry fuel load is specified for each size class (Andrews, 2018).

This research model has addressed the challenge of accurately determining fuel load from satellite images by establishing a range between 0.0205 lb/ft^2 and 0.41 lb/ft^2, though this simplification may result in a less accurate simulation. These values were derived from the 53 standard fire behavior fuel models used in the old BEHAVE system and NFDRS system (Andrews, 1986; Finney, 1998). However, it's important to note that these fuel models were primarily designed for wildland environments. In contrast, urban areas have their own unique characteristics, and the development of fuel models for combustible materials in urban settings is not as well-established. More surveying and experimental data are needed for more accurate results. As a result, the accuracy of the final simulations in urban areas may be somewhat compromised compared to those in wildland areas. This distinction underscores the complexity of modeling fire behavior in urban environments, where factors such as building materials, infrastructure, and land use patterns can significantly influence fire spread and intensity. Researchers and practitioners often face greater challenges when applying wildland-focused models to urban contexts, highlighting the need for further research and specialized modeling approaches for urban fire scenarios.

In wildland fire models, it's important to assign different moisture content values to each size class of live and dead fuel. In Rothermel's fire spread model, he introduced the concept of "moisture of extinction" as a fuel model parameter, primarily referring to the moisture content at which dead fuel no longer supports combustion effectively. This "moisture of extinction" mainly pertains to dead fuel, and for live fuel, Rothermel's model suggests calculating its moisture content based on the ratio of dead fuel moisture (Rothermel, 1972). However, it's worth noting that Rothermel's model assumes that when only living fuel is present, the model predicts no fire spread and no reaction intensity. In reality, though, live fuel alone can indeed propagate a fire (Andrews, 2018). This limitation in the model highlights an area of incompleteness, as it doesn't fully account for the potential contribution of live fuel to fire behavior. Fire models continually evolve and improve to better represent the complexities of real-world fire behavior, and researchers are constantly working to refine these models to make them more accurate and comprehensive.

In the research model developed for this study, a constant dead fuel moisture of extinction value of 0.3 is used. This value is derived from the original work by Rothermel in

1972. It's worth noting that Albini, in a subsequent paper in 1976, modified the calculation of the dead fuel moisture of extinction to avoid implicit assumptions (Rothermel, 1972; Albini, 1976). However, the results obtained from using the constant value of 0.3 do not vary significantly, and for simplicity, this research has chosen to retain the value of 0.3. This decision to use a constant value simplifies the modeling process while still providing reasonable results.

2.1.1.3 Environmental Input

The environmental input data consists of three variables: fuel moisture content, wind velocity, and slope steepness. Rothermel's model defines fuel moisture content as the ratio of the weight of water to the dry weight of the fuel, measured on a dry weight basis. Wind speed is considered at midflame height, which is the wind that directly influences surface fires. Additionally, slope steepness is expressed as a fraction known as tangent, calculated by dividing the vertical rise by the horizontal distance along the terrain (Rothermel, 1972). In this dissertation, the factor of slope steepness is combined with the factor of wind, as they are considered together in the calculation of the rate of spread using Rothermel’s model.

The moisture content for dead fuels can vary widely depending on environmental conditions, weather, and the specific type of dead fuel. In general, for live fuels that are converted to dead fuels, the moisture content ranges from approximately 30% to 150% or more. The moisture content tends to decrease as the fuels dry out. For dead leaves, grasses, and fine dead fuels, they usually have moisture content in the range of 5% to 20%. For logs and larger dead wood, they typically have lower moisture contents, often in the range of 5% to 15% or less. Larter diameter logs may have lower moisture contents due to their thicker bark and slower drying rates. Simultaneously, Rothermel's equation for the moisture damping coefficient implies that the ratio of moisture content to dead fuel moisture extinction should not exceed 1, effectively capping the maximum moisture content at 0.3 (Rothermel, 1972; Andrews, 2018). In this research model, moisture content is stochastically initiated within the range of 0.02 to 0.3.

Rothermel's surface fire spread model integrates both the influence of wind and slope, incorporating a modification factor to calculate the final rate of spread. Originally, the model was formulated without accounting for wind and slope effects, but these factors were subsequently introduced primarily to augment the rate of spread. The wind factor is associated with the geometric characteristics of fuel particles and the fuel bed. Specifically, it corresponds to the midflame wind speed, which signifies the wind conditions at the midpoint of the flame (Rothemel, 1983). It's worth noting that this definition of midflame wind speed, while useful, lacks precision and can be challenging to measure accurately (Andres, et al., 2013). Within this dissertation, we assume a linear relationship between the ground height and the height at which wind measurements are taken to estimate the midflame wind speed. Wind direction will be discussed in the next section since it is associated with the fire elliptical growth model.

The slope factor is inversely related to the packing ratio, meaning that as the packing ratio increases, the slope factor decreases. In other words, fuels that are tightly packed tend to exhibit lower slope factors (Andrews, 2018). In this dissertation, the slope factor is combined with the wind factor and randomly initialized. In other words, this dissertation combines slope steepness and wind velocity since they sum up with a factor ($1 + \varphi_w + \varphi_s$). Assuming steepness is zero and randomize the wind velocity. The reason is that it is very difficult to retrieve the exact slope information only from the satellite image.

2.1.2 Fuel models

Fuel type is not like a classification of a normal material. It is difficult to describe all physical or other characteristics for all fuels in an area. Thus, a fuel type has been defined as “an identifiable association of fuel elements of distinctive species, form, size arrangement, and continuity that will exhibit characteristic fire behavior under defined burning conditions” (Merrill and Alexander, 1987).

Rothermel’s surface fire spread model is one of the most widely used fire models and it was developed as a result of a 1968 plan for a complete fire danger rating system; besides this model, the National Fire Danger Rating System (NFDRS) just like BEHAVE and FARSITE is also a result of 1968 plan (Deeming et al., 1972; Lundgren, et al., 1995). The NFDRS developed a seasonal weather system and these weather conditions are along with the NFDRS fuel models (Deeming et al., 1972; Deeming et al.,1977). In addition to the NFDRS fuel type model, the BEHAVE system accomplished a different fuel type classification which is known as NFFL (Northern Forest Fire Laboratory) (Albini, 1976a; Burgan and Rothermel, 1984). Table 2.5 shows the NFDRS fuel classification system according to Cohen and Deeming (1982). Table 2.6 shows the NFFL fuel classification system according to Andrews (1986).

Table 2.5 - NFDRS fuel classification system. (from Arroyo, et al., 2008)

Fuel type	**Fuel model**	**Fuel parameters**							
		Fuel loadings (t/acre)						Fuel depth (ft)	Moisture of extinction dead fuels (%)
		1 h	10 h	100 h	1000 h	Wood	Herb		
Western grasses (annual)	A	0.20					0.30	0.80	15
Western grasses (periannial)	L	0.25					0.50	1.00	15
Sawgrass	N	1.50	1.50			2.00		3.00	25
Pine-grass savanna	C	0.40	1.00			0.50	0.80	0.75	10
Southern rough	D	2.00	1.00			3.00	0.75	2.00	30
Sagebrush-grass	T	1.00	0.50			2.50	0.50	1.25	15
California chaparral	B	3.50	4.00	0.50		11.50		4.50	15
Intermediate brush	F	2.50	2.00	1.50		9.00		4.50	15
Hardwood litter (winter)	E	1.50	0.50	0.25		0.50	0.50	0.40	25
Southern pine plantation	P	1.00	1.00	0.50		0.50	0.50	0.40	30
Hardwood litter (summer)	R	0.50	0.50	0.50		0.50	0.50	0.25	25
Western pines	U	1.50	1.50	1.00		0.50	0.50	0.50	20
Heavy slash	I	12.00	12.00	10.00	12.00			2.00	25
Intermediate slash	J	7.00	7.00	6.00	5.50			1.30	25
Light slash	K	2.50	2.50	2.00	2.50			0.60	25
High pocosin	O	2.00	3.00	3.00	2.00	7.00		4.00	30
Tundra	S	0.50	0.50	0.50	0.50	0.50	0.50	0.40	25
Short-needle conifer (normal dead)	H	1.50	1.00	2.00	2.00	0.50	0.50	0.30	20
Short-needle conifer (heavy dead)	G	2.50	2.00	5.00	12.00	0.50	0.50	1.00	25
Alaskan black spruce	Q	2.00	2.50	2.00	1.00	4.00	0.50	3.00	25

Table 2.6 - NFFL fuel classification system. (from Arroyo, et al., 2008)

Fuel type	**Fuel Model**	**Fuel parameters**					
		Fuel loadings (t/ha)				Fuel depth (ft)	Moisture of extinction
		1h	10h	100h	life		
Grass and grass-dominated							
Short grass (30 cm)	1	0.74	1	0.5	0.5	1	12
Timber	2	2				1	15
Tall grass (76 cm)	3	3.01				2.5	25
Chaparral and shrub fields							
Chaparral (18 cm)	4	5.01	4.01	2	5.01	6	20
Brush (61 cm)	5	1	0.5		2	2	20
Dormant brush, hardwood slash	6	1.5	2.5	2	0.37	2.5	25
Southern rough	7	1.13	1.87	1.5		2.5	40
Timber litter							
Closed timber litter	8	1.5	1	2.5		0.2	30
Hardwood litter	9	2.92	0.41	0.15		0.2	25
Timber (litter and understory)	10	3.01	2	5.01	2	1	25
Slash							
Light logging slash	11	1.5	4.51	5.51		1	15
Medium logging slash	12	4.01	14.03	16.53		2.3	20
Heavy jogging slash	13	7.01	23.04	28.05		3	25

Arroyo, et al. (2008) summarized the limitations of the different fuel models from past researches. Since each fuel classification is applied to a specific software application, they are not able to exchange or be used in other types of software (Arroyo, et al., 2008). Although the main theory that used in the U.S. is based on Rothermel's model, the development and application of those systems are different. As a result, the fuels for each set of equations are significantly different. Finally, it is difficult to map the fuel type. Time and space changes are probably the main problem for an accurate fuel mapping.

The most popular method currently used to determine the fuel type is remote sensing, though in the past, people used traditional methods to map fuel types like field survey. Clearly, field survey needs too much time and cost to be accomplished. At the same time, the accuracy of traditional methods varies with seasonal change and needs to be updated every certain period of years. Here, Arroyo, et al. (2008) also provided some advantages and disadvantages, techniques and accuracies for different remote sensing data applied to fuel mapping.

In the development of this fire spread model, a gridded map layout was adopted, where each grid node, represented a specific portion of the terrain. This approach allowed for an in-depth, localized analysis of fire spread, reflecting the complexity of real-world landscapes. Therefore, at the beginning of the simulation, fuel types were randomly assigned to each node, this step was crucial in modeling the natural diversity and unpredictability of fuel distribution in real environments, where various combustible materials area scattered in an irregular pattern. The random assignment of fuel types, each with its unique fire behavior properties, added a critical layer of realism to the simulation.

The Monte Carlo simulation technique was then employed to model the spread of fire across the grid. By calculating the random spread rate for fire spreading to adjacent nodes, based on fuel type and local environmental and fuel particle parameters, a dynamic and unpredictable pattern of fire spread was simulated. Each iteration of the model represented a different potential outcome, showing the varied ways that a wildfire might progress under different scenarios. This iterative process was key in capturing the inherent unpredictability and variability of wildfire behavior.

Recent studies show that there are now more accurate assessments of fuel conditions using deep learning neural networks. Although remote sensing for fuel identification has existed for decades, accuracy has improved in recent years due to advancements in machine learning and deep learning techniques. Large amounts of data were extracted from multispectral signatures, high-resolution imagery, and biophysical climate and terrain data (Alipour, et al., 2023; Shaik, et al., 2023). This significant increase in data has greatly enhanced the accuracy of fuel identification. An ensemble model developed by Shaik in March 2024 uses diverse data sources and is trained with USDA Forest Service FIA data. Initially, the results were poor, but after training with deep learning neural networks and other machine learning techniques, the final fuel mapping accuracy reached about 80% (Shaik, et al., 2024).

2.2 Fire Spread Simulation Models

2.2.1 Elliptical Growth Model

First of all, fire growth modeling is an intricate and essential process in understanding and managing wildfires. It involves using mathematical and computational methods to simulate the behavior and spread of wildfires under a variety of environmental conditions. At the same time, the elliptical growth model combines not only a mathematical approach but also empirical results that come from past research. Figure 2.1 shows the four geometric shapes that have been used to model fire growth in homogenous environments. The primary goal of this modeling is to accurately predict the rate of spread, direction, intensity, and eventual area coverage of a wildfire (Tymstra, et al., 2010). These predictions are critical for effective fire management strategies, including planning evacuations, allocating firefighting resources efficiently, and minimizing the overall impact and damage caused by wildfires.

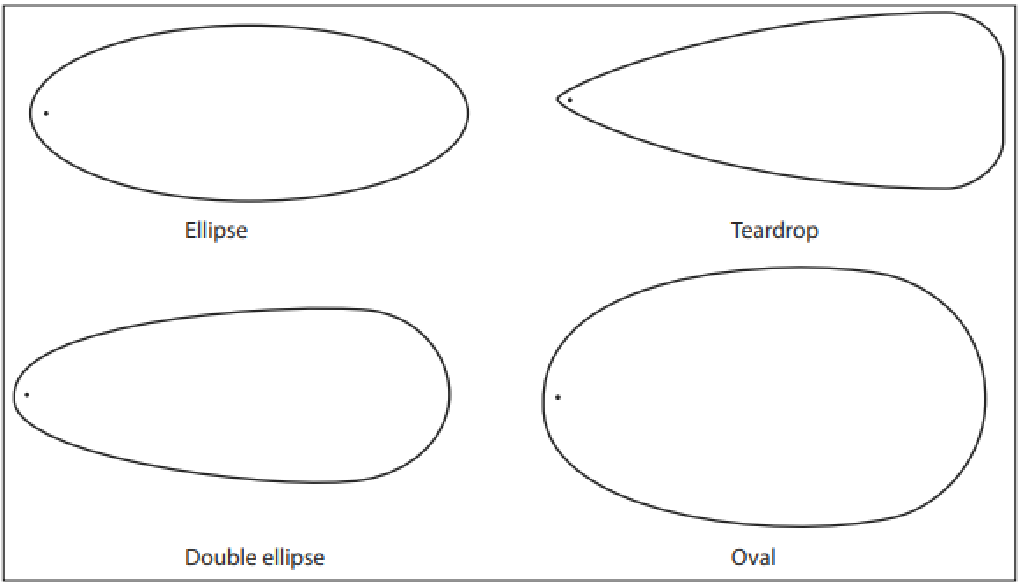


Figure 2.1 - Four geometric shapes that have been used to model fire growth in homogenous environments. (from Tymstra, et al., 2010)

One of the most popular methods in fire growth modeling is the elliptical growth model (Albini, 1976). This approach is particularly favored for its ability to accurately predict fire fronts, especially in grassland fires. Developed based on a system of differential equations, the elliptical growth model takes into account variable factors such as fuel types, weather conditions, and topography. Being a first-order, non-linear model, it utilizes parameters that are typically available from forestry data, enabling the simulation of complex fire scenarios including those with varying fuel, and wind conditions, and the inclusion of fire breaks (Albini, 1976). Figure 2.2 shows the approximate fire shapes for wind speeds of 5, 10, 20, and 30 mph.

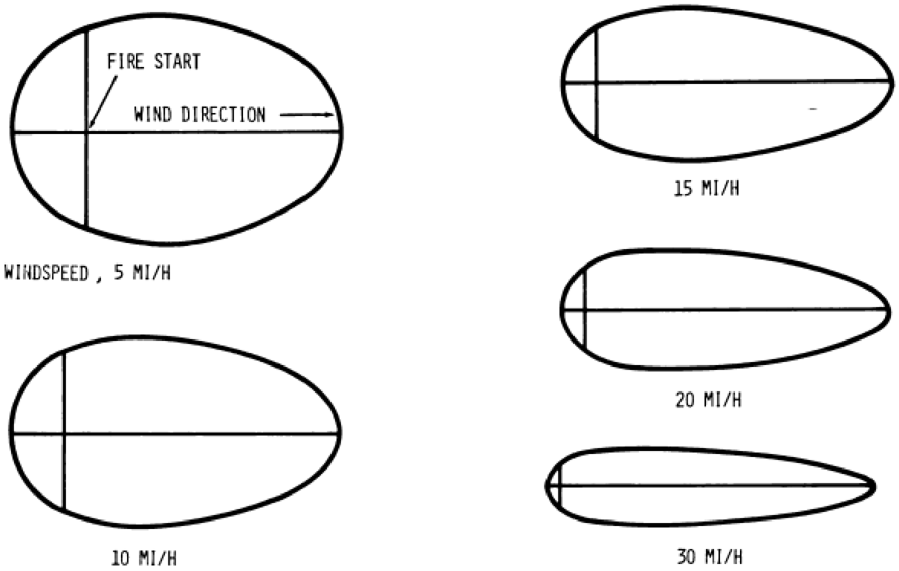


Figure 2.2 - Approximate fire shapes for windspeeds of 5, 10, 15, 20, and 30 mi/h. (from Albini, 1976)

The elliptical growth model is rooted in the concept that under constant, homogeneous conditions, like steady wind over flat terrain with a uniform fuel supply, a fire will spread in an elliptical shape. This model is particularly effective due to its simplicity and efficiency in homogeneous conditions, where factors like wind, fuel, and topography remain relatively constant. It is based on empirical observations and historical data on fire spread under specific conditions. Over time, this model has been refined to accurately reflect the behavior of fires in a wide range of scenarios.

The model's adaptability is one of its key strengths. While the basic shape of the fire front in the model is an ellipse, it can be adjusted to reflect different rates of spread in various directions. This is influenced by factors such as wind direction and speed, the slope of the terrain, and the types of fuel present. This adaptability makes the elliptical growth model a versatile tool for different fire scenarios, enhancing its applicability in diverse environmental settings (Albini, 1976).

Integration with other fire prediction systems and tools is another advantage of the elliptical growth model. This integration enhances the overall predictive capability of the model, providing a more comprehensive understanding of fire behavior. Such integration is crucial for strategic planning in fire management, as it aids in forecasting the potential spread of a fire. This forecasting is essential for making informed decisions about resource allocation, evacuation plans, and firefighting strategies.

Despite its advantages, it's important to acknowledge that the elliptical growth model also has limitations, particularly in heterogeneous conditions where environmental factors such as wind speed, fuel type, and topography change rapidly. In such complex circumstances, more sophisticated models that account for these varying conditions are often required. These models may involve more complex computational techniques but offer a more detailed and accurate representation of fire behavior in diverse landscapes. By integrating the elliptical growth model with other methods will help to improve the accuracy in a detailed manner.

In conclusion, the elliptical growth model is a significant contribution to fire management and research. It offers a balance of simplicity and adaptability, making it a valuable tool in both predicting and understanding fire behavior in a range of settings. However, the accuracy and applicability of the model depend on the reliability of input data and the assumptions made about fire behavior under different conditions. As such, it remains one part of a broader toolkit of models and strategies used by fire managers and researchers to tackle the complex challenge of understanding and managing wildfires.

2.2.2 Ellipse Dimensions

In 1982, D.H. Anderson proposed an alternative deterministic model, originally inspired by Albini's 1976 concept of an elliptically-shaped burning area and Huygens' Principle. It is assumed that a uniform, constant wind blows, with the x-axis aligned in the wind's direction. The fire's behavior is influenced by the wind, as it alters the angle of the flame front (Anderson, et al., 1982). Anderson defines the propagation of an elliptical fire from an ignition point with a uniform wind speed with parameters f, g, and h as shown in Figure 2.3a. These are wind-dependent parameters that will change the spread rate at that ignition point associated with the influence of wind. The length-to-width ratio which can be calculated by the ratio of $\frac{g+f}{h}$ and is used to demonstrate how the ellipse is going to be elongated with the wind speed.

On the other hand, in the paper "Explanation of Rothermel" (Andrews, 2018), provided a clearer definition of the ellipse parameters which are *DH* (heading spread distance), *DB* (backing spread distance), and *DF* (flanking spread distance); which is shown in Figure 2.3b (Andrews, 2018). Heading spread distance can be calculated through the multiplication of the heading spread rate and the time step. Backing spread distance and flanking spread distance can be calculated similarly. In this dissertation, the ellipse parameters f, g, and h will be used to generate the elliptical shape of fire propagation since it will be easier to implement in Matlab.

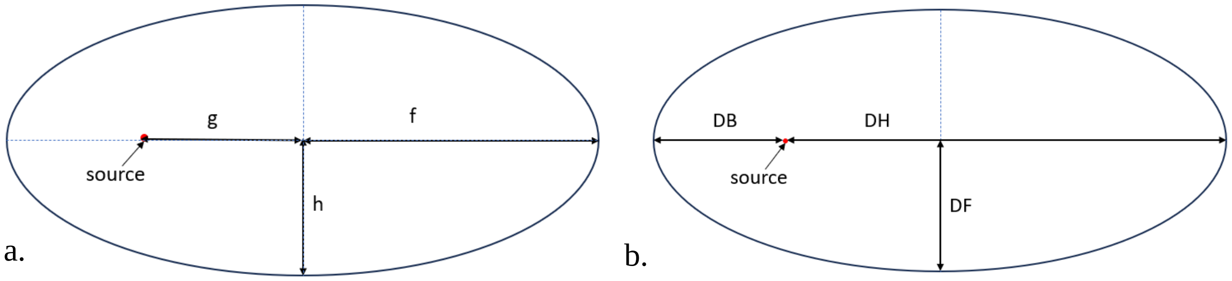


Figure 2.3 – a) Elliptical growth parameter defined by Anderson, 1982; b) Modified definition by Andrews, 2018. (Adapted from Anderson et al., 1982 & Andrews, 2018)

2.2.3 Huygens' Principle and Modification

Huygens' Principle of Fire Front Expansion is a pivotal concept in fire growth modeling, particularly in sophisticated vector or wave-type models (Huygens, 1962). Figure 2.4 shows the formation of a new wavefront using Huygens' principle. This principle is significant to understanding and predicting the behavior and spread of wildfires, especially in models like Prometheus, which is designed to assist fire managers in strategic decision-making. The principle operates on the premise that each point along a firefront serves as an independent origin for new growth, conceptualized as elliptical wavelets. These wavelets, shaped and oriented according to the local environmental conditions like fuel, weather, and topography at each point or vertex, represent potential fire growth (Finney, 1999). This approach is instrumental in addressing the challenges of spatial and temporal heterogeneity that are characteristic of wildfire behavior.

In practical terms, Huygens' Principle is applied by expanding the fire front through the accumulation of these individual wavelets around the existing front. This method effectively captures the highly variable conditions a fire might encounter, offering a dynamic and versatile means to model fire behavior. The flexibility of this approach lies in its ability to adapt the shapes and sizes of the wavelets according to the unique conditions at each vertex, enabling the model to depict complex fire growth patterns under a range of environmental settings. Integrating Huygens' Principle with fire behavior prediction systems like the Canadian Forest Fire Behavior Prediction System, models like Prometheus can simulate fire growth with precision. This integration facilitates the generation of detailed fire perimeters at set intervals, with each vertex along the perimeter yielding specific fire behavior outputs as shown in Figure 2.5 (Tymstra, et al., 2010). These outputs are not only comprehensive but also compatible with geographic information systems, enhancing their utility in fire management and planning.

In environments marked by heterogeneity, varying weather, fuel types, and terrain, Huygens' Principle proves particularly valuable. It allows for the simulation of fire spread rates and growth patterns in such complex settings, with each vertex on the fire front independently dictating the spread rate and direction based on local conditions. This method leads to the creation of an 'envelope' of wavelets, encapsulating the preceding fire front and projecting the fire's future trajectory based on its present state (Finney, 1999).

In essence, Huygens' Principle of Fire Front Expansion offers a robust, adaptable framework for modeling wildfires. It provides a nuanced way to understand and predict fire behavior across diverse landscapes and changing conditions, making it a cornerstone in advanced fire growth simulation efforts.

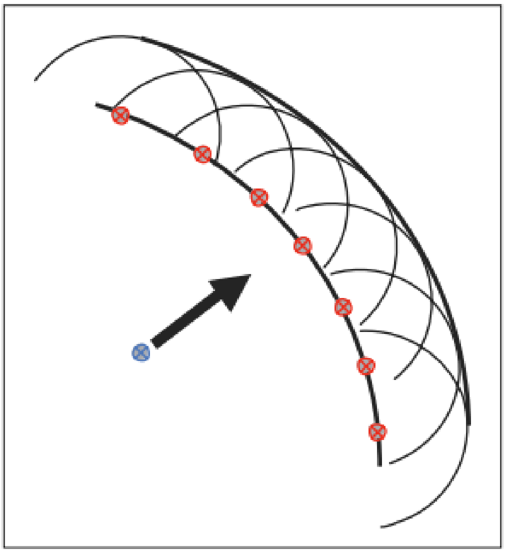


Figure 2.4 - **Formation of a new wavefront using Huygens' principle**. The blue crossed circle is the point of origin, the arrow represents the spread direction of the wave, and the red crossed circles represent the origins of secondary waves. (from Tymstra, 2010)

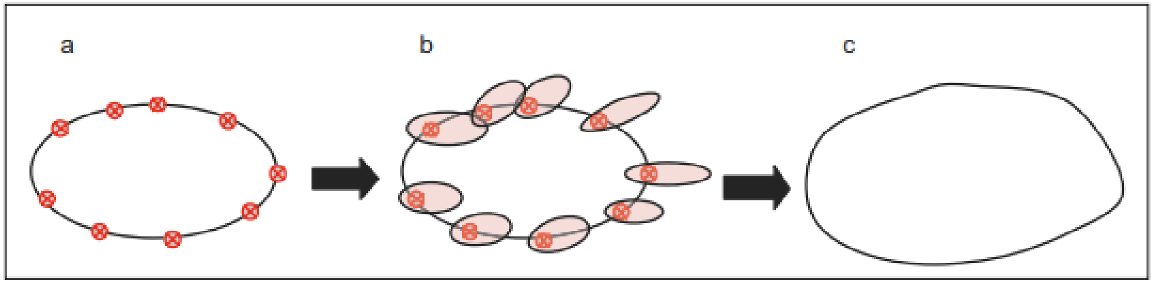


Figure 2.5 – **Application of Huygens' principle to simulate fire growth**. a) Points of origin (vertices) for wave propagation are identified around the fire perimeter (red crossed circles) at time *t*; b) Elliptical firelet growth over elapsed time *dt* (red-shaded ellipses) is projected using spread functions. c) The new fire perimeter at time *t* + *dt* is drawn as the tangential envelope of the firelets. (from Tymstra, 2010)

2.2.4 Integration of Fire Elliptical Growth Model and Huygens' Principle in a Grid-Based Simulation

This research delves into an advanced approach for simulating wildfire spread by combining the fire elliptical growth model with Huygens' Principle of Fire Front Expansion, uniquely adapted to a grid-based system. The primary innovation lies in the selection of vertices for wavelet propagation, which, unlike traditional methods that use the ellipse's perimeter points, are determined by the outermost grid nodes enclosed by the elliptical model. The detailed procedure is discussed as follows.

This study begins by segmenting the landscape into a grid, a method that allows for meticulous and localized analysis of fire spread. This grid framework is crucial for examining the fire's behavior across diverse terrains in a detailed manner. Figure 2.6 shows an example of a grid-based map. On each grid node, there will be several layers of data including the environmental parameters, fuel information, and a node type indicator. The function reads all the data from a spreadsheet and uses Rothermel's model to calculate the spread rate on each node. Besides assigning the node type and spread of rate, upon this grid, the elliptical growth model is implemented, projecting the fire spread in an elliptical shape under the influence of environmental factors like wind, fuel, and topography. This model establishes an initial estimation of the area likely to be affected by the fire, outlining the prospective spread of the fire front over time.

A significant adaptation in this research is the application of Huygens' Principle. Traditionally, this principle utilizes the vertices on the ellipse's perimeter to simulate new growth points. However, this study innovates by using the most external grid nodes that the ellipse encloses as vertices. Although there may be a slight discrepancy between the vertices and the ellipse, if the grid size is sufficiently small, the simulation results should be adequately accurate. This approach is designed to enhance the model’s precision, especially in representing the intricacies of fire behavior across varied and complex landscapes.

The simulation process treats each selected grid node as an independent origin for new fire growth. As the meshed grid gets smaller, the grid nodes will be closer to the ellipse itself, and finally, the grid node will be closed enough and can be considered as the vertices in Huygens’ Principle. Figure 2.7 shows an example of the simulation developed in this study integrating Hugens’ Principle, elliptical growth model, and grid-based map. The Red dashed line is the perimeter fire front used for the fire propagation next time step and the black ellipses represent the real firefront perimeter. This method provides a detailed, grid-by-grid analysis of how the fire is likely to expand, taking into account the interactions with local environmental conditions. This granular view offered by the grid-based approach contributes to a deeper understanding of wildfire dynamics, especially in heterogeneous landscapes.

The results from this methodological innovation indicate that focusing on the outermost grid nodes allows for capturing subtle variations in the fire's behavior. These variations, influenced by localized factors, might be overlooked in broader, traditional models. The integration of the elliptical growth model with Huygens’ Principle, tailored for a grid-based system, introduces a comprehensive and multifaceted framework for predicting wildfire spread.

Figure 2.6 - An example of a grid-based map

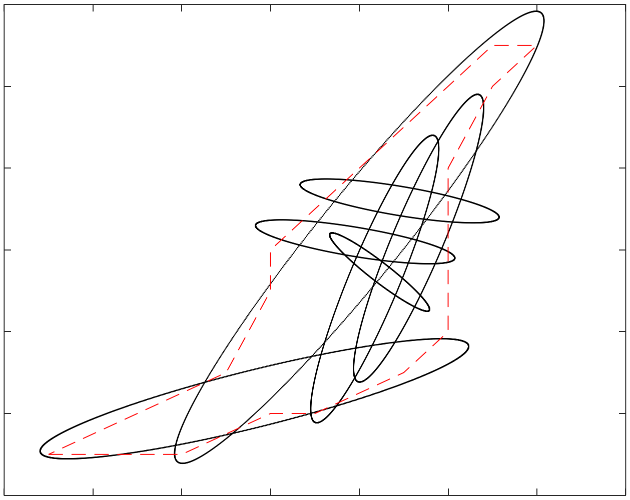


Figure 2.7 – **An example of the simulation in this model integrating Hugens' Principle, elliptical growth model, and grid-based map**. The Red dashed line is the perimeter fire front used for the fire propagation next time step. The black ellipses represent the real firefront perimeter.

2.2.5 Monte Carlo Simulation

Monte Carlo simulation is recognized as a computational technique whereby numerical results are obtained through random sampling. This method of simulation involves the definition of a domain of possible inputs, the generation of random inputs, the performance of a deterministic calculation for each input, and the aggregation of results. In this dissertation, the initiation will involve specifying wind velocity and wind direction as potential input variables in the model, with all other parameters, including fuel arrays and fuel particles, being held constant due to the predetermined information on the map.

From the range of inputs defined, a large number of random inputs are to be generated by the simulation. Uniform distribution will be employed in this model for the generation of random wind velocity and wind direction within a user-defined range. It will be ensured that, on each grid node, the wind velocity and direction are unique and independent from all others. Following the generation of random input, the model will be run, and results corresponding to each set of random inputs will be produced. The model is then expected to aggregate these results to produce a probability distribution, which will be utilized to estimate the probabilities of ignition at each node. Consequently, an estimation of the economic loss can be made.

The capability of Monte Carlo simulations to model complex systems and scenarios, where analytical solutions may be challenging or unattainable, is highlighted as a significant strength. The simulation of fire spread represents such a scenario, underscoring the rationale behind the adoption of Monte Carlo simulation for this purpose.

Another reason Monte Carlo simulation is employed relates to the model's approach of reading the spreadsheet as input, resulting in the wind velocity and direction remaining constant throughout the entire simulation. With the utilization of a grid map for the simulation, it is ensured that each node stores its wind velocity and direction. Through the application of Monte Carlo simulation, an investigation into all possible outcomes is conducted, culminating in the determination of the final fire spread range and the associated economic loss for the area.

2.3 Concluding Remarks

The research explored here innovatively combines the fire elliptical growth model with Huygens' Principle in a grid-based wildfire simulation to enhance the accuracy and depth of wildfire spread predictions. The elliptical growth model, grounded in empirical data and mathematical equations, provides a robust foundation for predicting the fire's spread rate, direction, intensity, and area coverage. This model's adaptability and integration with other systems make it an effective tool for fire management, especially in homogeneous conditions.

However, the real advancement in this research lies in the integration of Huygens' Principle. Traditionally used in complex vector or wave-type models, this principle considers each point on a firefront as an independent source of new growth, shaped by local conditions. The research's novel approach uses the outermost grid nodes enclosed by the elliptical model as the vertices for new growth, diverging from the traditional method of using the ellipse’s perimeter points. This grid-based system allows for a more detailed analysis of fire spread across diverse landscapes, taking into account the heterogeneous conditions often encountered in wildfires.

This integrated approach offers a nuanced understanding of wildfire dynamics, capturing subtle variations in fire behavior influenced by localized factors. The grid-based method provides a detailed view of the fire's potential expansion, enhancing the precision of the model in complex environments. The spread rate is provided by Rothermel's surface fire spread model, while the elliptical fire growth model supplies the foundational framework for fire propagation. The introduction of Huygens' Principle adds complexity and flexibility to the model. Ultimately, the inclusion of the firebrand model, which will be discussed in the next section, enhances the completeness, realism, and accuracy of the model, ensuring a more comprehensive approach to understanding and predicting fire spread.

In conclusion, this new method contributes significantly to the field of fire management and research. It presents a balanced approach that combines the simplicity and efficiency of the elliptical growth model with the adaptability and detailed predictive capabilities of Huygens' Principle in a grid-based simulation. This integration of elliptical growth model and Huygens' Principle offers a comprehensive framework for predicting and understanding wildfire behavior across various environmental settings, making it a valuable tool for strategic fire management planning. The success of this approach, however, hinges on the reliability of input data and the assumptions made about fire behavior, underscoring the importance of accurate and comprehensive data collection in fire management practices. Finally, the Monte Carlo simulation method will be employed to compensate the variation that comes from the wind direction and velocity.

3. FIRE SPREAD IN URBAN AREA AND STRUCTURE BEHAVIOR UNDER FIRE

3.1 Firebrands in Wildland Fire Spread Simulation

Firebrands play a crucial role in the spread of wildland fires, significantly impacting how these fires behave and extend their reach. One of the primary ways firebrands contribute to wildfire propagation is through the creation of spot fires. These are new fires ignited by burning embers or pieces of combustible material carried by the wind, often landing considerable distances from the main fire front (Suzuki, et al., 2014). The ability of firebrands to ignite spot fires leads to a rapid and unpredictable expansion of wildfires, as these new fires can occur far from the initial blaze.

The significance of firebrands extends to their ability to bypass natural and man-made barriers, such as rivers, roads, or firebreaks. This ability complicates fire containment efforts, as firebrands can easily cross these barriers, which would typically halt or slow down a fire front. As a result, wildfires can spread to new areas, making suppression efforts more challenging. Additionally, the ignition of multiple spot fires by firebrands can lead to a more intense and rapidly spreading wildfire. When these spot fires merge, they create a larger, more uncontrollable fire front. This not only intensifies the wildfire but also poses significant challenges for firefighters and impacts the strategies used in wildland fire management.

In regions where wildland interfaces with urban development, firebrands pose a particular threat. They can ignite structures and vegetation within urban areas, leading to property damage and potentially causing loss of life. This urban interface threat requires specialized attention and strategies to protect human lives and property. Firebrands also contribute to the overall severity and ecological impact of wildfires. By causing fires to spread over a larger area, they can lead to greater ecological damage, affecting wildlife habitats and biodiversity. The severity of these fires can have long-lasting impacts on the environment. Moreover, the unpredictable nature of spot fires caused by firebrands presents significant challenges in ensuring public safety (Fernandez-Pello, 2017). Evacuation plans need to be flexible and responsive to the rapid changes in fire behavior and direction that firebrands can cause.

In summary, the role of firebrands in wildland fire spread is multifaceted and significant. They influence the behavior, intensity, and management strategies of wildfires, making them a critical factor in understanding and responding to these natural disasters. Recognizing the impact of firebrands is essential for effective wildland fire suppression, safeguarding public safety, and minimizing damage to both natural and urban environments.

However, the contributions to the spread of the fire by firebrands have not been included in Rothermel's model (Andrew, 2018). In other words, the mere presence of firebrands in the air and their landing ahead of a firefront does not automatically equate to them being effective in advancing the fire. This distinction is crucial. Berlad's study is referenced to support this point, indicating that not all firebrands significantly contribute to the spread of fire (Berlad, 1970). For a firebrand to be significant in this context, it must meet certain conditions. First, the firebrand must release enough heat to ignite adjacent fuels. This is a critical requirement because, without sufficient heat, the potential for starting a new fire or spot fire is greatly diminished. Second, the ignition by the firebrand must occur before the area would have been reached by the fire through conventional heat transfer mechanisms (Berlad, 1970). In other words, the firebrand must create a new ignition point that accelerates the spread of the fire beyond its natural progression.

This perspective offers a more refined understanding of the role of firebrands in wildfire dynamics. It suggests that while firebrands are a visible and dramatic aspect of wildfires, their actual impact on fire spread must be evaluated based on their ability to generate sufficient heat quickly enough to outpace the fire's natural spread (Berlad, 1970).

In the context of wildfire modeling, such as in the Rothermel model, this understanding may explain why contributions from firebrands are not explicitly included. The model may focus more on conventional heat transfer mechanisms to predict fire spread (Andrews, 2018). However, this does not diminish the potential importance of firebrands, especially in specific scenarios where their heat release and timing do significantly influence fire behavior. It does, however, highlight the complexity of accurately incorporating firebrand behavior into predictive models and the need for continued research and observation to better understand their role in wildfire dynamics.

3.1.1 Past Firebrand Experiments

The research on firebrands from past literature offers a detailed exploration into their role in fire spread. In structural fires, firebrands are predominantly small and lightweight, a characteristic that allows them to be easily transported by wind. The studies show that the size, weight, and material of firebrands are critical factors in their flight and combustion behavior. In wildland fires, especially those originating from trees, the modeling of firebrands' flight paths highlights the influence of wind and fire intensity on their travel and burning behavior. According to the full-scale structure firebrands test which was conducted under controlled laboratory conditions in a wind tunnel gives a better understanding of the firebrand generation during structural fires (Suzuki, et al., 2014). The main findings include that more than 90% of the firebrands were under 1 gram, with a significant portion (56%) being less than 0.1 gram.

Additionally, this full-scale structure was not equipped with furniture, as would be typical in a normal residential building (Suzuki, et al., 2014). It is a reasonable assumption that a real residential structure could generate more firebrands than what was observed in this experiment. Figure 3.1 to Figure 3.3 shows the size and mass distribution from different experiments (Suzuki, et al., 2014; Suzuki, et al., 2012; Suzuki, et al., 2018).

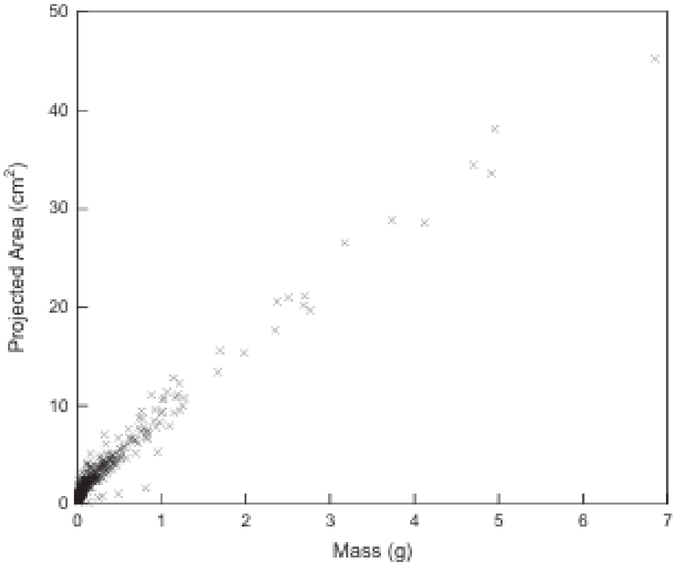


Figure 3.1 – Size and mass distribution of firebrands collected from a full-scale structure burning under well-controlled latboratory conditions. (from Suzuki, et al., 2014)

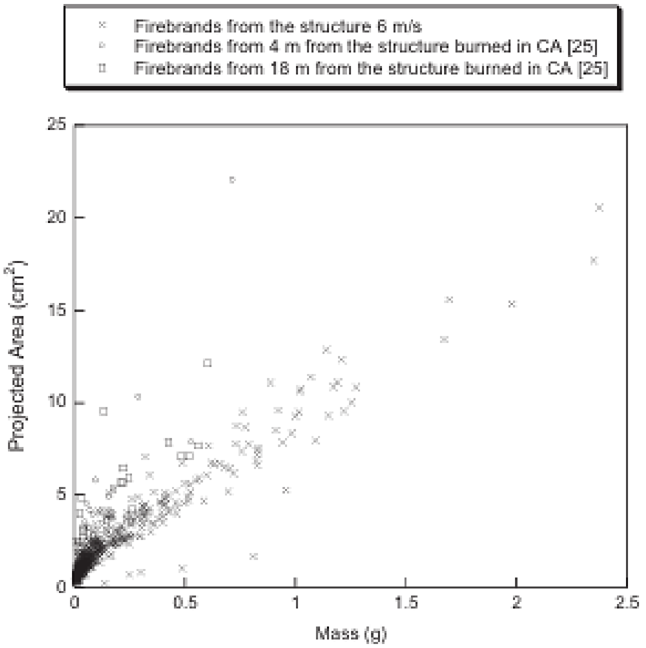


Figure 3.2 – Comparison of size and mass distribution with full-scale structure burn in Dixon, CA. (from Suzuki, et al., 2012)

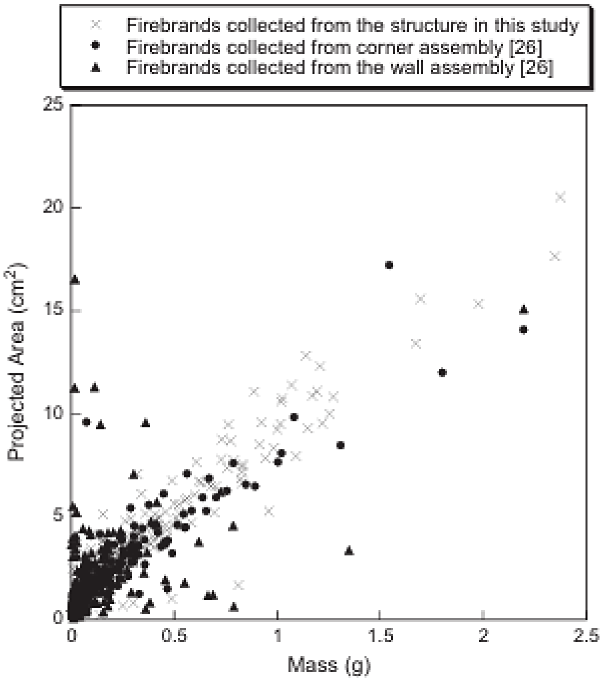


Figure 3.3 – Comparison of size and mass distribution with components test under similar laboratory conditions. (from Suzuki, et al., 2013)

Lastly, the ignition potential of firebrands, determined by factors like their size, material, and environmental conditions, underscores their significance in initiating new fires. This body of work emphasizes the complex interactions and significant impact of firebrands in fire propagation, highlighting the necessity for advanced strategies in fire prevention and management.

3.1.2 Firebrand Model

In this dissertation, the firebrand model adopted is derived from the paper "Development and validation of a physics-based urban fire spread model" by Himoto and Tanaka in 2008. This model, designed for fire spread in densely built urban areas, explicitly incorporates physics-based knowledge to describe fire spread phenomena. When addressing building-to-building fire spread, three mechanisms are identified as key contributors: thermal radiation from buildings involved in fire, temperature rise due to wind-driven fire plumes, and firebrand spotting. Figure 3.4 demonstrates the probability of firebrand scattering released from a fire-involved building (Himoto & Tanaka, 2008). However, this dissertation will utilize Rothermal's surface spread model to represent the first two mechanisms, while the spot fire spread model will be implemented separately.

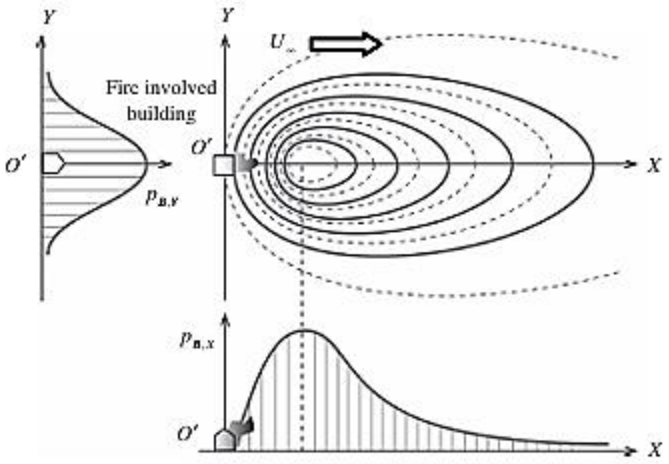


Figure 3.4 – Probability of firebrand scattering released from a fire-involved building. (from Himoto & Tanaka, 2008)

In this dissertation, a 3-D firebrand model was generated, in which the x and y axes represent the distance, and the z axis represents the probability of the firebrand ultimately reaching a location. From the time perspective, calculating the probability of the firebrand's location at every time step would be excessively slow. Moreover, the running time could potentially increase exponentially due to the exponential increase in affected nodes at each time step. The details about the 3-D firebrand model will be discussed in Chapter 4.

It should be noted that in this model, only the distance that a firebrand may be added to the original firefront is considered. In other words, the firebrand will not be considered as a new ignition point that starts a new fire ellipse at that location. It is assumed that the distance is relatively small compared to the rate of fire spread, and the gap will be encompassed within a single time step. This assumption is based on the likelihood that the distance a firebrand can travel will most likely not exceed 8 meters or 26.2 feet, which is significantly less than the fire spread rate within one time step (Himoto & Tanaka, 2008).

3.2 Structure Behavior Under Fire

The study of structural behavior under fire conditions is a critical aspect of fire safety engineering, as it directly impacts the integrity of buildings and the safety of their occupants during fire incidents. Understanding how different materials and components, such as concrete and steel, respond to fire is essential for designing structures that are both safe and resilient. This becomes particularly significant when considering the dynamic relationship between time and temperature during a fire event.

As a fire progresses within a compartment, the temperature increases, causing a series of physical and chemical changes in the structural materials. The behavior of these materials under high-temperature conditions is complex and varies depending on their properties. For instance, concrete may experience spalling and a reduction in strength, while steel might lose its load-bearing capacity due to thermal expansion and softening (Khoury, 2000; Mahmoud, et al., 2015).

The relationship between time and temperature is vital to understand for several reasons. Firstly, it helps in predicting the structural integrity and failure modes of a building during a fire, allowing for better emergency response strategies and evacuation plans. Secondly, it informs the design and selection of materials and construction techniques to enhance fire resistance. Lastly, this knowledge is crucial for the development of fire safety codes and standards, which aim to minimize the risk of collapse and ensure the safety of occupants and firefighters.

3.2.1 Compartment Fire Time Temperature Curve

This dissertation incorporates findings on the compartment fire time temperature curve from the full-scale testing at Cardington (Lennon & Moore, 2003). This research presented a study involving a series of full-scale fire tests conducted by the UK's Building Research Establishment. These tests were part of a European collaborative research program to support the development of the Natural Fire Safety Concept (NFSC), aiming to validate and refine a user-friendly design tool for fire safety in Eurocodes. This research explored the effects of various parameters like compartment linings, fire load types, and through-draft conditions on post-flashover fires, seeking to enhance the predictive methods for determining parametric temperature-time response compared to actual test results. The document delves into the intricacies of fire behavior in compartments, examining the influence of construction materials, ventilation, and fire load on temperature development during fires, ultimately providing data to validate the NFSC and inform fire safety design standards in Europe (Lennon & Moore, 2003).

The compartment fire time-temperature curve from the Cardington tests was validated by the fire tests conducted in Phoenix, Arizona (Lennon & Moore, 2003). These tests aimed to evaluate the predictability of structural collapse. Notably, the maximum temperatures in living and bedroom areas in Phoenix ranged between 540 °C and 815 °C, lower than the over 1000 °C observed in the UK's Building Research Establishment tests. In Phoenix, the peak temperature was reached in approximately 10 minutes, in contrast to 20 to 40 minutes in Cardington. This discrepancy could be attributed to differences in fire load, fire load distribution, ventilation, structural materials, local temperature, and weather conditions (Lennon & Moore, 2003). However, the overall fire behavior pattern remains similar, marked by a rapid initial increase, a sustained high-temperature phase, and a gradual cooling phase. Figure 3.5 shows the experimental time-temperature curves from the previous tests, the fire curve from Eurocode 3, and the standard fire curve. The fire curve from EC 3 and the experiments all consist the cooling down phase. On the other hand, the yellow curve, which is the standard fire curve, does not have the cooling down phase.

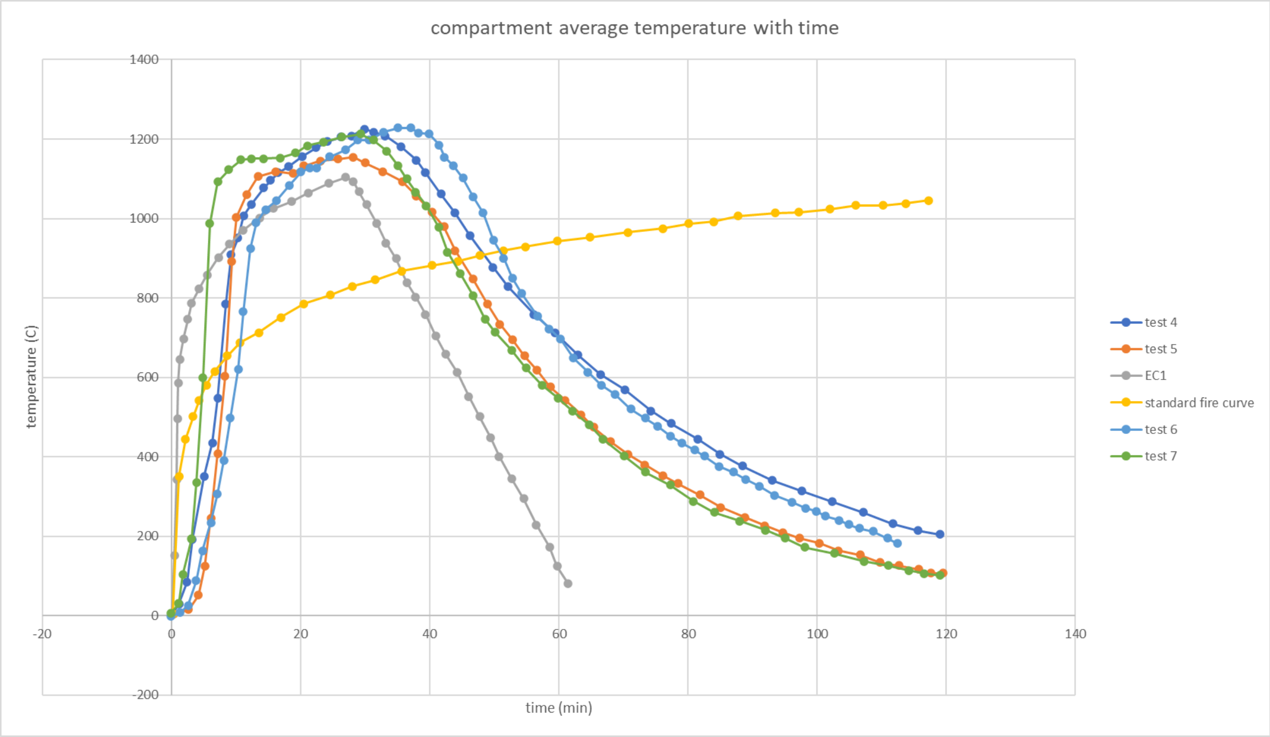


Figure 3.5 – Average Compartment temperature between different experiments and the ISO curve from EC1. (Adapted from Lennon & Moore, 2003)

3.2.1.1 Eurocode Parametric Fire Curve

In this dissertation, the standard fire curve is utilized, as there is no cooling down phase in this model. The provision of nominal temperature-time curves by Eurocode is not intended to represent a real fire and should be considered conventional. Out of the three different nominal temperature-time curves proposed by Eurocode 1, only the standard temperature-time curve is used here, due to its historical utilization and its ongoing use today in standard fire tests for estimating the heating rate of structural elements. Furthermore, according to Eurocode, a fully developed fire in a compartment can be represented by the standard temperature-time curve (Franssen & Real, 2015). This curve is often referred to as the ISO curve, as its expression was derived from the ISO 834 standard as shown in Equation 3.1:

$$\theta_g = 20 + 345 \log_{10}(8t + 1) \tag{3.1}$$

where θ_g is the gas temperature in °C and t is the time in minutes. On the other hand, Chapter 3, Section 3.3 of Eurocode 3 includes a cooling down phase, wherein Eurocode 3 specifies a constant value for slope at this phase until the compartment cools back to room temperature (Franssen & Real, 2015).

3.2.2 Behavior of Different Structural Components Under Fire

3.2.2.1 Concrete Components

Concrete, known for its durability and strength, has a high level of fire resistance. This resistance stems from concrete's non-combustible nature and its ability to retain structural integrity under high temperatures for extended periods. However, the behavior of concrete in fire is complex. It can experience spalling (the breaking off of surface layers) due to thermal stresses and moisture trapped within. Additionally, prolonged exposure to high temperatures can lead to a reduction in strength. The performance of concrete in fire is complicated and depends on factors like its composition, density, and moisture content (Khoury, 2000).

Figure 3.6 shows the relationship between temperature and concrete strength reduction factor. The data points for each concrete type are plotted against the temperature, and a line of best fit is drawn through them, which helps to visualize the trend. The data points and lines for each concrete type are presented in different symbols and line styles, as indicated in the legend (Khoury, 2000; Qureshi, et al., 2020). All the types of concrete show a general trend: as temperature increases, the reduction factor decreases, indicating a reduction in some properties of the concrete. The reduction factor is greater than 1 for some types of concrete at lower temperatures, suggesting that at the beginning of the heating process, the component strength may increase due to the expansion. This is different from the steel and wood components. After the strength reaches its maximum, around 1.15 times its original strength, it will drop gradually until the strength comes down to 0 at around 1000°C. The final strength loss curve for concrete will be the black solid line and will be used to generate the probabilistic curve for concrete.

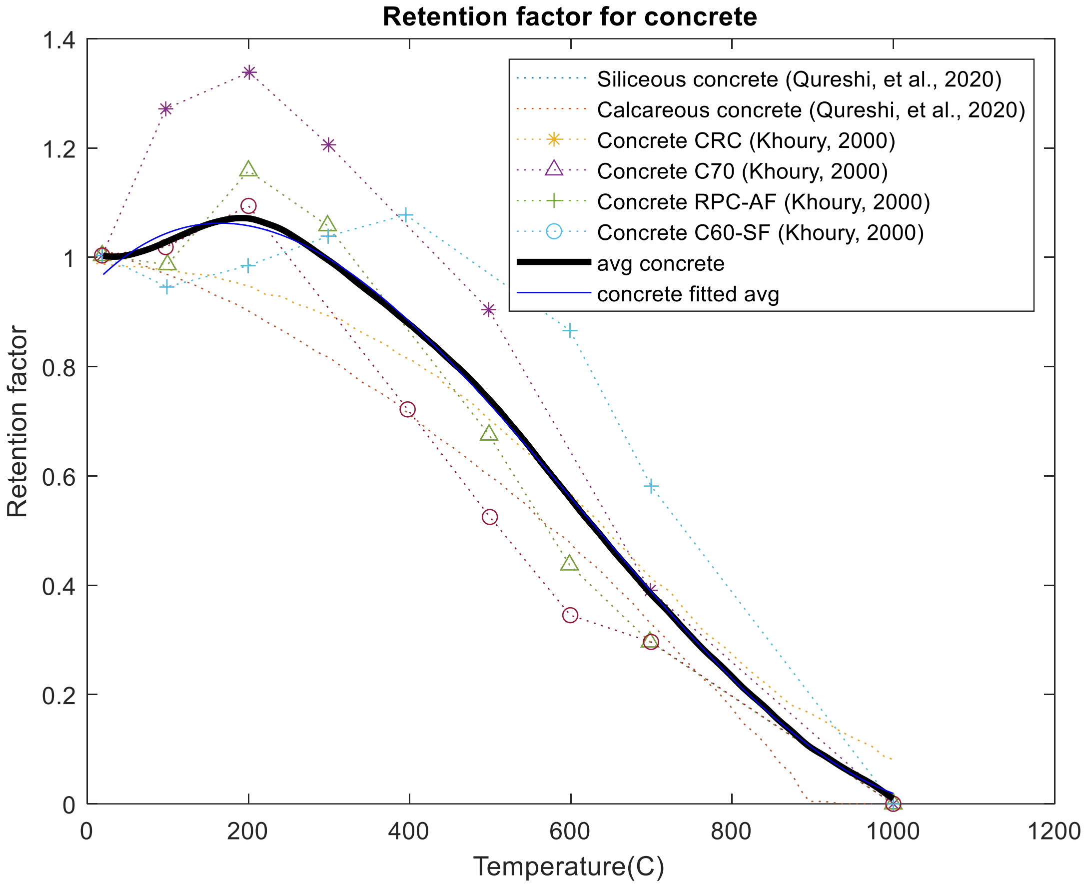


Figure 3.6 – Retention factor curve for concrete

3.2.2.2 Steel Components

Steel is a good structural material and known for its high strength-to-weight ratio and flexibility in design. However, its behavior under fire conditions is a significant concern. Steel loses strength rapidly when exposed to high temperatures, a phenomenon that can lead to structural collapse. This loss of strength typically starts to become significant at temperatures around 400°C, and by 600°C, many steel grades have lost over half their strength. To mitigate these risks, steel structures often require protective coatings or cladding to enhance their fire resistance, and their design must consider the potential for thermal expansion and distortion under fire conditions.

Figure 3.7 shows the relationship between the temperature and steel reduction factor (Choe et al., 2011; Outinen and Makelainen, 2004; Wang, et al., 2020; Eurocode3, 2015). The curve always starts at 1 means at the beginning, there is no strength loss. Different curves comes from different experiments and may be based on different assumptions, however, the overall trend is similar. Figure 3.7 also shows that at lower temperatures, there is little to no reduction in the steel’s strength. However, as the temperature rises beyond a certain point, around 600°C to 800 °C, there is a sharp decline in the reduction factor, indicating a significant loss of strength. EC3 stands for Eurocode3 and it assumes that the strength of steel will be constant before 400°C. After 400°C it will drops quickly and the slowly come down to 0 at around 1000°C. The black solid line will be the fitted curve among all the data and will be used to generate the probabilistic model for steel.

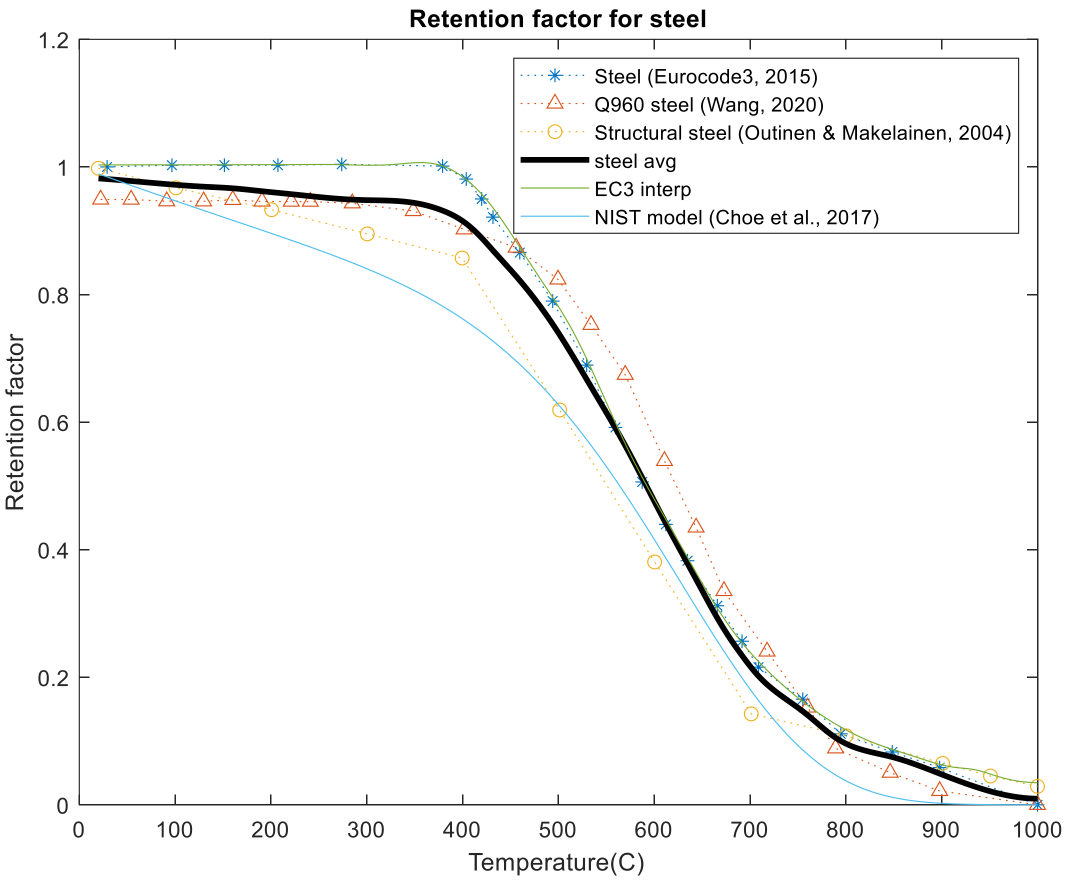


Figure 3.7 – Retention factor curve for steel

3.2.2.3 Wood Components

Wood is a combustible material, and its behavior in fire is significantly different from concrete or steel. When exposed to fire, wood is more likely to lose its structural strength due to the loss of mass. Modern engineered wood products can offer enhanced fire performance. The key to wood's behavior in fire is its rate of combustion, which depends on factors like the type of wood, its density, moisture content, and the presence of fire retardants. Fire safety in wooden structures often involves using fire-resistant treatments and incorporating design strategies to limit fire spread and duration.

There are mainly two groups of lass loss data for wood (Tang & Eickner, 1968; Janssens, 1997; Schaffer, 1984; White et al., 1993; Ostman, 1985; ASCE, 1992). The first group contains the thermogravimetric analysis of wood treated with different inorganic additives from Tang, W.K and Eickner, H.W., (1968). The second group of data comes from past experiments conducted by different teams. From Figure 3.8, it is clear that the strength of wood from past experiments drops quickly at slightly lower temperatures than those wood component with inorganic treatments. As a result, in this dissertation, two average curves from these two groups of data will be used and then find a fitted curve from the two averaged curves. Another thing that needs to be noticed is that since the data from past experiments drops quickly at lower temperatures, data after 280°C will be lost. Therefore, an assumption that the strength or the mass of the wood will be considered as 0 after 280°C. The final mass loss curve versus the temperature will be the black solid line and will be used to generate the probabilistic model for wood.

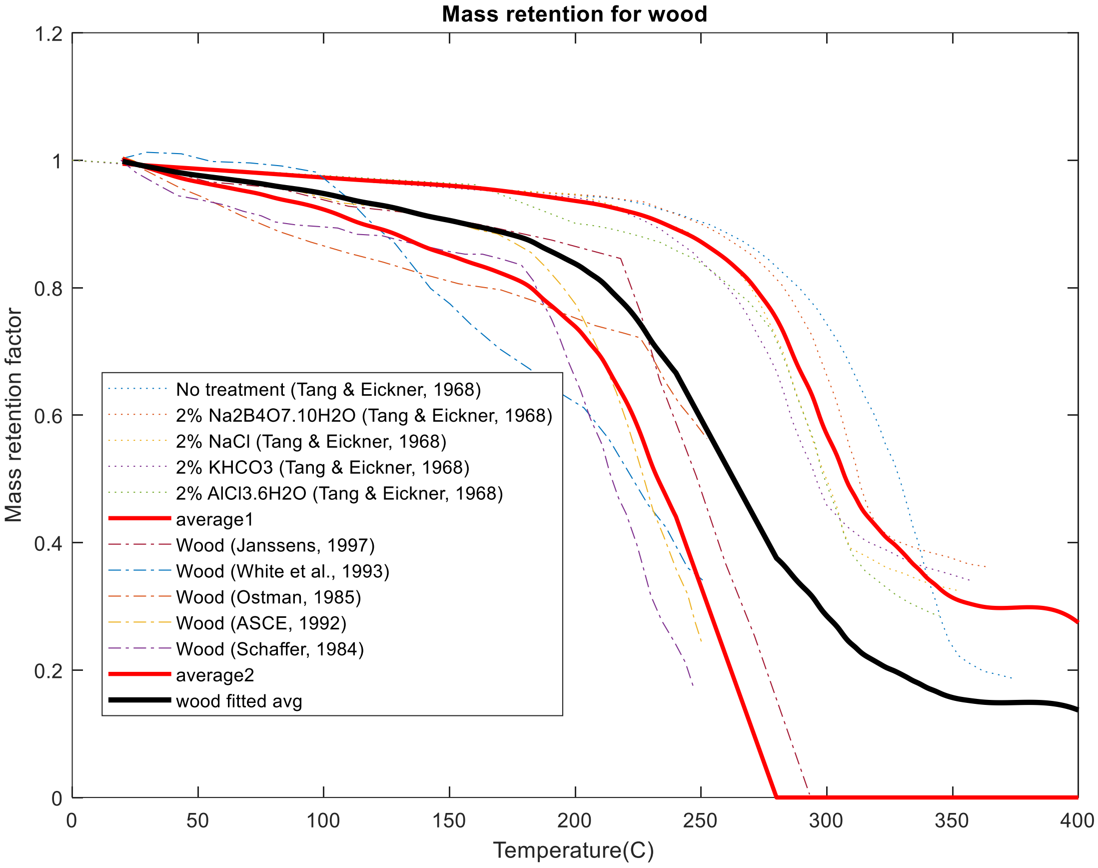


Figure 3.8 – Mass retention curve for wood

3.3 Probabilistic Models and Fragility Curve

The shift towards performance-based design in structural fire engineering emphasizes the importance of risk-informed practices that address uncertainties in extreme situations. Probabilistic analyses are essential for evaluating the effectiveness of safety measures in the design of innovative structures, supported by frameworks from previous research for probabilistic performance-based designs. These efforts aim to integrate a clear safety objective, evaluate risk and reliability, minimize damage, and consider the cost of safety measures (Qureshi, et al., 2020). However, the development of risk-informed fire safety designs is hindered by the lack of established models for analyzing thermal-structural parameters, posing a significant challenge to advancing in this field.

In this study, probabilistic models for structural components made from various materials will be generated exclusively. Temperature-dependent material strengths are identified as some of the most crucial input parameters within structural simulations. It is observed that the strength of construction materials, such as concrete and steel, may exhibit a highly scattered distribution at elevated temperatures. The strength loss curve generated in the previous section will be used to generate probabilistic models for concrete, steel, and wood components. Subsequently, a fragility curve will be generated. This fragility curve will contain only one damage state, which will indicate whether the component has yielded or not. Finally, this fragility curve for the component will be utilized to estimate the percentage of failure of the entire structure. Although the failure of a component may not represent the failure of the entire structure, further simulations can be conducted using Opensees or other thermal structural modeling software to investigate the influence of a component's failure on the entire structure. Figure 3.9 and Figure 3.10 demonstrate the three probabilistic models of temperature-specific PDFs for steel at 500°C, 550°C, 600°C, and 650°C and siliceous concrete at 400°C, 450°C, 500°C, and 650°C (Qureshi, et al., 2020).

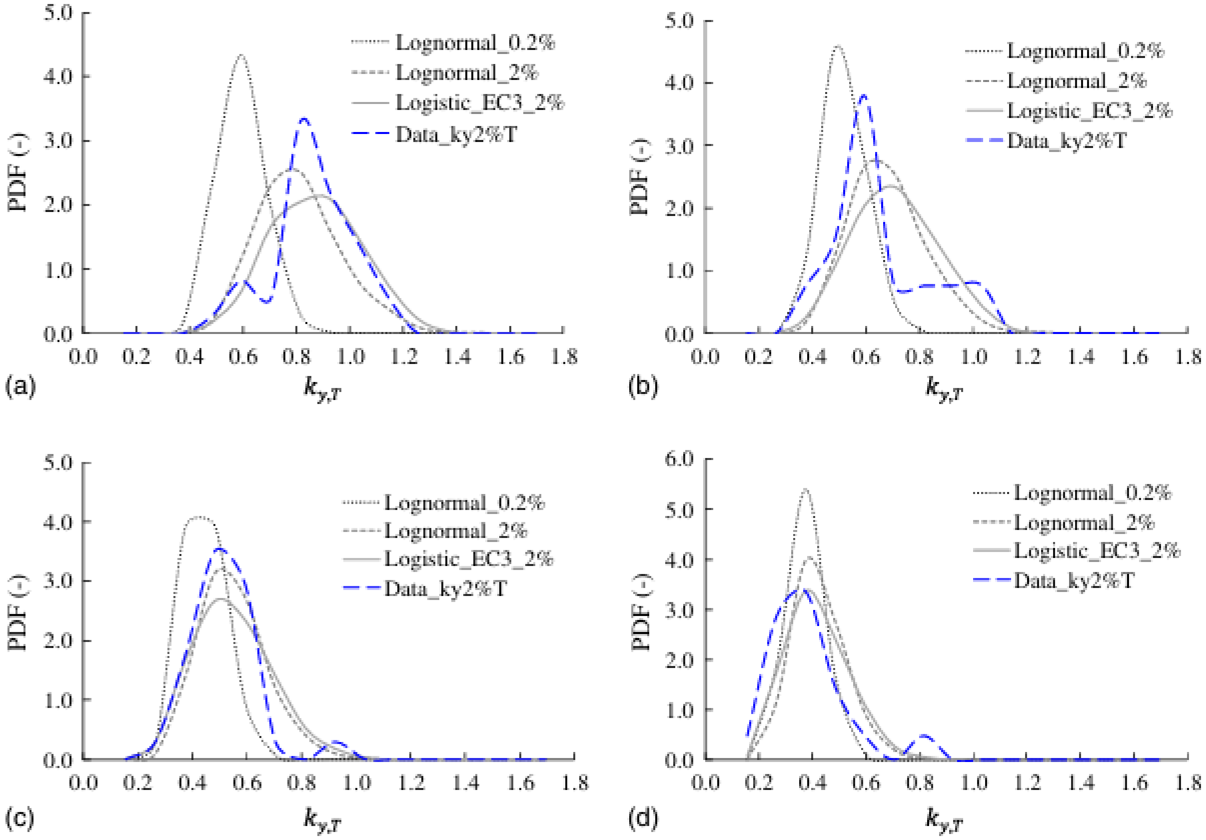


Figure 3.9 – Temperature-specific PDF for all three probabilistic models for steel at (a) 500°C, (b) 550°C, (c) 600°C, and (d) 650°C. (from Qureshi, et al., 2020)

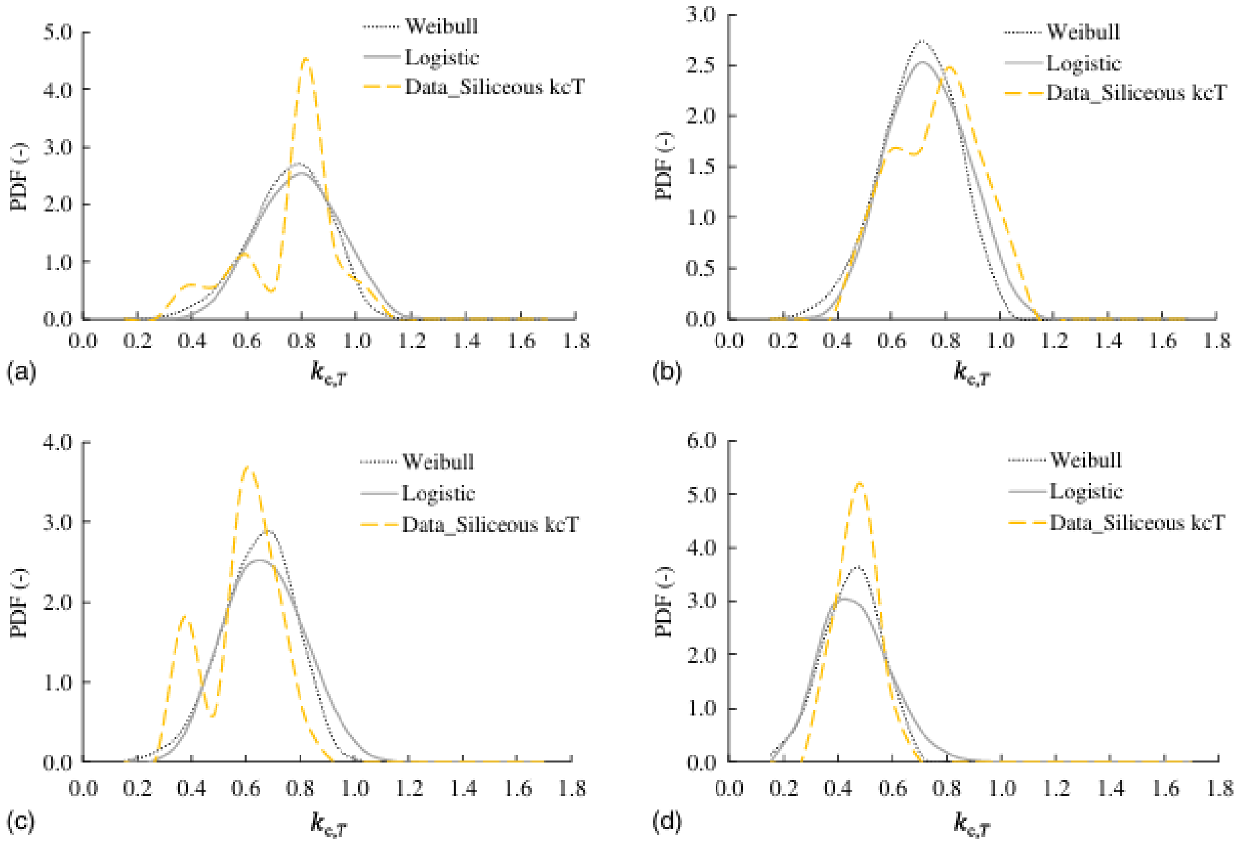


Figure 3.10 - Temperature-specific PDF for all three probabilistic models for siliceous concrete at (a) 400°C, (b) 450°C, (c) 500°C, and (d) 650°C. (from Qureshi, et al., 2020)

Figure 3.11 to Fugure 3.14 shows the probabilistic models of strength retention factor using normal distribution PDFs at each temperature for steel, siliceous concrete, calcareous and wood components. In this dissertation, the siliceous concrete and calcareous will be combined as concrete.The average strength retention curve will be used to generate the fragility curve since these two curves are similar. The two blue dashed lines represent the two standard deviation envelop and the black solid line represents the mean. It should be noted that the standard deviation decreases as the temperature increases in order to match the experimental results provided by Qureshi in 2020 (Qureshi, et al., 2020).

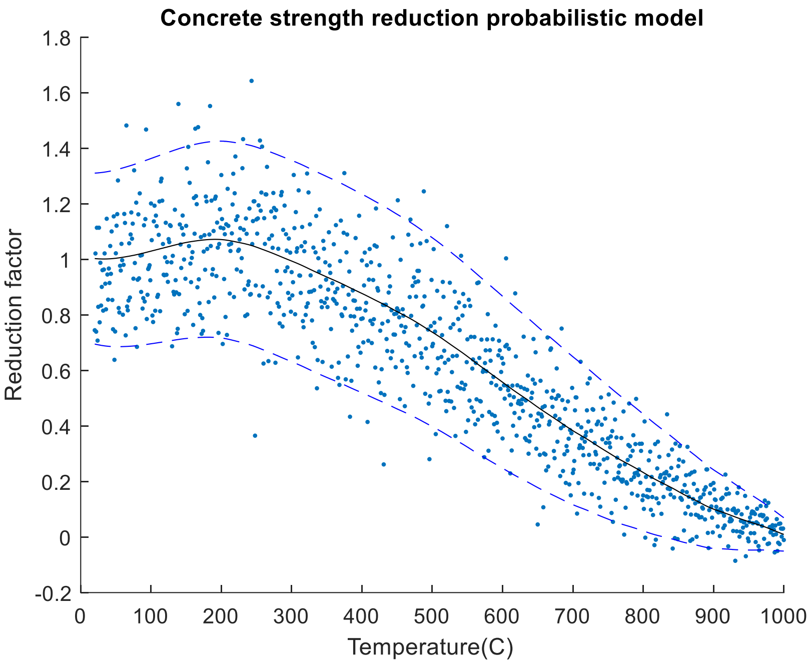


Figure 3.11 – Steel strength retention factors with normal distribution PDF

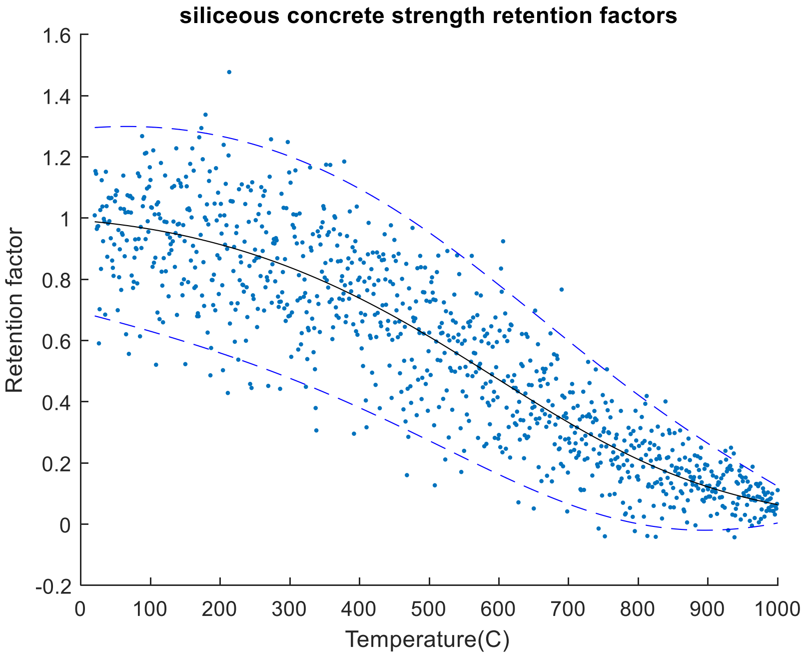


Figure 3.12 – Siliceous concrete strength retention factors with normal distribution PDF

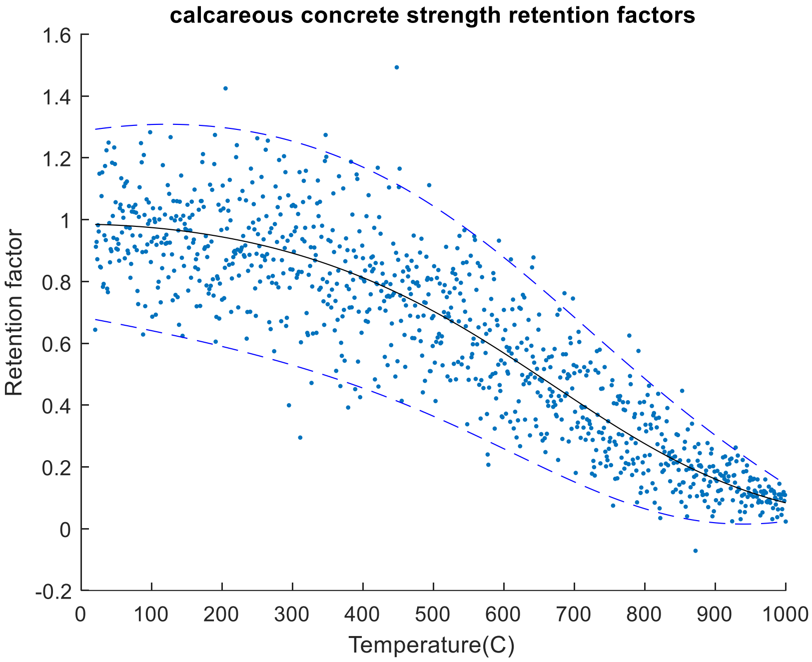


Figure 3.13 - Calcareous concrete strength retention factors with normal distribution PDF

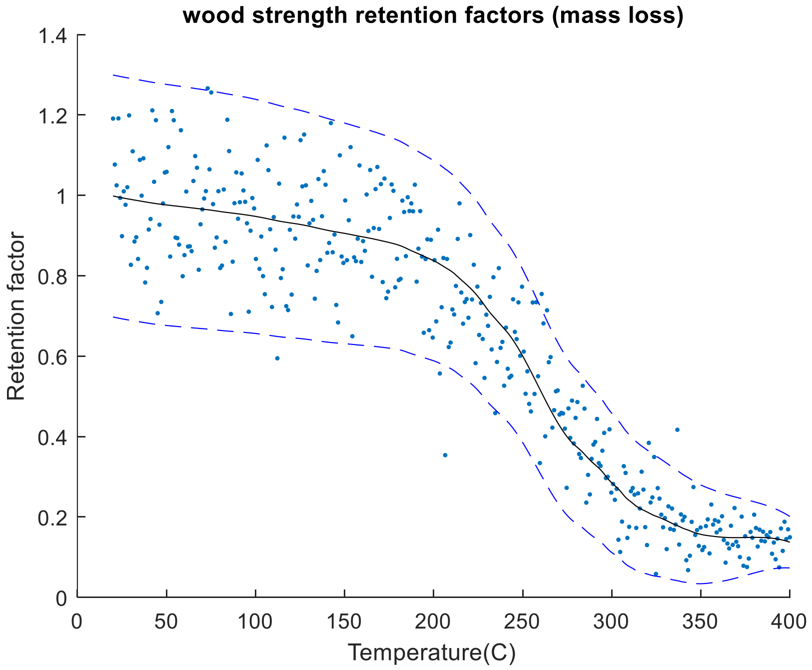


Figure 3.14 - Wood strength retention factors with normal distribution PDF

In this dissertation, the fragility curve considers only one damage state for simplification: when the component loses 50 percent of its original strength, this component is considered yielded. Figure 3.15 shows the details of how to generate the fragility curve from the probabilistic model. The horizontal line in Figure 3.15a represents 50 percent of the original strength retention line. From the probabilistic model, at each temperature, there is a normal distribution of different strength data. The process starts with integrating the normal distribution below the 50 percent line to generate the CDFs. Figure 3.15b shows the result by assembling all the CDFs into a fragility curve. For example, The CDF at the first data point is zero, that is no distribution is below the 0.5 strength retention line. So when it comes to the fragility curve, it will not fail at all and the probability of failure is zero. For the last data point, it's completely below the 0.5 strength retention line, so when it comes to the fragility curve, it's a hundred percent it will fail.

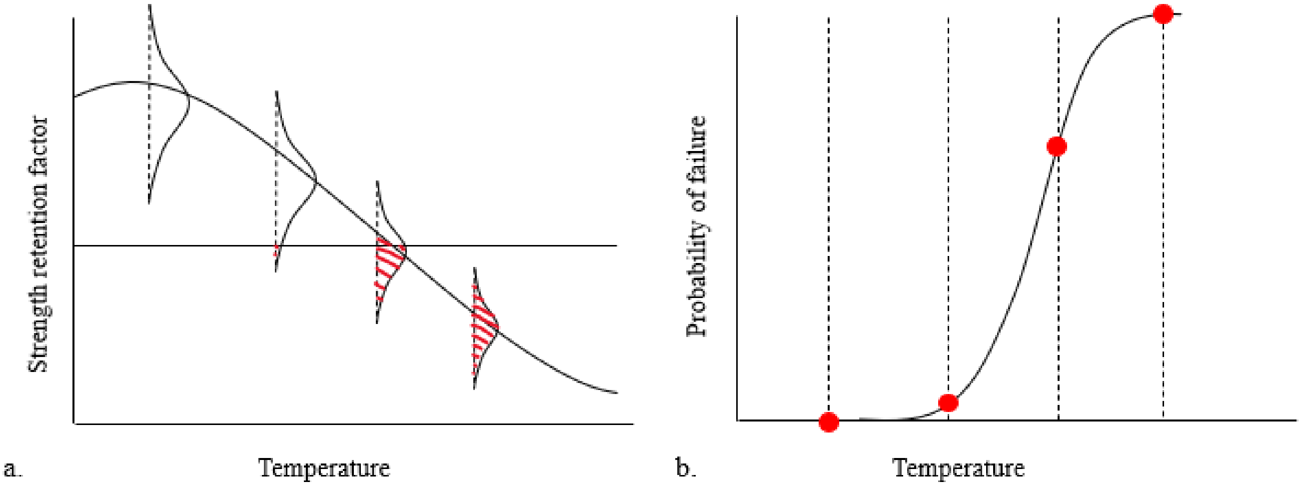


Figure 3.15 – **From probabilistic model to fragility curve**. a) Integrating at each temperature below 0.5 of the original strength; b) Generating CDFs and assembling them into a fragility curve

This model allows users to adjust the damage state, thereby enabling a more detailed assessment of structural damage. Wooden materials differ significantly from steel and concrete. Typically, compartment fires do not reach temperatures high enough to melt steel or concrete. However, a temperature of 250°C (482°F) is sufficient for the autoignition of wood materials. At this temperature, the strength of wood components decreases dramatically and results in a sharp increase in its fragility curve. The yielding temperature for steel and concrete components ranges from 600°C to 700°C. In this research, a normal distribution is used to generate the probabilistic model at each temperature for these three materials as shown in Figure 3.16. This is compared with the steel fragility curve generated by Qureshi shown in Figure 3.17 (Qureshi, et al., 2020). The fragility curves developed in this study and from Qureshi both has a yielding temperature of around 660°C, validating the appropriateness of using a normal distribution for this assumption.

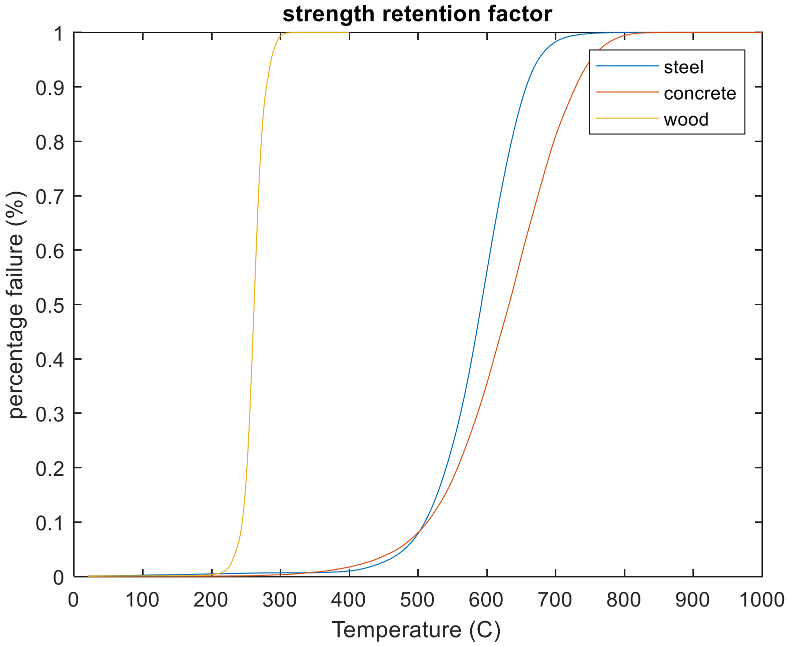


Figure 3.16 – Steel, concrete and wood material fragility curves with one damage state

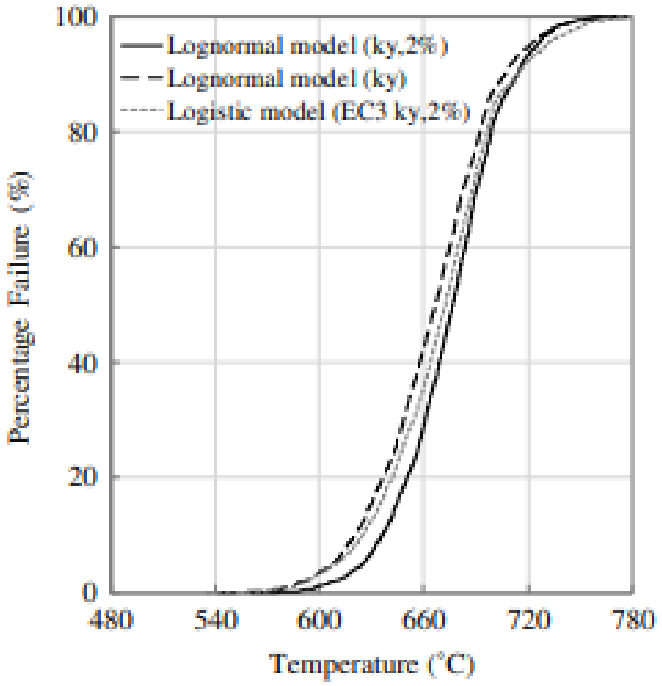


Figure 3.17 – Comparison with steel fragility curve. (from Qureshi, et al., 2020)

3.4 Probabilistic Treatment

As demonstrated above, from the fire spread simulation to the probabilistic model of strength loss for structural components, a significant amount of uncertainty and variability is involved. Therefore, this study incorporates a series of probabilistic analyses to account for these uncertainties and variabilities. In the fire spread simulation part, wind speed and direction are randomly initialized at all nodes, resulting in variability in the ignition time at the affected nodes. The subsequent structural damage assessment is based on the ignition time at the corresponding nodes, which is derived from the time-to-temperature relationship. Monte Carlo Simulation is used to perform multiple iterations, and the average ignition time among all nodes is used for the case study. Although the initial assumption of using Monte Carlo Simulation was to consider the maximum affected area and ignition time, thereby overestimating the results under fire conditions, overestimation is more realistic for fire management strategies. However, for case validation, the average results are deemed more reasonable.

The next step is to generate the probabilistic model for structural strength loss. This part of the study investigates the relationship between temperature and the strength loss of structural components. The fragility curve is generated by integrating probabilistic distribution functions from the strength loss model. Finally, the probabilistic repair cost for structural components is calculated using the initial build cost and the probability of failure. This comprehensive approach ensures that both the fire spread dynamics and the resulting structural impacts are accurately represented, providing valuable insights for fire management and mitigation strategies.

3.5 Concluding Remarks

In conclusion, the research delineated within this dissertation significantly advances the field of fire safety engineering by intricately studying the structural behavior under fire conditions. Through the integration of the fire elliptical growth model with Huygens' Principle in a grid-based simulation, an enhanced understanding of wildfire spread and its impact on structural integrity is achieved. This approach not only improves the accuracy of predictions concerning fire spread rate, direction, intensity, and area coverage but also offers valuable insights into the design and selection of materials for fire resistance and the development of safety codes and standards.

The incorporation of the compartment fire time-temperature curve and the Eurocode parametric fire curve further enriches this study, providing a robust framework for analyzing the thermal response of various materials, such as concrete, steel, and wood, under fire exposure. These analyses contribute to a nuanced comprehension of material behavior at elevated temperatures, guiding the selection of construction techniques that enhance structural resilience against fire.

Moreover, the development of probabilistic models and the introduction of a fragility curve represent a significant stride towards performance-based design in structural fire engineering. By acknowledging the inherent uncertainties in material strength at high temperatures and integrating risk-informed practices, this research paves the way for more effective and efficient fire safety measures.

Ultimately, this dissertation not only furthers the theoretical understanding of fire dynamics and structural response but also lays down a practical foundation for safer, more fire-resilient building designs. By bridging the gap between empirical data and mathematical modeling, it sets a new benchmark for predictive accuracy in fire safety engineering, underscoring the critical importance of innovative, evidence-based approaches in enhancing structural integrity and occupant safety during fire incidents.

Although this research delved into fire spreading at the wildland-urban interface and subsequent structural damage assessment, it only accounted for the phase of increasing temperature. In reality, fuel will eventually be consumed, leading to a decrease in temperature. This study does not encompass the cooling down phase for structural components. Therefore, future research could explore this aspect to attain more accurate results in damage assessment for structures exposed to fire.

4. MODEL DEVELOPMENT

This chapter details the development of the Matlab model, starting with the input parameters sourced from Excel. It then delves into the model structure, covering its definition, simulation process, and post-analysis for cost estimation. This chapter concludes by discussing the model's capabilities, limitations, and suggestions for future improvement. Figure 4.1 shows the framework for the Matlab model structure developed in this study. The Matlab source code is included in Appendix A.

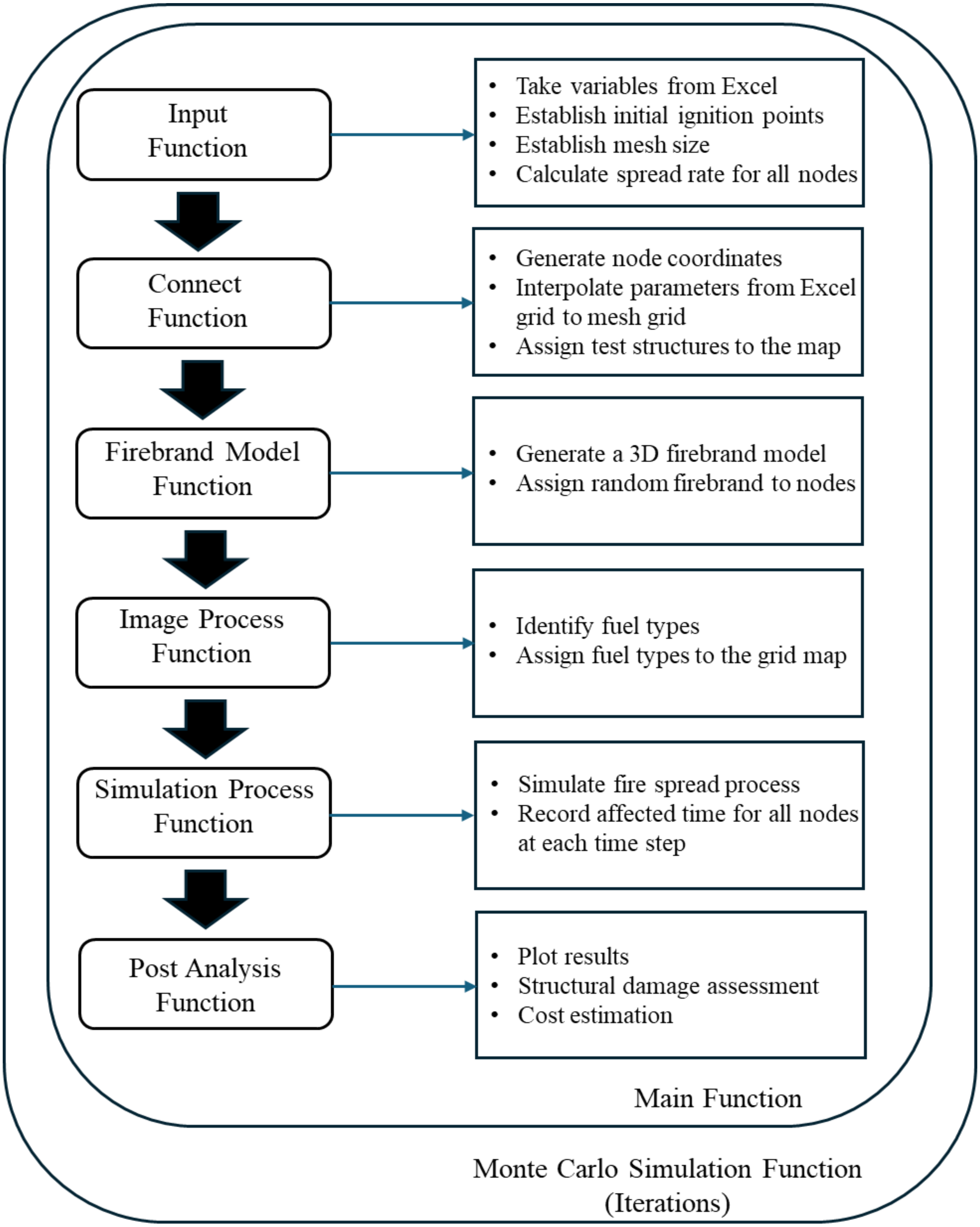


Figure 4.1 – Framework for Matlab model structure

4.1 Input Excel Variable Definition

Table 4.1 shows the basic input parameters for this Matlab model. The variables are compared with the original parameters used in Rothermel's model and the descriptions (Rothermel, 1972).

Table 4.1 – Comparison of notations between Rothermel's model and Matlab model

Type	Notation in Rothermel's model	Notation in Matlab model	Description
Fuel Particle	h	*h*	Low heat content $(\frac{Btu}{lb})$
	S_T	*S_T*	Total mineral content (fraction)
	S_e	*S_e*	Effective mineral content (fraction)
	ρ_p	*rho_p*	Oven-dry particle density $(\frac{lb}{ft^3})$
Fuel array	σ	*sigma*	Surface-area-to-volume ratio $(\frac{ft^2}{ft^3})$
	w_0	*w0*	Oven-dry fuel load $(\frac{lb}{ft^2})$
	δ	*delta*	Fuel bed depth (ft)
	M_x	*M_x*	Dead fuel moisture of extinction (fraction)
Environmental input	M_f	*M_f*	Moisture content (fraction)
	U	*U_wind*	Wind velocity at midflame height $(\frac{ft}{min})$
	$\tan\phi$	*steepness*	Slope steepness (fraction)
Other		*slope*	Wind direction ($radian$)

The Monte-Carlo simulation approach is also employed in this model, which involves random initialization and is repeated multiple times to obtain the final results. However, since the fuel information is set once the map is read, variables related to fuel and geometry will remain unchanged. The only aspects subject to variation are the environmental parameters, in this case, the wind velocity and direction. As outlined in Section 2.1.1, a random range will be used for the following parameters in this model. The low heat content, denoted as h, is typically 8000 Btu/lb; hence, a range from 7000 Btu/lb to 9000 Btu/lb is assigned to h in this model. The total mineral content, represented by S_T, usually stands at 0.0555, and a range from 0.03 to 0.0555 will be adopted. The effective mineral content, indicated by S_e, is maintained constant at 0.01. The oven-dry particle density, represented by ρ_p, is set to a constant value of 32 lb/ft^3. The surface-area-to-volume ratio, denoted as σ, is allowed to vary from 300 ft^2/ft^3 to 1200 ft^2/ft^3. The oven-dry fuel load, $w0$, is allowed to fluctuate between 0.0205 lb/ft^2 and 0.41 lb/ft^2. The fuel bed depth, δ, is permitted to range from 1 ft to 5 ft. The dead fuel moisture of extinction, M_x, is fixed at 0.3. The moisture content, M_f, can vary from 0.02 to 0.3 (Andrews, 2018; Albini, 1976). The *steepness* and the wind factor are considered together; therefore, the initial slope steepness is set to zero for all nodes and will change as the wind factor changes. The user will define the variation range for wind velocity and direction to investigate the probability of the spread of fire. Wind direction is denoted by *slope* in this model.

It should be noted that wind direction is not incorporated into Rothermel's model. The rationale behind this omission is that the output of Rothermel’s model is scalar, lacking any directional component. However, the inclusion of wind direction is essential for the simulation being conducted. Fortunately, wind velocity data can be retrieved from the National Oceanic and Atmospheric Administration (NOAA). The detailed process for incorporating this data will be outlined in the case study section.

4.2 Model Structure

4.2.1 Main Function

The main function of this model is essentially to display the model's overall structure. Initially, inputs are taken from an Excel file by the input function, which defines the map size, grid size, and calculates the rate of spread based on Rothermel's model and the parameters, in order to generate the fire ellipse. The second function, referred to as the fun_connect function, is responsible for generating node connections, creating a mesh, assigning structures to the map for testing, and interpolating all parameters in accordance with the map size and mesh size.

Next step is to determine the firebrand matrix and add the firebrand distance to the ellipse parameter. After that is the imageprocess function. This function scans the image and updates the node properties based on the node type. The process involves pixel differentiation, though in the future, machine learning can be employed for image segmentation and fuel type identification. Users have the option to define barriers, allowing them to observe the impact on fire spread. Variables are saved in the format 'filename.mat', allowing them to be utilized for calculations in Python, although the output results will continue to be generated in MATLAB. After the node properties have been assigned, the firefront at every time step is calculated and stored by the fun_process3 function, which also determines the nodes affected and the time of impact for assessing structural damage and plotting the affected area.

The next function is the fun_postresults function. Several outputs related to the fire simulation results are generated by this function. An animation that showcases the fire spread and the changing fire front over time is created. Furthermore, the time-temperature curve for concrete, steel, and wood components is used to generate the probabilistic model and fragility curves. Finally, the percentage failure for each structure is outputted, and the economic loss for the affected area is determined.

4.2.2 Input Function

The input function will first have the map dimension defined with the unit of feet by the user. The entire duration time of the fire spread can also be defined by the user. Additionally, the user can specify the dt for each time step. While the default units will be in feet and minutes, this model is also capable of accepting inputs in hours and miles, which will then be converted into feet and minutes. The next step involves the definition of the grid size and mesh size by the user. It is assumed that the input data will have a surveying grid distance, referred to as excel distance in the model. However, during the simulation, the meshed grid map will be used for calculation by the model. In other words, the meshed node properties will be interpolated from the real grid map by the model. The user is allowed to determine the size of the meshed grid. The accuracy of the simulation will increase, but the running time will be longer if the grid size is small. Conversely, the simulation will be less detailed but will require less time if the grid size is large. In addition to that, the meshed grid size could be either larger or smaller than the excel grid size. Below is an example of the excel grid and meshed grid on a same image.

There are limitations when it comes to spatially correlated data. For example, the interpolation of fuel data is not feasible in reality because point-based fuel measurements cannot be interpolated or extrapolated, particularly in urban areas. This problem can be resolved when the surveying grid matches the mesh grid, eliminating the need for interpolation. Besides fuel data, spatial geological data may also be affected by the viewing angle and the altitude of the local topography. Figure 4.2 shows an example of the difference between the surveying grid (Excel grid) and the meshed grid. The yellow line represents the real surveying grids and the white line represents the meshed grids. Generally, the surveying grid will be larger than the meshed grid. These aspects may require further investigation to improve the model and enhance its accuracy.

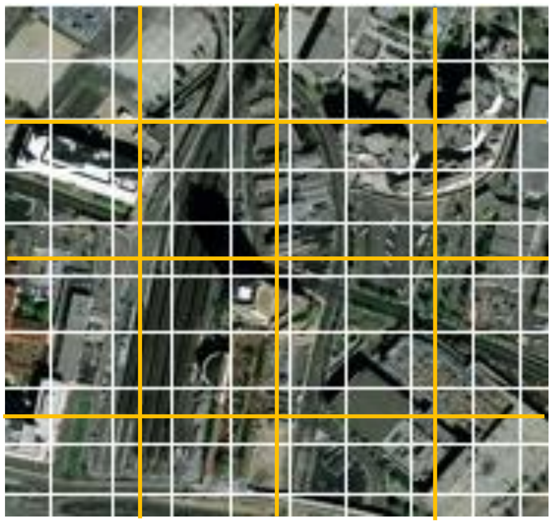


Figure 4.2 – **Example of the difference between Excel grid and meshed grid**. The yellow line represents the excel grids and the white line represents the meshed grids.

After the dimension has been defined, the function also permits the user to specify the initial ignition points, and the model is capable of handling multiple initial ignition points. However, it should be noted that as the number of initial points increases, the running time will increase exponentially. The next step involves the acceptance of input parameters from the excel and the calculation of the spread rate at each node. The spread rate is calculated using the v_cal function. The final task in the input function is the calculation of the ellipse parameters at each node, based on the spread rate at that node. This model operates under the assumption that the wind direction and velocity will remain constant during the simulation, even though this may not be realistic. The rationale behind this assumption is that recalculating the spread rate and the associated ellipse parameters during the simulation would lead to a significant increase in running time. It is anticipated that future technological advancements may significantly reduce running time, thereby making it feasible to implement changes to parameters during the simulation. Besides the parameters used to calculate the spread rate and direction, this model also incorporates a random floor variable for structures. This variable is used to estimate the economic loss of a structure and then used to estimate the entire loss of the area under a fire incident.

4.2.3 Connect Function

This function first generates the node coordinates, defining [0,0] at the bottom left corner and [m,n] at the top right corner. The coordinates are stored in the variable 'node'. The next step is to generate the connectivity array. This means that given one node, the user can quickly find the other 4 nodes surrounding it. If this node is at a corner or edge, NaN will be used to indicate the absence of neighboring nodes in that direction. Please note that this connectivity array is not used in the current simulation process; it was developed for 'fun_process1' and is saved for future use. Following the connectivity array, this function interpolates parameters from the Excel file and fits them into the meshed map for later calculations. The interpolated parameters include spread rate, wind velocity, wind slope, and three parameters used to generate ellipses. Although using directly calculated values could yield more accurate results, it would introduce more variables requiring interpolation. All NaN values in the data are converted into 0 for calculation purposes. Finally, all the variables including the coordinates ellipse parameters, wind velocity, spread rate and node type will be stored in the variable called nodeproperty. It is noted that the node type at current step is set to the default and will be modified in the image process function.

This function also outputs the vector plot of the wind velocity. This vector plot will be used to compare with the final fire spread map. Figure 4.3 shows an example of the vector plot with random wind velocity and direction at each node.

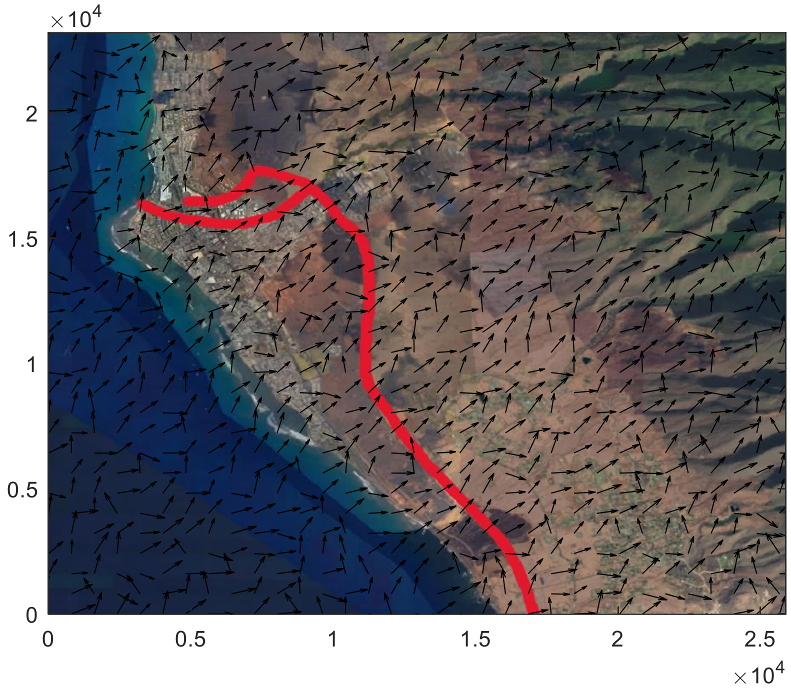


Figure 4.3 – Example of the wind vector plot with random wind velocity

In addition to generating the connectivity array, interpolating parameters and generating the wind velocity vector plot, this function also permits the user to manually modify the node type for testing purposes. For instance, the node type can be changed to a structure to observe the damage resulting from the fire spread; similarly, the node type can be altered to a barrier to examine how the fire barrier will alter the shape of the fire spread. This model is designed to permit the user to alter the initial cost of wildland, unoccupied land, and the unit cost for various types of structures. In this dissertation's example, it is shown that the unit cost for concrete structures can range from $400 to $600 per square foot, for steel structures from $300 to $500 per square foot, and for wood structures from $200 to $400 per square foot. In addition to the unit costs, the model also enables the user to modify the coverage area for different types of structures. Consequently, the total initial cost is calculated by multiplying the total area by the unit cost and then by the number of floors each structure possesses.

4.2.4 Firebrand Model

The implementation of the firebrand model is mainly to account for the effect of spot fire. The main idea of this firebrand model is to add an additional distance to the original elliptical fire spread region. The firebrand model is generated based on the main idea of creating a 3-D spread model. In this model, the x and y axes represent the distances in two directions, while the z-axis represents the probability for each firebrand to spread. This 3-D representation allows for a comprehensive understanding of the firebrand spread probabilities in different directions. The x-axis is defined with limits ranging from 0m to 10m, assuming the firebrand starts from 0 and can travel up to a distance of 10 meters with user-defined units. The y-axis is defined with limits ranging from -5m to 5m and 0 is the position where the firebrand is generated. A 3-D surface plot is generated by multiplying the mesh. Figure 4.4 shows the side views of the firebrand model adapted from Himoto and Tanaka (Himoto & Tanaka, 2008). Figure 4.5 shows the 3-D possibility of the firebrand model generated in this study. Figure 4.6 is the sketch diagram of the probability of firebrand scattering released from a fire-involved building developed by Himoto and Tanaka in 2008 and it’s similar to the 3-D firebrand model generated in this study (Himoto & Tanaka, 2008).

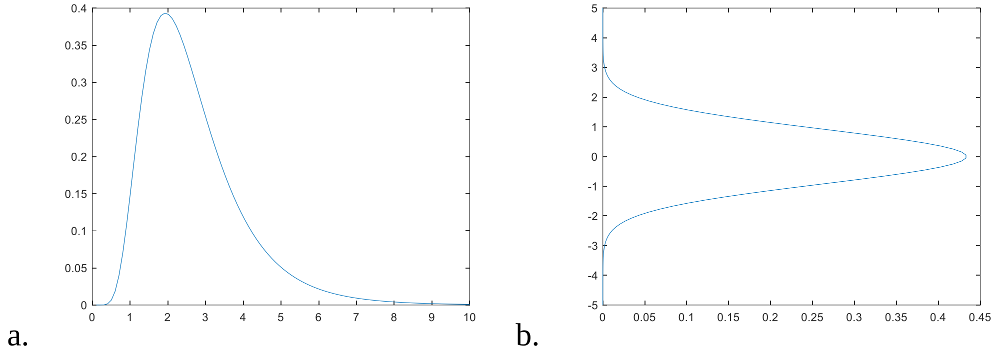


Figure 4.4 – **Side views of the firebrand model**. a) The possibility for firebrand reaching in x direction; b) The possibility for firebrand reaching in y direction. (Adapted from Himoto & Tanaka, 2008)

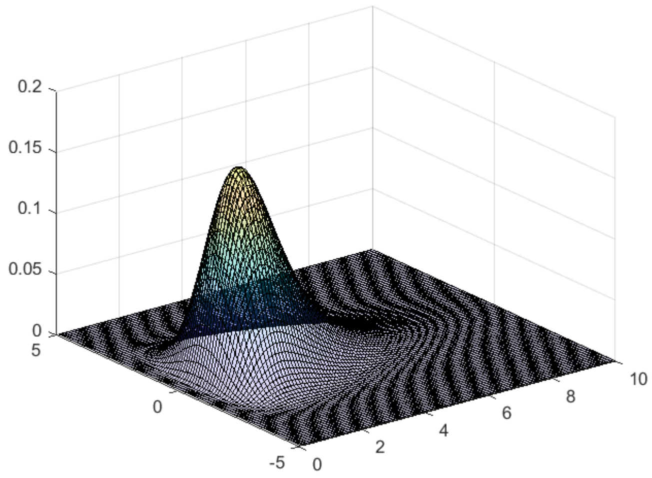


Figure 4.5 – Visualization of the joint possibility of firebrand

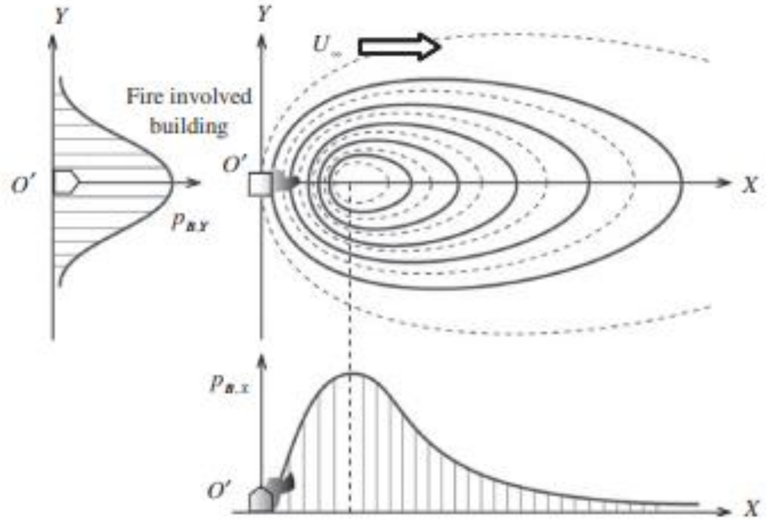


Figure 4.6 – Probability of firebrand scattering released from a fire-involved building. (from Himoto & Tanaka, 2008)

The next step is to generate random points and simulate this model. The probability is determined based on the x and y axes, integrating the unit volume under the surface. However, this approach would take much longer as the integration needs to be performed at every step. More than 10,000 points are generated and randomly distributed in the given space. Every point below the surface is marked in red, while all other points are marked in blue. Consequently, when a random point is selected, its x and y coordinates are used as additional distances. This method is similar to a 3-D distribution model. Figure 4.7 demonstrates the 3-D firebrand model with randomly assigned points. The output of this function provides the x and y distances at a random point.

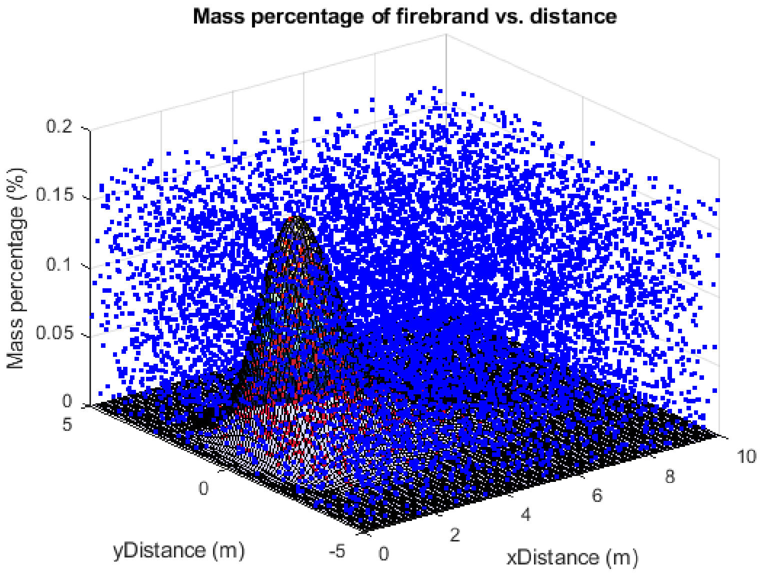


Figure 4.7 – 3D model for firebrands. During each iteration, this model will output one firebrand position for each node

It should be noted that the distance is included and treated as ignition in this method. Nevertheless, firebrands can only cause point ignition, and the distance in between should not be affected. According to section 3.1.2, it is assumed that the distance is relatively small compared to the rate of fire spread, and the gap will be encompassed within a single time step. This assumption is based on the likelihood that the distance a firebrand can travel will most likely not exceed 8 meters or 26.2 feet, which is significantly less than the fire spread rate within one time step.

4.2.5 Image Process Function

The image process function is designed to assist with fuel identification. It starts with scanning the image and updating the node properties based on the node type. The process involves pixel differentiation, though in the future, machine learning can be employed for image segmentation and fuel type identification. Users have the option to define barriers, allowing them to observe the impact on fire spread. Currently, this model accepts images with 1(1 x 1) and 9 (3x3) sub-images. This division is necessary because fuel identification requires a high-resolution image, which can be quite large. As a result, processing such images could involve over 1,000,000 nodes, leading to a significant increase in time costs. Figure 4.8 shows an example of the image process function with 9 sub-images and Figure 4.9 shows the ensembled image with 9 sub-images. The variables '*m1*' and '*n1*' in the source code represent the number of nodes in the x and y directions for each sub-map, and 'nnodes1' represents the total number of nodes in each sub-map. Figure 4.10 shows the image process function result with one sub-image, the result is similar to the result from 9 sub-images. It can be noticed that there is a small gap at the edge of adjacent sub-images. The reason might be that when using the Matlab built-in function to do the pixel identification, it will automatically ignore the edge of the image to avoid errors. The next step is to create an empty matrix to save the coordinates for different fuel types. Currently, only water and barriers are identified at this stage. It should be noted that if more fuel types need to be identified in the future, additional empty matrices will need to be created. Finally, this function is going to run through each sub-image to get the node number and coordinates for each fuel type; and then assemble them into the original image.

Furthermore, this function will also read the image using the imread MATLAB built-in function. It is necessary to convert the image resolution into the mesh size for a sub-image to ensure the number of nodes remains consistent. It is important to note that MATLAB reads the image from the left top corner to the right bottom corner (upside down), which is inconsistent with the coordinates previously defined by this model; therefore, the conversion of the y-axis to an upside-down orientation is required.

The next step involves using another MATLAB built-in function to identify the coordinates for different fuel types, utilizing the Color Thresholder tool within the APPS menu, opting for HSV over RGB to avoid confusion caused by similar colors with varying densities. The primary objective of this function is to create a mask over the selected color. The coordinates must also be inverted, as this built-in function operates using image coordinates. In this context, red is used to denote water, and black is used to signify a barrier.

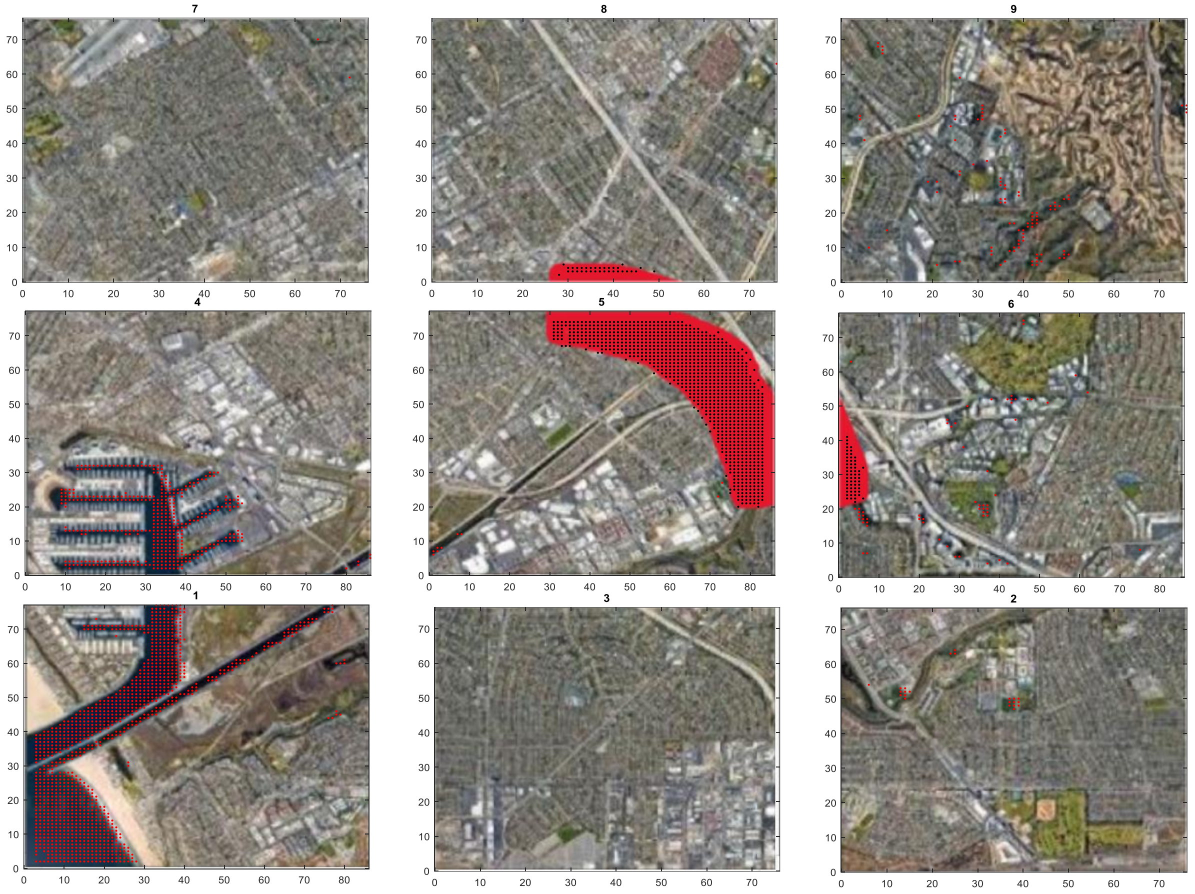


Figure 4.8 – Example of the image process with 9 sub-images

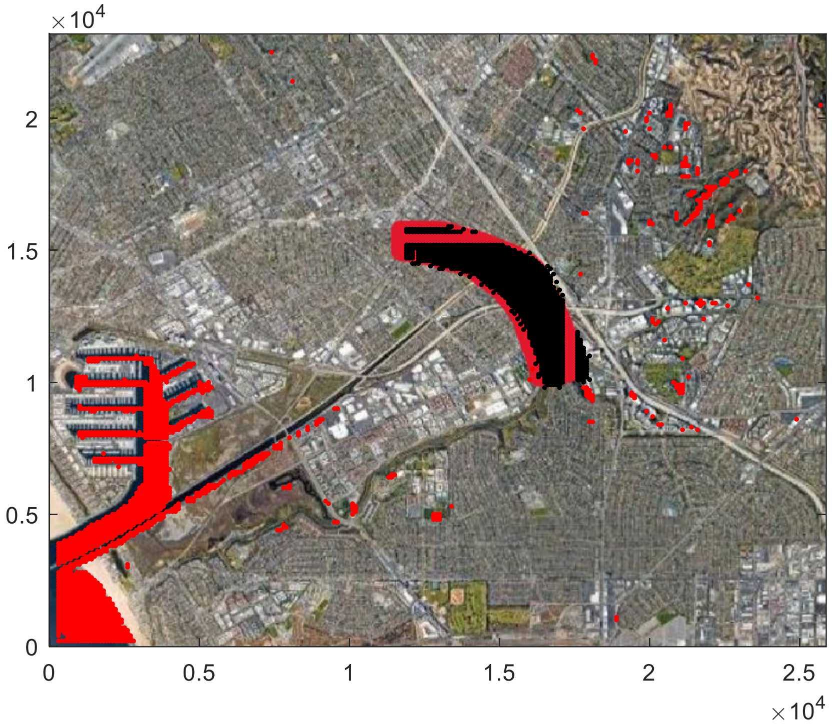


Figure 4.9 – Example of the ensembled image using the 9 sub-images

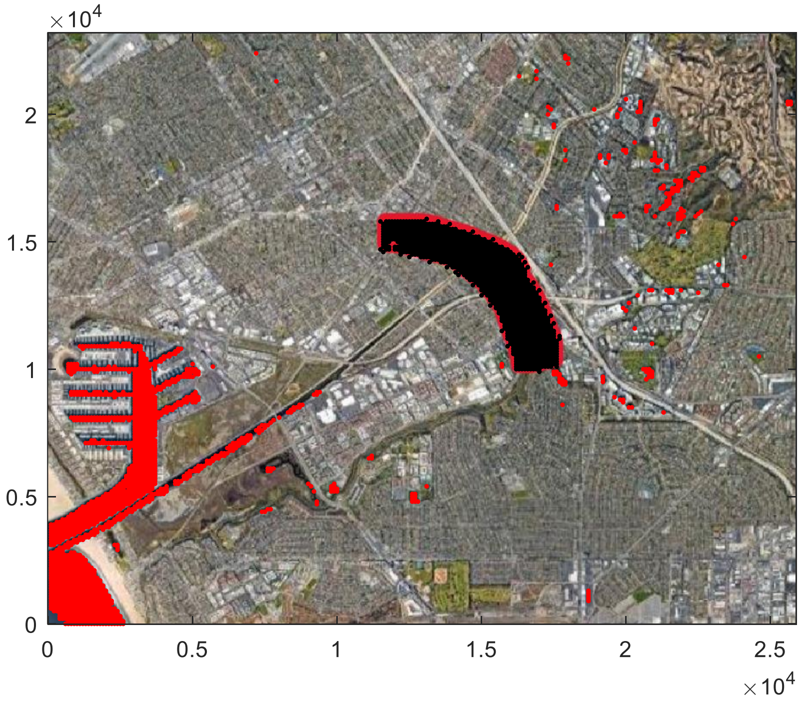


Figure 4.10 - Example of image process with 1 sub-image

4.2.6 Simulation Process Function

This function is identified as the most crucial within the model. The elliptical fire spread range is calculated for each node at the fire front. As outlined in section 2.2.4, a series of small ellipses are generated at every time step, and the connection of the outermost vertices from all ellipses forms the final spread range at that time step. The parameters for these ellipses are calculated by the input function, indicating that the parameters are established at the start of the simulation and remain unchanged over time. This series of elliptical vertices is considered to represent the fire front, and these vertices serve as the centers for small ellipses in the subsequent time step. In addition, all nodes encompassed by the ellipses are deemed affected nodes, and the duration of time affected by the fire continues to increase until the simulation concludes.

It should be noted that there may be multiple ignition points at the beginning, and in this model, the fire spread process will be simulated separately. This is necessary when simulating scenarios with multiple ignition locations; otherwise, the boundary at each time step may be affected by different ignition locations. The distance between the initial ignition points might be large, and the area between the initial points will not get burnt. In addition, Once the two fire ranges overlap each other, the node will still be marked as affected, but the affected time will be recalculated. Figure 4.11 shows the comparison of whether distinguishing the initial ignition points. Figure 4.11a shows there is no distinguishment between initial ignition points and Figure 4.11b shows there is distinguishment between initial ignition points.

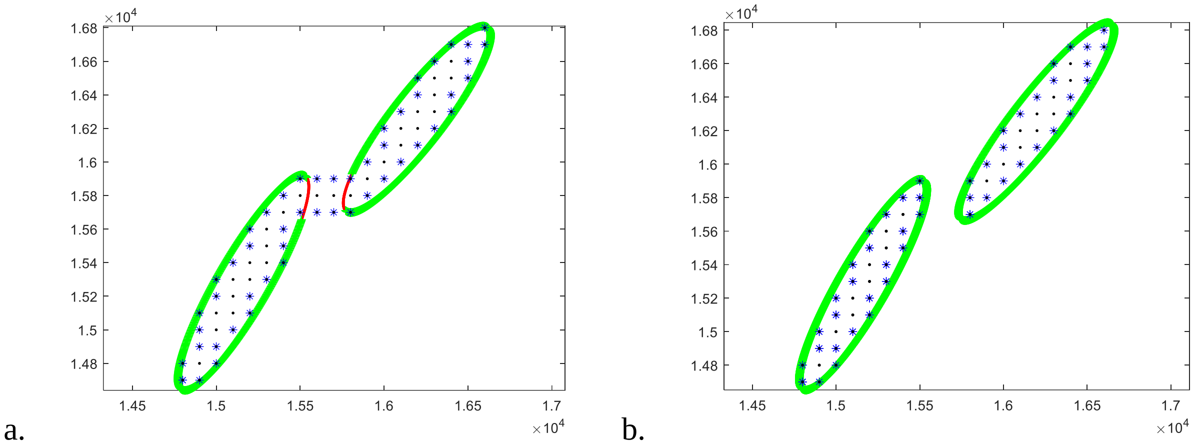


Figure 4.11 – **Comparison of whether distinguishing the initial ignition points**. a) No distinguishment between initial ignition points; b) Distinguishment between initial ignition points

The first iteration of the initial ignition points will also b separated from the subsequent time step. This distinction is crucial to ensure that the ellipse distinguishes between the different ignition points. During the spread simulation and when the fire front encounters a node type of water or fire barrier, the spread rate will be changed to zero and therefore, no ellipse will be generated at that node. As the time step increases, the time at the affected nodes will be added with one time step and will be stored in the results matrix.

Figure 4.12 shows an example of a fire spread simulation during the process without water or fire barriers. By considering only one ignition point, the convolved black nodes on the grid are considered affected. The red ellipses are generated from each vertex at the last time step. The green line represents the actual firefront. The blue stars represent the vertices considered by this model, according to Huygens' Principle, and will be used to generate ellipses for the next time step. If the mesh size is small enough, the blue stars will closely approximate the green line. The final output of this function will include the index and coordinates of the fire front, affected nodes, and the duration the affected nodes have been exposed.

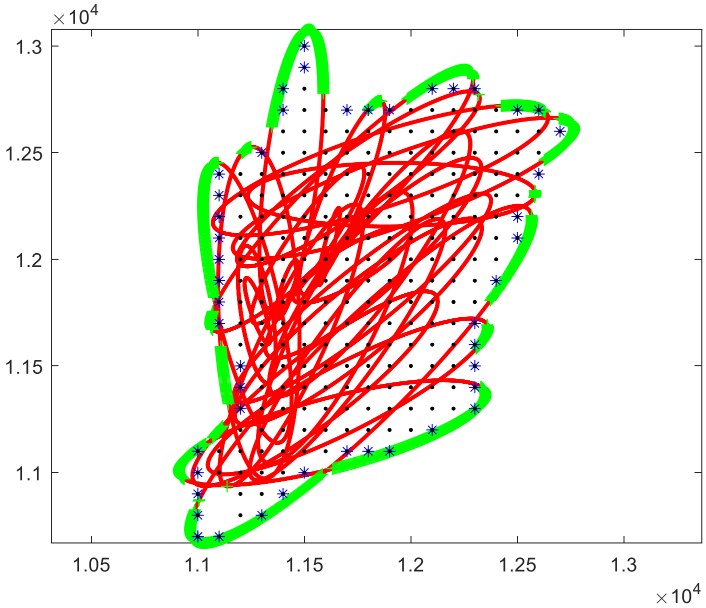


Figure 4.12 – Example of fire spread simulation during the process

4.2.7 Post-Analysis Function

This function will generate animations depicting the firefront location and the area affected by the fire over increasing time steps. Additionally, it will assess the damage to structures based on their exposure time to the fire, plotting various damage levels for different types of structures on the fire spread map. During this process, it will call the probabilistic model for different materials and then use the fragility curve to estimate the structure retention factor. For the visualization aspect, the different types of structures will be illustrated in different colors of boxes, and the damage level will vary from cool color to warm color. The model incorporates input parameters to specify the height and width dimensions of various structures. Correspondingly, the rectangle-shaped marker will be adjusted to reflect the dimensions of these structures. Then this function will plot the structures overlap the fire spread map and this aids in how the fire is approaching the structures and the effect of fire barrier. The function will further analyze the repair costs for different types of structures, taking into account varying levels of damage. These costs will cover both structural and non-structural components. For wildlands, the model will assume complete damage regardless of the number of time steps it has undergone. Ultimately, the sum of the repair costs across all nodes will be considered the regional economic loss for the specified fire scenario.

Finally, this dissertation utilizes rectangle markers to visualize the affected structures. The dimensions of these rectangles accurately represent the actual size of each structure. Additionally, this model can accept parameters for width and height. In the future, with access to the precise dimensions of all structures, the visualization of affected structures will become even more accurate.

4.2.8 Monte Carlo Simulation Function

The Monte Carlo simulation is outside the main function since it needs to call the main function and iterate it. This function enables the user to execute the model a specified number of times. It will record the outcomes of each iteration, including the time impacted, level of structure damage, and estimated economic loss. The subsequent analysis following the Monte Carlo simulation entails determining the final range of fire spread over increasing time steps. In this research, we will adopt a conservative approach by assuming that if a fire reaches a certain node during any iteration, this node will be designated as affected. This analysis will also assess the potential economic loss within the affected areas across different iterations or fire scenarios, adopting a similar conservative stance for the spread range. Additionally, the analysis will consider the likelihood of ignition and the potential damage to all structures.

4.3 Concluding Remarks and Capability of Matlab Model

4.3.1 Capability

This fire spread model accepts detailed input parameters, including map dimension, grid size, mesh size, initial ignition points, fuel, and environmental parameters that help to calculate the spread rate. This level of detail allows for highly customized simulations that can reflect real-world scenarios closely. It also provides an advanced fire spread simulation by integrating Rothermel’s model for calculating the rate of spread, Huygens’ principle and elliptical spread model for simulating fire propagation, and a firebrand model to account for spot fires adds a layer of complexity and realism to the simulation, enabling the model to predict fire behavior under various conditions accurately.

This model also includes the fuel type identification and image processing function and its potential use of machine learning for future enhancements demonstrates the model’s advanced capability in recognizing and categorizing different fuel types, crucial for accurate fire behavior prediction.

This model also permits the user to manually add and change fire barriers within the simulation model. By enabling the manipulation of fire barriers within the simulated environment, users can assess how strategic placement and modification of these barriers affect the overall spread of fire across various landscapes. This capability not only enhances the model’s utility for theoretical research but also serves as a vital tool for practical wildfire management and planning.

Another capability of this model will be the versatile output and analysis. It generates a wide array of outputs, including animations of fire spread, probability and fragility curves for different materials, and estimates of economic loss. Such comprehensive outputs are invaluable for assessing the impact of fire on structures and landscapes.

Finally, this model is capable of running Monte Carlo simulations which allows the user to explore a range of ire scenarios, providing insights into probable fire behavior and potential damage, ths aiding in better preparedness and mitigation strategies.

4.3.2 Limitation and Future Work

Despite its robust capabilities, this model also presents several limitations that could impact its utility and accuracy. First limitation about this model is the assumption of constant wind direction and velocity. This model assumes that wind direction and velocity remain constant during the simulation, this simplification may not accurately reflect real-world conditions where wind patterns can change rapidly, affecting fire spread direction and speed. Although this model employs Monte Carlo simulations to mitigate the effects of this assumption, discrepancies between the simulation and the real world will still persist. In the future, integrating this model with enhanced real-time data could significantly improve the accuracy of fire spread predictions. For example, every 10 minutes, the fire spread simulation can be compared with the real fire scenario and correct current simulation based on the real-time wind speed and direction.

The second limitation may be the fuel assignment. When the fuel is assigned, point-based fuel measurements cannot be interpolated or extrapolated. Unlike other models that treat the fuel at the firefront as a homogeneous fuel type, this model treats all nodes independently. However, this does not mean the fuel itself can be interpolated or extrapolated if the actual surveying grid does not match the mesh grid. This limitation can be addressed by obtaining an accurate fuel map and ensuring the grid distances are consistent.

Another significant perspective on the limitation concerns the running time. The computational complexity and running time of this model will significantly increase if multiple ignition points are involved or if the mesh size is reduced. Decreasing the mesh size results in a higher number of nodes on the map, thereby increasing the model's complexity. This may limit its practicality in urgent or extensive fire scenarios. Future work aimed at reducing running time and enhancing the model's efficiency could explore parallel computing for scenarios involving multiple ignition points. Additionally, revising and optimizing algorithms could help minimize the number of iterations, thereby improving efficiency. Adaptive Mesh Refinement (AMR) techniques, which dynamically adjust mesh size according to the simulation's needs, could address concerns related to mesh size. Finer meshes would be employed in areas of interest, while coarser meshes could be utilized in less critical regions.

In conclusion, this model offers a solid foundation for simulating fire spread, with extensive capabilities for customization and scenario analysis. Its comprehensive approach to modeling fire spread, coupled with the potential for future enhancements, positions it as a valuable tool for researchers, policymakers, and emergency management professionals seeking to mitigate the impacts of wildfires through informed planning and response strategies.

5. PARAMETRIC AND CASE STUDY

This chapter explores the parametric study and includes a case study of the Lahaina fire in Hawaii from August 2023. The parametric modeling section demonstrates the correlation between spread area and variations in input parameters, as well as the influence of changing fire barriers. The Matlab model discussed in Chapter 4 is validated for its capability to handle variations in input parameters. The case study begins by selecting the main variables for the fire scenario, followed by fire spread simulation, structural damage assessment, and loss evaluation. Additionally, the model enables users to manually add fire barriers and observe their impact on mitigating fire spread. The simulated fire spread region and the estimated economic loss are very close to the real estimates for the Lahaina Fire.

5.1 Case Study: Maui Fire 2023

5.1.1 Introduction

In August 2023, a series of devastating wildfires broke out in Hawaii, primarily affecting the island of Maui. The fires began on August 8, 2023, primarily affecting the island of Maui, Hawaii, especially the historic resort town of Lahaina. These wind-driven fires prompted evacuations and caused widespread damage, particularly in the town of Lahaina on Maui's northwest coast. The wildfires resulted in the deaths of at least 100 people and left four individuals missing, marking them as the deadliest wildfires in the U.S. in at least a century. Approximately 1,550 parcels and 2,200 structures were affected. The fires were particularly challenging for firefighters due to erratic winds and difficult terrain (PDC & Federal Emergency Management Agency, 2023).

By August 21, three of the four wildfires were still active. The fires significantly impacted the historic town of Lahaina, burning approximately 2,200 acres of residential homes and about 500 acres of commercial businesses. The fires were likely caused by a combination of factors, including sparks from a downed power line, exacerbated by the dry conditions of Hawaii's dry season, lasting from April to October. El Niño contributed to these conditions by causing drought on the Hawaiian Islands. Additionally, the hurricane Dora, passing south of the Hawaiian Islands, created high winds up to 107.8 km per hour on Maui, which intensified and spread the wildfires. In the wildfire risk perspective, the increase in wildfire intensity in recent decades in Hawaii has been attributed to the spread of non-native vegetation and hotter, drier weather due to climate change. The decline of agriculture in Hawaii has led to large areas of land being left unmanaged and overrun by flammable invasive species like guinea grass. Various organizations and agencies, including NASA, FEMA, and local Hawaiian organizations, engaged in relief and recovery efforts. NASA's Earth Applied Sciences Disasters program provided crucial data and mapping support for these efforts (Mak, et al., 2023).

5.1.2 Model Implementation

According to ABC News, "In the early morning hours, around 6:30 a.m., a brush fire spanning approximately three acres was reported near Lahainaluna Road, traversing the historic town of Lahaina in West Maui." The initial ignition points for the Lahaina fire can be located in Figure 5.1. By 9 a.m., Maui County officials declared the fire in Lahaina to be 100% contained (Alfonseca, 2023). Thus, a realistic estimation of the Lahaina fire's total duration ranges from 2 to 3 hours. It is presumed that approximately one hour was required for the fire to cool down and for firefighters to bring it under control. For this case study, a duration of 2 hours will be adopted.

Figure 5.1 – The initial ignition point according to ABC News.

Weather data for Maui from August 8 to August 11 can be obtained from NOAA and is available for download as an .nc file. Figure 5.2a and Figure 5.2b herein illustrate examples of eastward and northward winds on Maui on August 8, 2023. After computing the wind speed and direction, it was found that the maximum wind speed is approximately 17 m/s (3346 ft/min) with the wind direction spanning from northeast to southwest. It is important to note that NOAA employs atmospheric pressure, rather than altitude, to measure wind velocity. Consequently, the wind velocity reported by NOAA may be measured at over 100 meters above sea level. However, in Rothermel's model, wind velocity should be assessed at the midpoint of the flame, which approximates the ground level wind velocity. Thus, the speeds derived from NOAA might not be suitable inputs for this model. In this research, the equation developed by J.C.Solano (2021) will be utilized to calculate the average wind velocity at the pedestrian level in Lahaina. Equation 5.1 is used to calculate the average wind speed.

$$w_{avg} = -1.04z^4 + 1.127z^3 + 2.836z^2 + 1.167z + 4.115 \tag{5.1}$$

Here, z is determined by Equation 5.2 where h denotes the elevation above sea level.

$$x = \frac{\mathrm{h}-1644}{1346}, 0 < h < 4000 \tag{5.2}$$

Although originally devised for assessing wind velocity in the southern region of Ecuador, the equation will be applied to Lahaina, given the proximity of both locales to the sea. An average wind speed of approximately 1.98 m/s (390 ft/min) at 10 meters above sea level has been calculated. Consequently, it is presumed that the wind velocity at mid-flame during fire propagation will fall within a range of 300 ft/min to 600 ft/min. The wind direction will be considered within a range of 135 to 315 degrees. Figure 5.3 shows the final wind velocity vector plot for this case study.

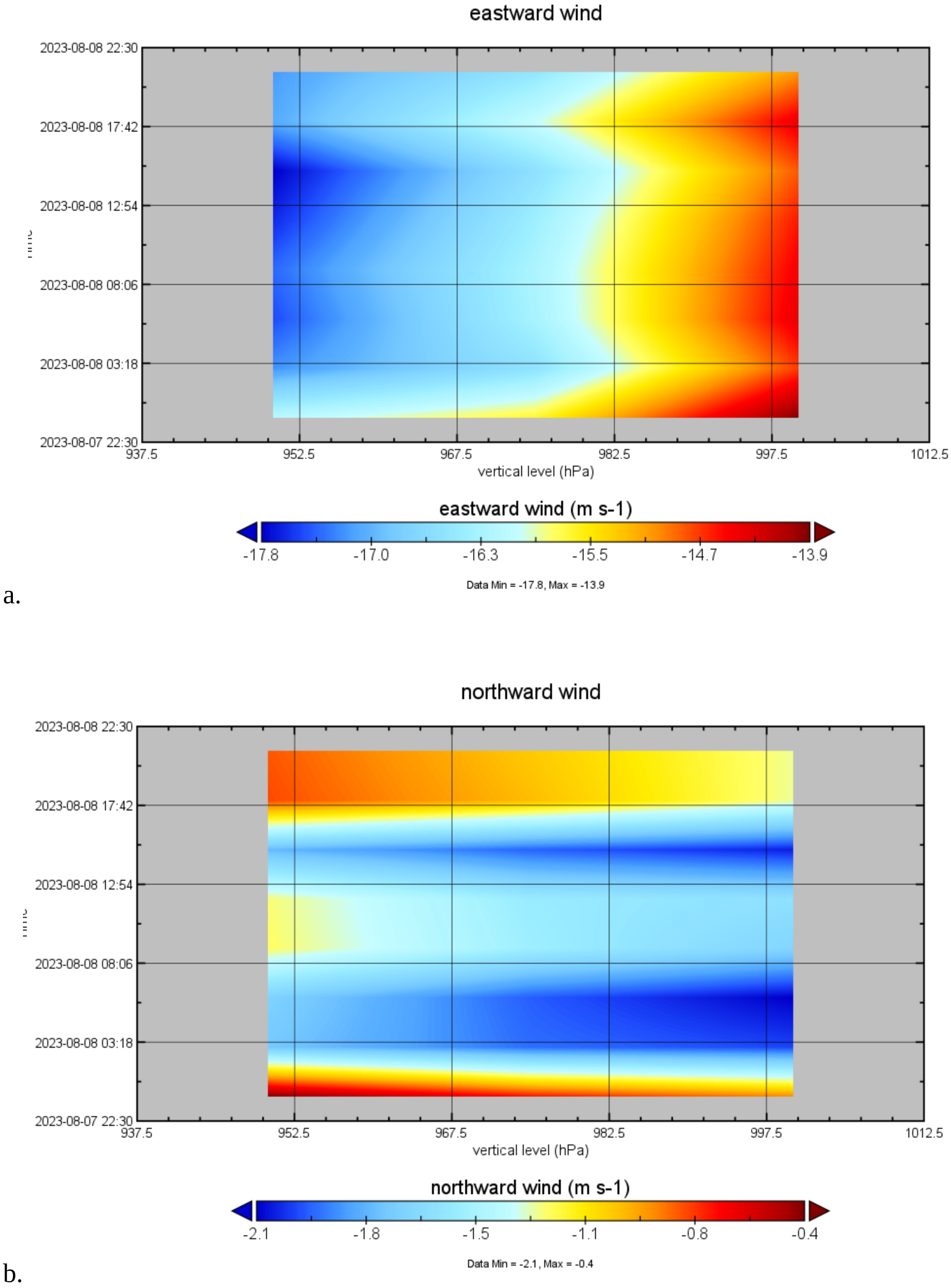


Figure 5.2 – **Example wind velocity retrieved from NOAA**. a) Eastward wind. b) Northward wind

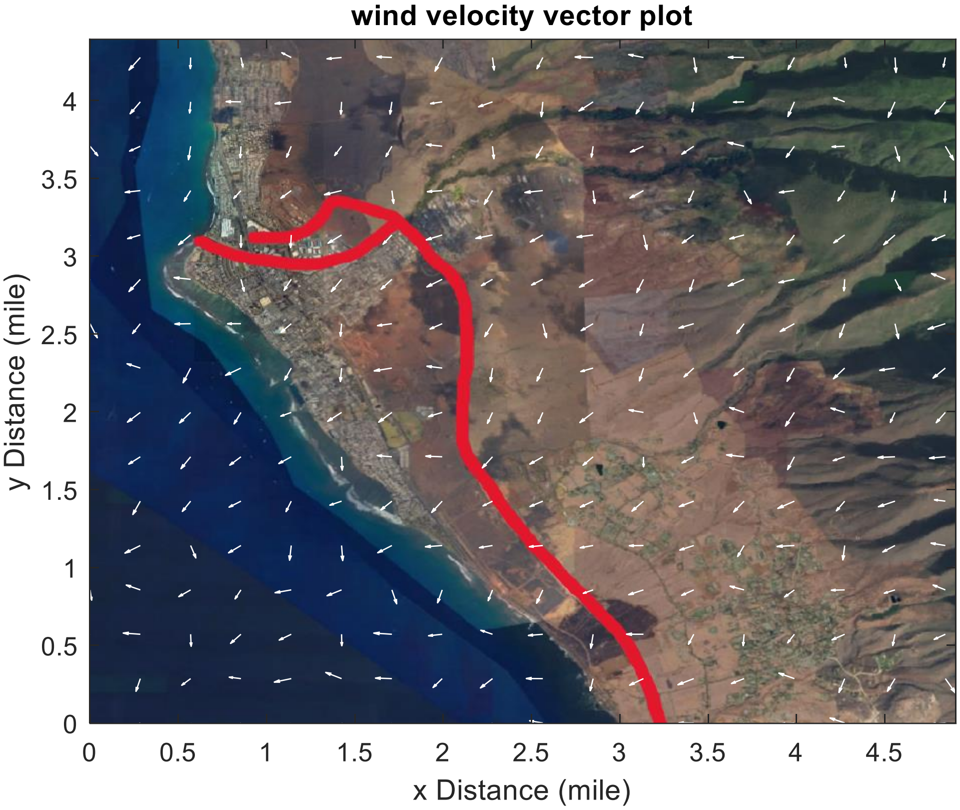


Figure 5.3 – Wind velocity vector plot for Lahaina Fire

Fuel data is randomly selected from the fuel models within the BEHAVE and NFDRS systems, then assigned to each node on the map (Andrews, 1986; Finney, 1998). Currently, obtaining precise fuel data directly from the map is challenging. However, in the future, machine learning technique could potentially offer a viable method to acquire exact fuel data, thereby enhancing the accuracy of simulation results. The image processing function in this model assists in identifying water and fire barriers but cannot differentiate between fuel types based solely on pixel data. This map features a total of 16 test structures, comprising nine concrete structures, five steel structures, and two wood structures. A portion of these structures is strategically positioned at a considerable distance from the ignition points, potentially rendering them unaffected by the fire. Conversely, others are situated in close proximity to the ignition points, serving a crucial role in examining and assessing the extent of damage incurred. This diverse placement allows for a comprehensive analysis of how different materials and distances from ignition points influence the vulnerability and resilience of structures to fire damage.

5.2 Parametric Study

5.2.1 Fire Spread Parametric Study

For the fire spread parametric study, this model compares various factors, including changes in total duration, wind speed, mesh size, wind direction, and initial ignition points. By analyzing these variables, the model provides a comprehensive understanding of how different parameters influence fire behavior and spread.

Figure 5.4a to Figure 5.4e demonstrate the simulations with different input parameters. Figure 5.4a represents the base case with a total duration of 2 hours, time step is 1/3 hour, wind direction is between 135 and 315 degrees, wind speed is between 300 ft/min and 600 ft/min, and with only one ignition point. Figure 5.4b changes the total duration time to 4 hours. The affected area is much larger than the base case which is Figure 5.4a. Figure 5.4c changes the mesh grid to 50 feet and the affected area becomes larger. This is because when the mesh size gets smaller when generating the sub-ellipses, the vertices used for next time step will be closer to the edge of the ellipse, which will make the affected area larger. Figure 5.4d changes the wind speed between 600 ft/min to 900 ft/min and the affected area is about 1.5 to 2 times larger than the based case. Figure 5.4e changes the wind direction between -45 to 135 degrees. It is clear that the boundary of the sea is stopping the propagation of the firefront, if the direction changes, there will be more area burnt and the damage will be more serious.

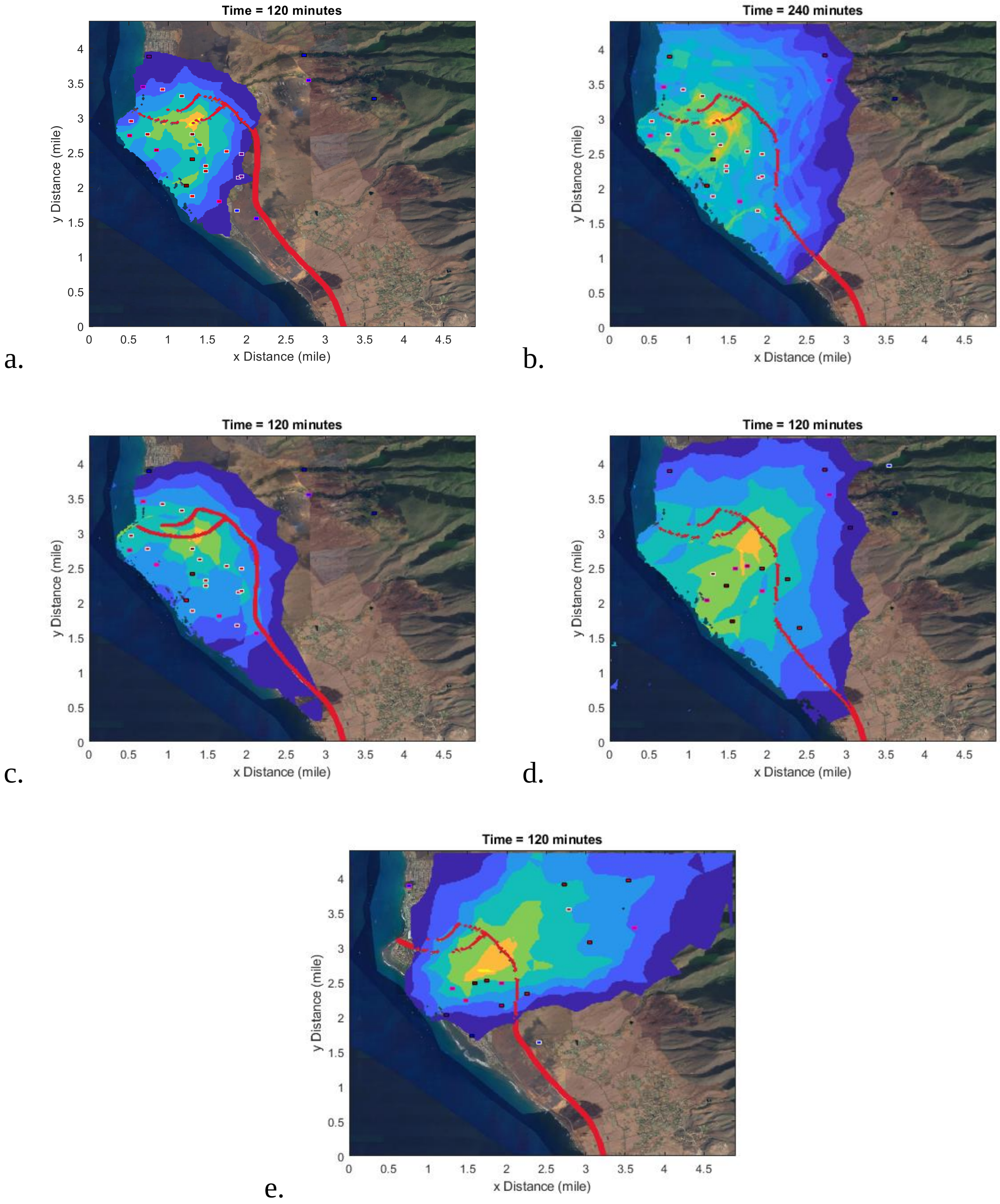
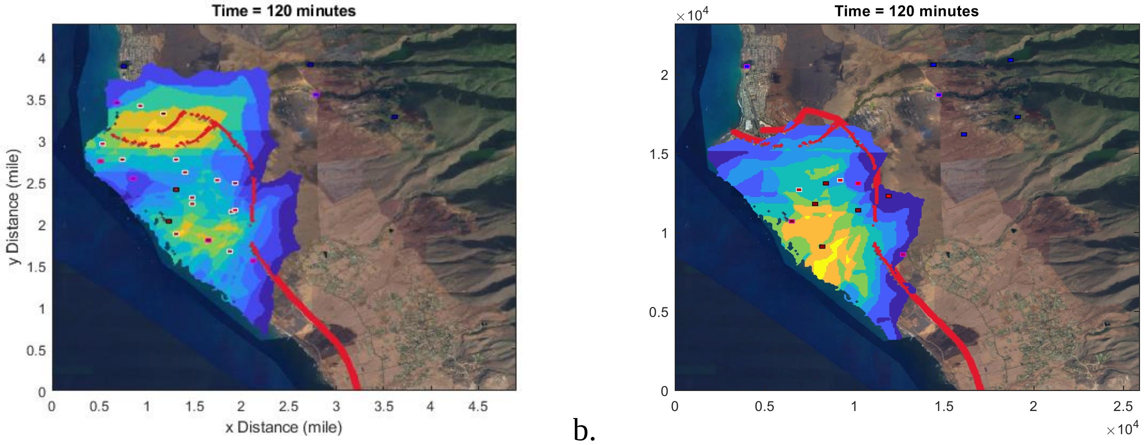


Figure 5.4 – **Parametric Study**. a) The base case; b) Changing the total duration to 4 hours; c) Changing mesh gird size to 50 feet; d) Changing the wind speed to between 600 ft/min and 900 ft/min; e) Changing the wind direction to between -45 to 135 degrees.

This model is able to handle multiple initial ignition points. Figure 5.5 shows the difference for the changing in the initial ignition points. Compared to the base case, the affected area has extended southward. Furthermore, because the two ignition points are close, the fire intensity is relatively high at the overlapped area.



a.

b.

Figure 5.5 – **Comparison with different numbers of ignition points**. a) Simulation with two ignition points; b) Simulation with one ignition point

Figure 5.6 compares the effect of firebrand with the original case. The significance of the firebrand's effect is considered to be minimal. This is likely attributed to an overly conservative approach towards the firebrand model. At each node, it is assumed that only a single firebrand will be generated, with the median distance traveled by the firebrand being approximately 4 meters (13.12 feet). Given that a mesh size of 100 feet is utilized, the impact of the firebrand is deemed to be negligible. Nonetheless, distinctions can still be observed in the images. For future considerations, it is possible that more firebrands could be generated from a single node, and the utilization of a smaller mesh size would render the effects of the firebrand more pronounced.

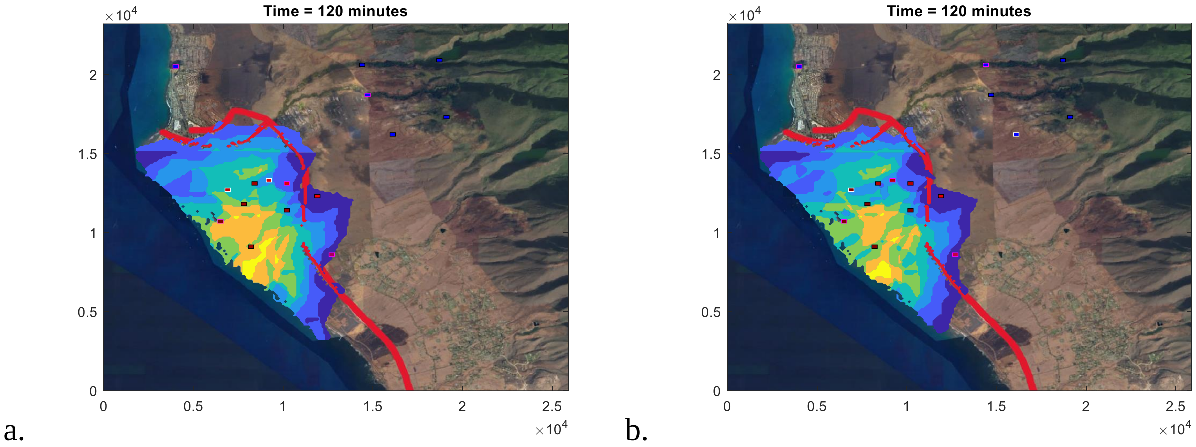
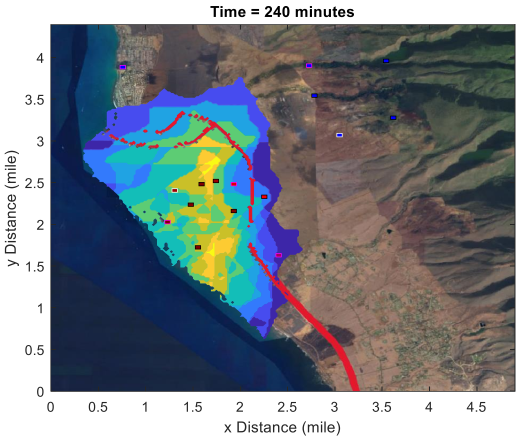


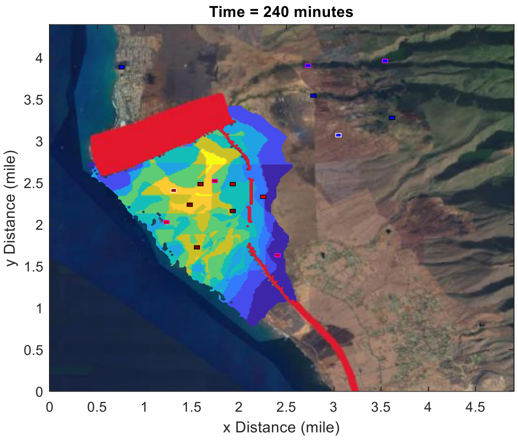
Figure 5.6 – Comparison with the effect of firebrand

5.2.2 Fire Barrier Parametric Study

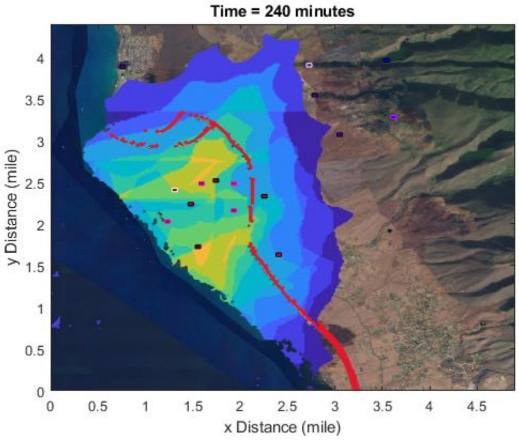
Figure 5.7a to Figure 5.7f represents how different fire barriers will affect the spreading of the fire. By the addition of the fire barrier, significant changes are observed in the propagation of the fire and the final affected area. As depicted in Figure 5.7a and Figure 5.7b, the inclusion of a fire barrier prevents further generation of the fire front, leaving the area north of the fire barrier unaffected. Figure 5.7d illustrates that, with an increase in wind velocity, fire propagation surpasses the fire barrier on its east side, subsequently spreading from east to west. This contrasts with the base case where, as shown in Figure 5.7c, the fire predominantly propagates from the east side rather than the south. In Figures 5.7e and 5.7f, both the speed and direction of the wind are altered to observe differences. The area north of the barrier remains unaffected, whereas the shape of the area south of the barrier undergoes significant alteration. It is noted that certain portions of the firefront are unable to cross the barrier, leaving the area to the east unaffected. Nonetheless, a segment of the fire front breaches the barrier, initiating new ignition points from which the fire spreads. This occurrence is attributed to the increased wind velocity, which elevates the ratio of the elliptical parameters, thereby elongating the ellipse. Should the major axis of the ellipse exceed the width of the fire barrier, it is concluded that the fire can cross the barrier, potentially igniting a new fire on the opposite side.



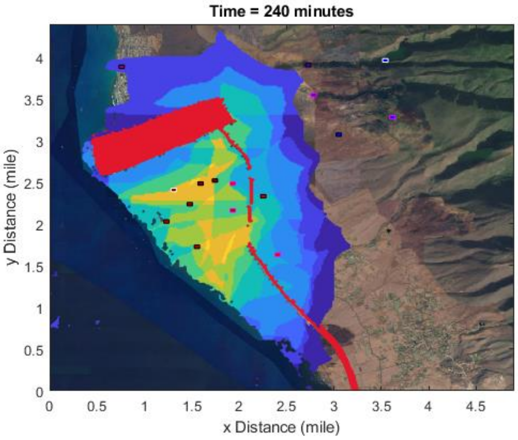
a.



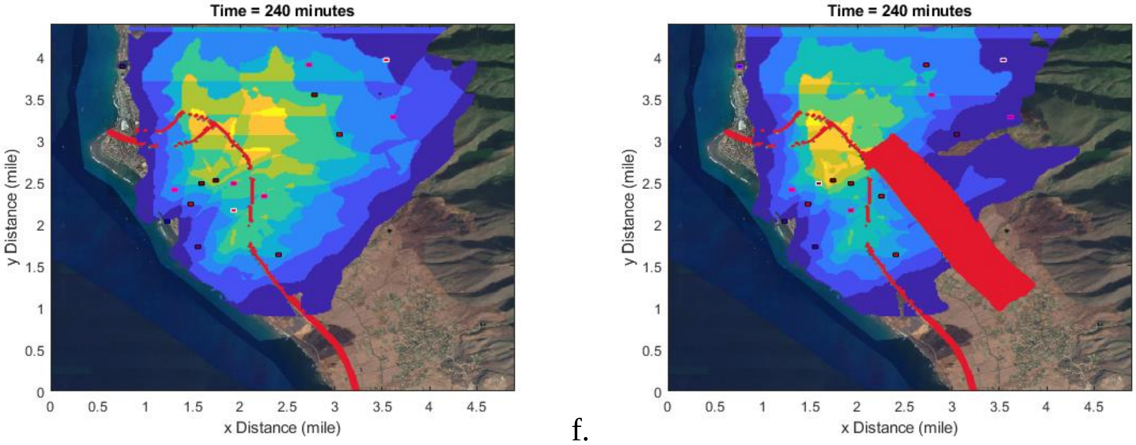
b.



c.



d.

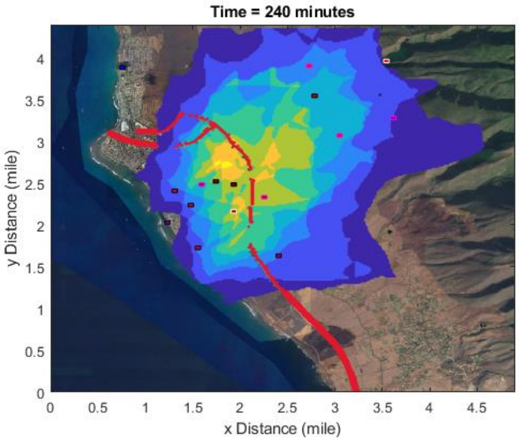


e. f.

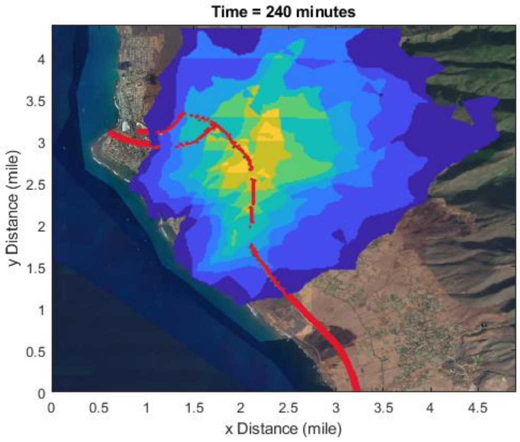
Figure 5.7 – Comparison with the effect of fire barriers

5.2.3 Monte Carlo Simulation

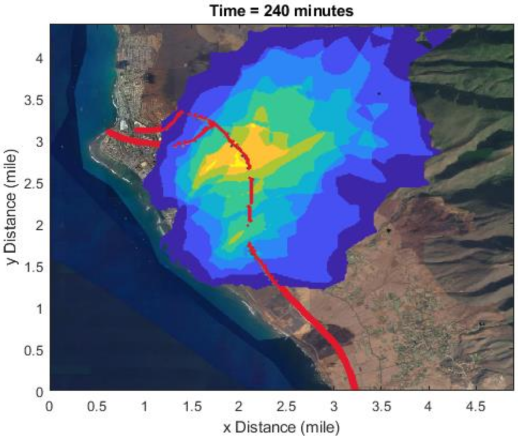
Figure 5.8a to Figure 5.8j represents the Monte Carlos simulation with 10 iterations and Figure 5.8k is the final simulation results, which combines all the results from the 10 iterations. Within the iteration, the affected time for all nodes is recorded. For structures, both the affected time and the retention factors are documented. Upon the completion of all iterations, the average affected time for all nodes is calculated. Nodes with an affected time greater than 0 are depicted in the final simulation. In such instances, a node affected in even a single iteration is considered affected. The use of the average affected time also indicates the potential intensity of damage a node might experience. The application of the Monte Carlo simulation further elucidates the probability of ignition for each structure. Future projections include the generation of a Poisson distribution for all structures to evaluate their damages in a specified fire scenario. The final economic loss and repair cost are determined using the results from the final simulation.


Time = 240 minutes
y Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
x Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

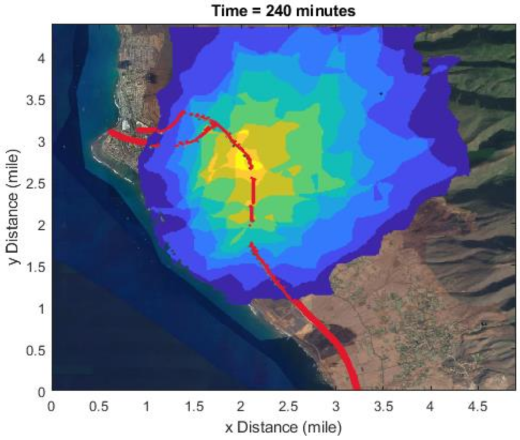
a.


Time = 240 minutes
y Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
x Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

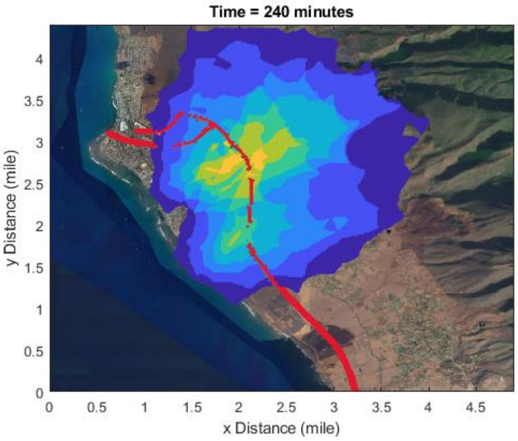
b.


Time = 240 minutes
y Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
x Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

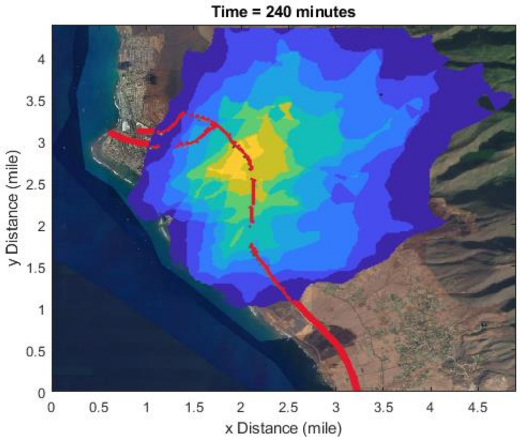
c.


Time = 240 minutes
y Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
x Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

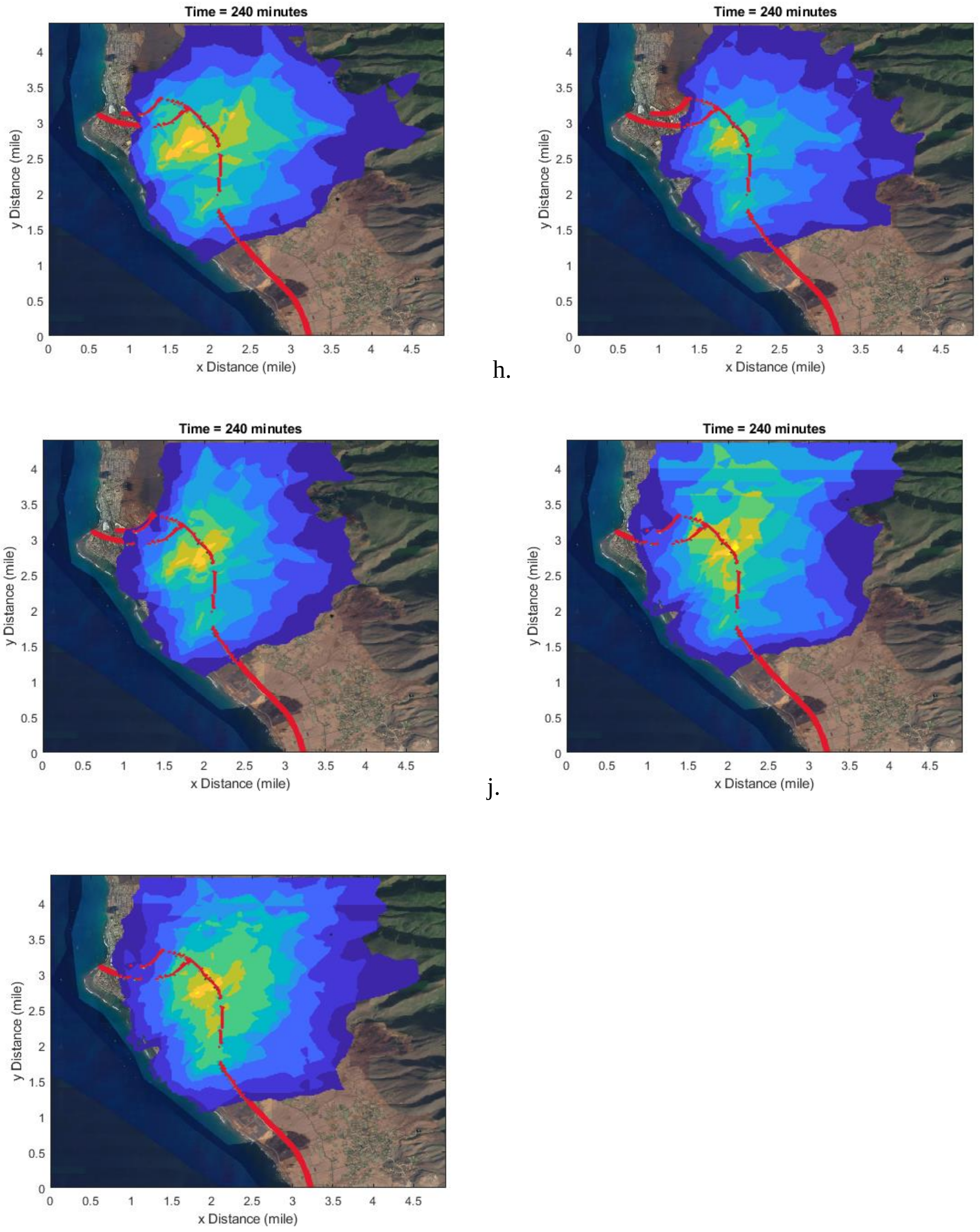
d.


Time = 240 minutes
y Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
x Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

e.


Time = 240 minutes
y Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
x Distance (mile)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

f.



g.

h.

i.

j.

k.

Figure 5.8 – Monte Carlo simulation with 10 iterations

5.3 Comparison of Loss Assessment

To estimate economic losses and reconstruction costs in the affected area, random unit prices were allocated to each node, including those representing wildlands and structures. The unit cost for wildland or unoccupied land is presumed to range from \$3 to \$10 per square foot. It is assumed that once a wildland node is reached by the fire, regardless of the duration of exposure, it is considered a total loss. Regarding the structures, the simulation considered three types: concrete, wood, and steel. Concrete structures, typically used for offices, schools, or hospitals, tend to incur higher unit costs due to their larger occupied areas and greater number of floors. Residential buildings, usually made of wood, often have lower unit costs, smaller footprints, and fewer floors. Steel structures fall between the two in terms of unit cost, occupied space, and floor count. In this research, the unit cost for concrete structures ranges from \$450 to \$650 per square foot, with floor areas between 3,000 and 5,000 square feet. The number of floors is randomly assigned a value between 4 and 6. For steel structures, the unit cost ranges from \$400 to \$550 per square foot, and floor areas span from 2,500 to 3,500 square feet, with the number of floors being randomly selected from 3 to 5. Lastly, wood structures have a unit cost ranging from \$250 to \$500 per square foot, with their occupied areas ranging from 1,200 to 2,500 square feet, and floor levels set between 1 and 3. Table 5.1 summaries the assumptions for different structures and wildland costs

Table 5.1 – Assumptions for structures and wildland costs

	Occupying area (ft^2)	Unit cost $(\frac{\$}{ft^2})$	Floor level
Concrete structure	3000 - 5000	450 - 650	4 - 6
Steel structure	2500 - 3500	400 - 550	3 - 5
Wood structure	1200 - 2500	250 - 500	1 - 3
Wildland	Varies with grid distance	3 - 10	1

Figure 5.9 originates from the Pacific Disaster Center and the Federal Emergency Management Agency, showcasing the estimated fire perimeter highlighted in light red. In contrast, Figure 5.10 represents a simulation conducted as part of this research. This simulation suggests that the extent of the estimated burnt area spreads along the coastal region, both to the north and to the south. A potential explanation for this discrepancy could be the emergence of new fires that are not accounted for in this model, which solely focuses on the initial ignition point. This model is capable of handling multiple initial ignition points; however, it is limited in its ability to generate new ignition points during the simulation. Another potential reason for this could be attributed to the wind. In this study, the wind direction is predetermined and strictly confined to a range of 135 to 315 degrees. Although the Monte Carlo method helps to mitigate some of this issue, the fixed wind direction still restricts the fire's spread to the north or south. A possible solution to the first issue could involve pre-identifying areas that are prone to catching fire and designating them as new ignition points within the simulation. To address the second issue, it would be advantageous to integrate this model with external tools capable of determining real-time wind speed and direction. This integration would allow for the wind speed and direction to be updated at every timestep, thereby enhancing the accuracy of the simulation results. The simulation generated from this study is also compared to the SWUIFT simulation

generated by Juliano in 2024 which is shown in Figure 5.11. The concerns are the same as this dissertation, the uncertainties of wind speed and directions, however, Juliano considers the dynamics of wind, like hydraulic jump and turbulent flow (Juliano, et al., 2024). In the future, the model developed in this dissertation can also be improved by implementing the dynamics of wind.

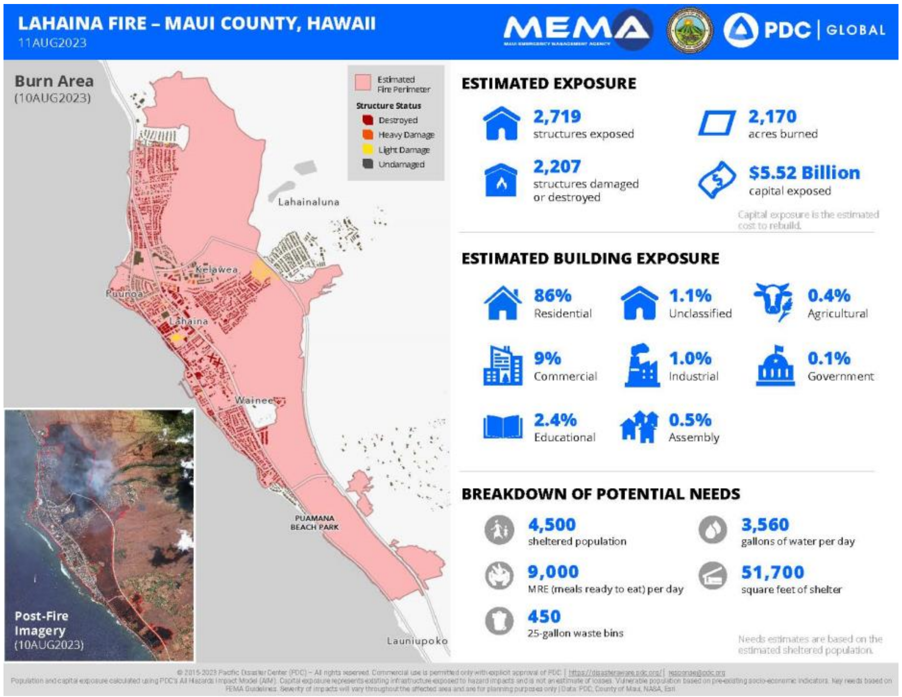


Figure 5.9 – Lahaina Fire report from PDC and Federal Emergency Management Agency (retrieved on 6/1/2024)

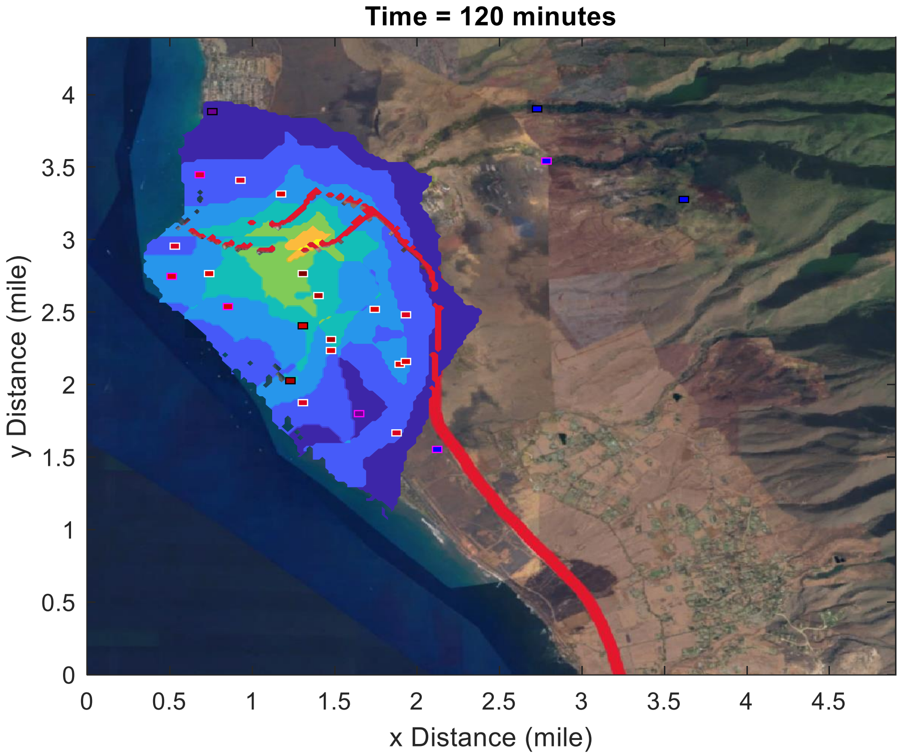


Figure 5.10 – Simulated affected area for Lahaina Fire from this study

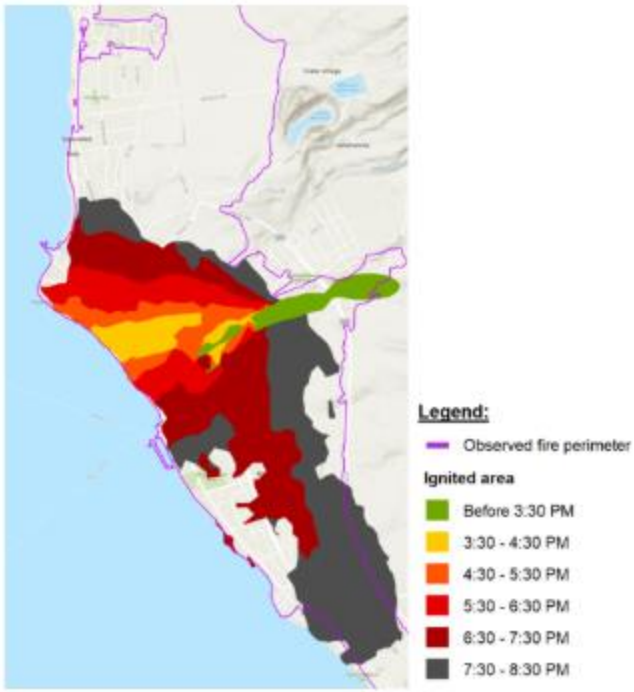


Figure 5.11 – Fire perimeters from the SWUIFT simulation (from Juliano, et al., 2024)

According to the Pacific Disaster Center (PDC), the estimated burned area encompasses approximately 2,170 acres. In contrast, simulations suggest a slightly smaller burned area of about 2021.4 acres. When considering potential extensions to both the northern and southern areas, the estimated burned areas are expected to be very similar. The University of Hawaii Economic Research Organization (UHERO) estimates total economic losses to be in the range of $4 billion to $6 billion (Mak, et al., 2023). However, it's important to note that this estimate does not account for damage to wildlands and unoccupied land. According to this research, the total economic loss is estimated to be about $1.05 billion. It's important to note that this study estimates the unit cost for wildland to be between $3 and $15 per square foot, and considers only 20 structures to be affected during the simulation. As a result, the estimated total economic loss may differ significantly. In this simulation, a total of 25 test structures were introduced into the map, with 20 being affected by the fire. Of these, 2 were concrete, 4 were steel, and 14 were wood structures. Assuming the residential buildings comprise steel and wood, approximately 90% of the test structures represented residential buildings. This figure is close to the 86% of residential buildings reported as damaged during the Lahaina fire by the PDC. When focusing solely on the rebuilding cost of structures, according to the Pacific Disaster Center (PDC), 2,207 structures were damaged or destroyed with an estimated reconstruction cost of approximately $5.52 billion. According to this research, the reconstruction cost for the 20 structures is $45.6 million. Extrapolating from this and accounting for the same amount ration of residential buildings, if a total of 2,207 structures are considered, the total reconstruction cost would be approximately $5.03 billion. Taking into account the possibility of the fire spreading further north and south, it is probable that additional structures would incur damage. Consequently, the estimated reconstruction cost as determined by this research may exceed $5.03 billion. Figure

5.12 shows a sample regional loss assessment at each time step. The blue line is the structure loss and the red line is the wildland loss. The yellow line is the total loss and is calculated by the sum of structure loss and wildland loss at that time step. The slope of the loss curve increases as time increases and it significantly increases at 100 minutes. This is reasonable since the dimension of the firefront will increase exponentially as the area gets larger and larger. There will be more structure affected as the time increases. For example, the area generated for time step 4 will be much larger than the area generated between time step 1.

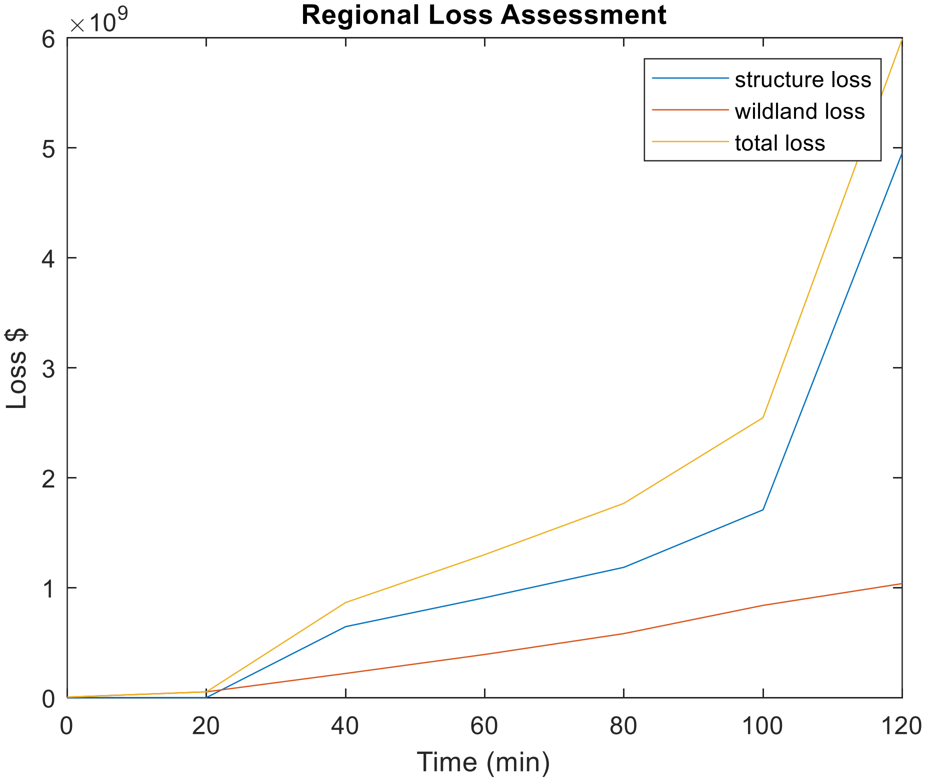
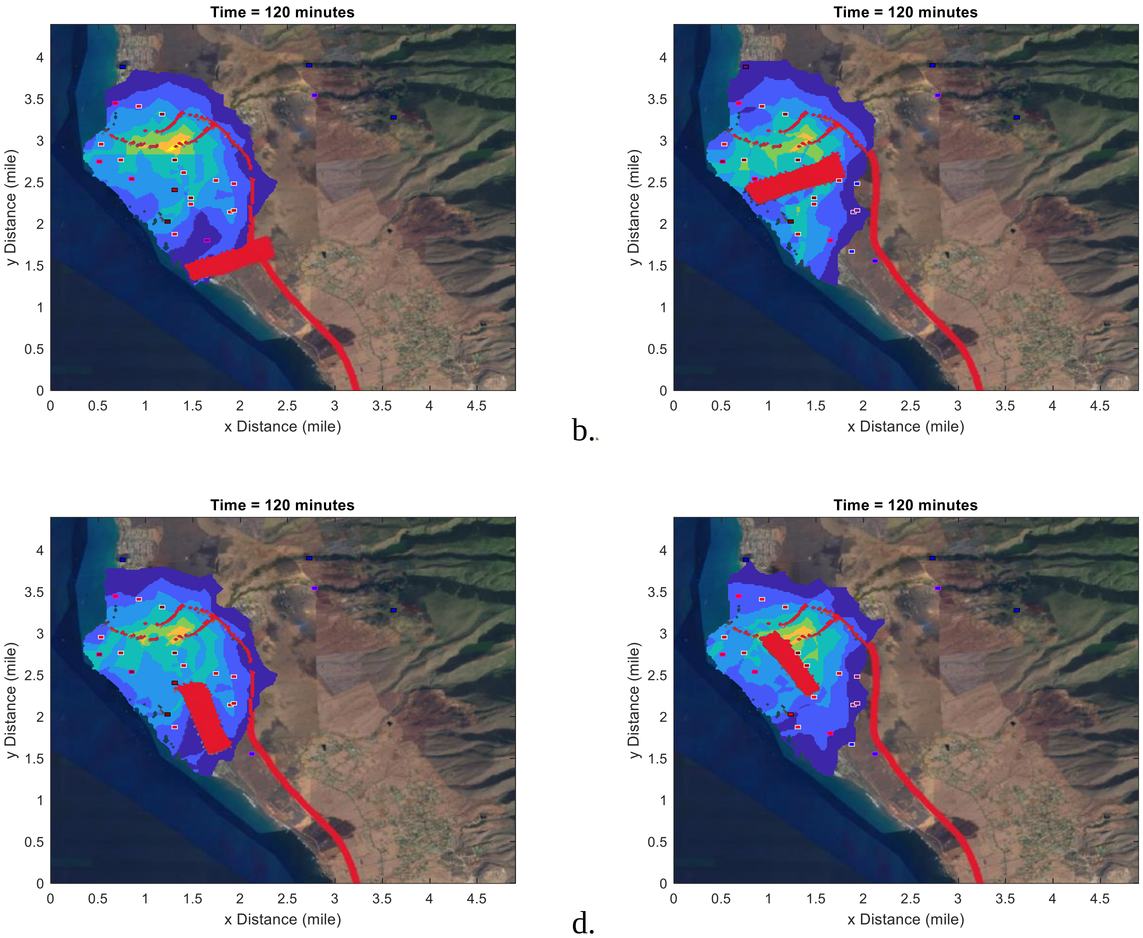


Figure 5.12 – Sample regional loss assessment versus time

5.4 Case Study Mitigation

Figure 5.13a to Figure 5.13d show that four different test barriers have been added to the map to observe their effects. The total affected areas for these four scenarios are 1955.9 acres, 1736.7 acres, 1853.3 acres, and 1681.4 acres, respectively. Placing the fire barrier as close as possible to the ignition point, given a fixed duration, significantly decreases the total affected area. Although the firefront may breach the fire barrier, both the fire intensity and the total affected area are reduced. Figure 5.13c demonstrates that the fire propagates around the barrier rather than penetrating it, provided the barrier is sufficiently long in the direction of the fire's spread. According to Figure 5.13b, positioning the fire barrier perpendicular to the fire's direction helps greatly reduce the total affected area. Conversely, when the location of structures is fixed, the numbers of affected structures in these four scenarios are 19, 17, 16, and 17, respectively. The total economic loss, including the destruction of wildlands and damage to structures, amounts to $1.02 billion, $0.90 billion, $0.94 billion, and $0.88 billion.



a. b. c. d.

Figure 5.13 – Comparison between different fire barriers

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

This dissertation offers a comprehensive exploration of fire spread simulation and its impact on structural behavior in both wildland and urban interfaces. Central to this research is the development and implementation of an advanced fire spread model, which integrates Rothermel’s surface fire spread model, Huygens' principle of fire front expansion, and Monte Carlo simulation. This combination of methodologies forms the foundation of the research, providing accuracy and insight into the dynamics of fire spread across various terrains and conditions. The novelty of this research lies in its holistic approach to modeling fire behavior, considering a variety of factors including weather conditions, terrain variability, and fuel types. This multifaceted model not only predicts the rate and direction of fire spread but also assesses the damage on concrete, steel, and wooden structures after exposure.

Furthermore, the incorporation of firebrands into the model addresses one of the most challenging aspects of fire spread in urban areas: the ignition of spot fires. By considering the impact of firebrands, this research explains the complex process by which fires propagate in densely built environments, enhancing the understanding of urban fire dynamics.

Another significant part of the research focuses on assessing structural damage after fire exposure. The introduction of a probabilistic model and fragility curves for different types of structures provides a more precise way to estimate potential damage. This methodology not only improves the accuracy of damage assessment but also helps in planning resource allocation for fire suppression and recovery efforts.

Additionally, using Monte Carlo simulation to model the economic losses from fire damage adds a new aspect to risk assessment. This approach allows for the estimation of losses probabilistically, giving a more detailed and comprehensive understanding of the economic impact of fires on both wildlands and urban infrastructures.

In conclusion, this dissertation transcends traditional fire modeling techniques by offering a more robust and versatile tool for predicting fire spread and assessing structural vulnerability. Its contributions to the field of fire safety engineering are manifold, including the development of a model that integrates advanced computational methods with empirical data to simulate the complex behavior of fires in diverse settings. This model is crucial for urban planners, civil engineers, and fire management professionals, offering a valuable tool to better predict, mitigate, and manage the adverse effects of fires.

6.2 Future Work

The research undertaken in this dissertation has laid a significant foundation in the field of multi-hazard risk assessment, particularly focusing on fire spread and structural damage in various environments. However, there is always room for expansion and improvement. The future work proposed here aims to address several key areas that would not only enhance the model's accuracy and applicability but also its practical utility in fire engineering and urban planning.

First, further refinement of the firebrand model could enhance the accuracy of predicting spot fire ignition and spread. Additionally, integrating real-time data on weather and terrain could improve the model's predictive capabilities, enabling more dynamic and responsive fire management strategies.

Exploring the development of urban-specific models could also yield significant benefits, considering the unique challenges presented by urban fire spread. Experimental validation of the model's predictions through controlled field experiments would provide valuable feedback for refining and improving its accuracy and reliability.

Moreover, future research could delve into more sophisticated methods for estimating economic losses, incorporating indirect costs and long-term impacts on ecosystems and communities. Collaboration across disciplines, including meteorology, urban planning, and material science, could foster the development of holistic models that address the multifaceted nature of fire risk and management.

In addition to refining the model itself, future endeavors will encompass a range of other critical areas. These areas aim to broaden the scope of our research, integrating a holistic approach to fire risk management that merges technological advancements with broader societal and environmental considerations.

Incorporation of Human Intervention (Fire Suppression) Mechanisms: Future iterations of the model should aim to integrate human intervention strategies, particularly fire suppression efforts. This enhancement would involve modeling the effectiveness of various fire suppression techniques, including water spraying, chemical retardants, and physical barriers, in altering the course and intensity of fires. Understanding the dynamics of fire suppression can aid in optimizing response strategies and minimizing damage.

Fuel Cooling Down Phase: Adding a fuel cooling down phase to the model would represent a significant improvement in simulating the lifecycle of a fire more accurately. This phase would account for the reduction in temperature and cessation of burning in areas where the fire has passed, affecting subsequent fire spread and suppression efforts. Modeling the cooling phase is crucial for a comprehensive understanding of fire dynamics, especially in predicting re-ignition and evaluating the effectiveness of suppression tactics.

Machine Learning for Optimizing Fire Hydrant and Fire Department Locations: Leveraging machine learning algorithms could revolutionize how resources for fire suppression are allocated, particularly regarding the placement of fire hydrants and fire departments. By analyzing historical fire incident data, urban layouts, and the model's fire spread predictions, machine learning can identify patterns and optimize the locations of fire suppression resources. This approach would not only enhance response times but also improve the overall efficiency of urban fire management systems.

In conclusion, this research embodies a comprehensive, innovative approach to understanding and managing the risks associated with fires, offering valuable insights and tools for professionals in civil engineering, urban planning, and fire safety management. As the threats posed by fires continue to evolve in the face of climate change and urban expansion, the contributions of this research will undoubtedly play a crucial role in shaping future strategies for fire risk mitigation and community resilience.

APPENDIX

Source Code

Main functions

montecarlo.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to use Monte Carlo Simulation to output final
% results
%
% Input
%
% Output
%      ignition_possibility - probability for structures get ignition
%      avg_retention_factor - mean retention factor for structures
%      expected_cost - expected ignition probability * initial cost
%      Total_cost - economic loss within the area (wildland + structures)
%
%      figure() - Monte Carlo Simulation plot with all iterations
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc, clear all, close all
% tic
imagename = 'maui_fire.png';
[a b M N Z nodeproperty] = main();
k = 10;                                                                     %number
of iteration
MC_Z = zeros([size(Z),k]);
MC_nodeproperty = zeros([size(nodeproperty),k]);
MC_Z(:,:,:,1) = Z;
MC_nodeproperty(:,:,1) = nodeproperty;
for i = 2:k
    [a b M N Z nodeproperty] = main();
    MC_Z(:,:,:,k) = Z;
    MC_nodeproperty(:,:,k) = nodeproperty;
end

MC_Z(isnan(MC_Z)) = 0;                                                      %change
all NaNs into 0 to find average
avg_Z = mean(MC_Z,4);
avg_Z(avg_Z == 0) = NaN;                                                    %change
all zeros into NaNs for plotting

a_mile = a * 0.000189394;
b_mile = b * 0.000189394;
M = M .* 0.000189394;
N = N .* 0.000189394;
```

```
figure()
I = imread(imagename);
h = image(linspace(0,a_mile,a),linspace(0,b_mile,b),flipud(I));
set(gca,'YDir','normal');
xlabel('x Distance (mile)')
ylabel('y Distance (mile)')
title(['Fire spread simulation with ' num2str(k) ' iterations'])
hold on

xlim([0 a])
ylim([0 b])

for j = 1:size(MC_Z,3)
    h = contourf(M,N,avg_Z(:,:,j),'LineColor','none');
    xlim([0 a])
    ylim([0 b])
    shading interp

    drawnow();
    axis tight

    pause(0.5)
end
hold off
% toc

%expected repair cost
all_affected_time = MC_nodeproperty(:,7,:);
all_affected_time(all_affected_time > 0) = 1;
all_affected_cost = MC_nodeproperty(:,8,:);
ignition_possibility = mean(all_affected_time,3);
avg_retention_factor = mean(all_affected_cost,3);
expected_cost = ignition_possibility .* nodeproperty(:,6);
Total_cost = sum(expected_cost);
```

main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is the main function and runs for one iteration
%
% Input
%
% Output
%       a - real map length (matrix columns)
%       b - real map width (matrix rows)
%       M - meshed M matrix
%       N - meshed N matirx
%       Z - 3-D matrix stores affected time step for all nodes
%       nodeproperty - consists all the parameters of all nodes
%                    - [node_num, x_co, y_co, ROS, node_type, tributary_cost, affected 
time, retension factor, repair cost]
```

```
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [a b M N Z nodeproperty] = main()
% This function is going to find the spread rate based on environmental
% parameters and generate an animation that shows the spread

% This is the main function
% clc, clear all, close all

tic
% input
imagename = "maui_fire_barrier.png";
[a b wind_a wind_b m n time nnodes node connectivity input_co ...
    grid_distance_a grid_distance_b excel_grid_distance dt velocity...
    U_wind slope ellipse_a ellipse_b ellipse_c unit_area] = fun_input;

% generate connectivity                         change node property structure
[connectivity x y M N node nodeproperty slope ellipse_a ellipse_b ellipse_c] =
fun_connect(a, b, wind_a, wind_b, m, n, nnodes, node, velocity,...
                                                                            U_wind,
slope, grid_distance_a, grid_distance_b, excel_grid_distance, ...
                                                                            ellipse_a,
ellipse_b, ellipse_c, unit_area, imagename);

% generate firebrand model
[spot_co] = firebrand_distance3D(nnodes);                                   %generate
a 3D firebrand model and output all the coordinates
fb_id = randi(length(spot_co(:,1)),nnodes,1);                               %generate
one random firebrand node number
fb_coord = spot_co(fb_id,:);                                                %find the
coordinate according to the node number

% add firebrand model to ellipse parameter
fb_coord = 3.28084 * fb_coord;                                              %convert
meter to feet
ellipse_b = ellipse_b + fb_coord(:,1);
ellipse_a = ellipse_a + fb_coord(:,2);

%[nodeproperty,water,barrier] =
imageprocess(a,b,m,n,nnodes,nodeproperty,grid_distance_a,grid_distance_b);
[nodeproperty,water,barrier] = image_preprocess1(a, b, m, n, nnodes, nodeproperty,
grid_distance_a, grid_distance_b, imagename);

%save variables for python
filename = 'matlabdata.mat';
save(filename)

% ignition process and generate animation
% [aff_node aff_node_nz ig_counter aff_coord result result_d fire_front Z] =
fun_process(connectivity, result, time, nnodes, node, x, y, input_co, grid_distance_a,
dt,a, b, m, n, M, N, velocity4dir, nodeproperty, excel_grid_distance);
[firefront_x firefront_y firefront_xcheck firefront_ycheck result Z] = fun_process3(m,
n, x, y, node, nnodes, connectivity, input_co,time,dt,...
```

```
grid_distance_a,grid_distance_b,slope, ellipse_a, ellipse_b, ellipse_c,
nodeproperty,water,barrier);

% post simulation including structure damage and repair cost analysis
[M N Z nodeproperty] = fun_postresults(result, nodeproperty, time, dt, M, N, Z, a, b,
firefront_x, firefront_y, firefront_xcheck, firefront_ycheck, imagename);

total_area = sum(nodeproperty(:,7) > 0) * grid_distance_a * grid_distance_b *
2.29568e-5;       %in acres
str_idx = find(nodeproperty(:,5) == 4 | nodeproperty(:,5) == 5 | nodeproperty(:,5) ==
6);
num_affected_str = sum(nodeproperty(str_idx,7)>0);
structure_cost = sum(nodeproperty(str_idx,9));
total_cost = sum(nodeproperty(:,9));

toc
```

fun_input

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function should intake all the parameters from outside file, user
% should define the mesh grid distance
% Input
%       parameters from outside file (ie. excel)
%
% Output
%       connectivity matrix (node_num down left right up)
%       a - real map length (matrix columns)
%       b - real map width (matrix rows)
%       m - mesh length
%       n - mesh width
%       time - simulation duration
%       nnodes - total number of nodes
%       node - [node_num x_co y_co]
%       result - ignition level
%       connectivity - connect four nodes (down left right up)
%       input_co - input ignition coordinates
%       grid_distance_a - mesh grid distance in x direction
%       grid_distance_b - mesh grid distance in y direction
%       dt - time step
%       velocity - velocity array contains velocity at that node
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [a b real_an real_bn m n time nnodes node connectivity input_co ...
    grid_distance_a grid_distance_b excel_grid_distance dt velocity ...
    U_wind slope ellipse_a ellipse_b ellipse_c unit_area] = fun_input

% input data (feet,hours)
```

```
% unitratio = [mph, ft/min]
% mapunits = [mile, feet]
a = 25872;                                                                  %a&b are
the map size in feet
b = 23180;
time = 2;                                                                   %specify
the time duration (hour)
dt = 1/3;

% distance_unit = input('What is the input distance unit?','s')               %ask
the unit for input
% time_unit = input('What is the input time unit?', 's')
% if strcmp(distance_unit, 'mile')
%     a = 5280 * a;
%     b = 5280 * b;
% end
% if strcmp(time_unit, 'minutes')
%     time = time / 60
% end

excel_grid_distance = 100;                                                  %grid
distance from excel (ft)
real_an = floor(a/excel_grid_distance);                                     %used to
find interpolation of excel grid
real_bn = floor(b/excel_grid_distance);

grid_distance_a = 100;                                                      %user
defined grid distance
grid_distance_b = grid_distance_a;

m = floor(a/grid_distance_a);                                               %matlab
mesh size
n = floor(b/grid_distance_a);

% input_co = [11400,11400];                                                 %initial
ignition points' coordinates (according to mesh size)
input_co = [7300,15700];                                         %initial ignition
points' coordinates (according to mesh size)
% input_co =
[8900,14100;10200,9500];                                         %initial ignition
points' coordinates (according to mesh size)

%%%use (x, y) find closest (m,n)

% output zero matrix
nnodes = m*n;                                                               %total
nodes with meshed map
node = zeros(nnodes,3);
connectivity = zeros(nnodes,5);                                             %find
adjscent nodes (not used in process3)

%get input parameter and calculate velocity
input_data = xlsread('matlab_input_fixed.xlsx');
data = input_data(input_data(:,1) <= (floor(a)*floor(b)),:);
h = data(:,2);
```

```
S_T = data(:,3);
S_e = data(:,4);
rho_p = data(:,5);
sigma = data(:,6);
w0 = data(:,7);
delta = data(:,8);
M_x = data(:,9);
M_f = data(:,10);
U_wind = data(:,11);                                                        %unit
feet/min
steepness =
data(:,12);                                                     %steepness in degrees
change tand to tan if use radians
slope = data(:,33);                                                         %wind
direction

[velocity R_0 phi_w phi_s C B E beta beta_op] =
v_cal(h,S_T,S_e,rho_p,sigma,w0,delta,M_x,M_f,U_wind,steepness);  %calculate velocity
at each node in feet/min
slope = deg2rad(slope);                                                     %convert
degree to radian

%calculate ellipse parameters for interpolation (ellipse_axis2)
%[ratio2 ellipse_a2 ellipse_b2 ellipse_c2] = ellipse_axis(velocity, U_wind,dt);
[ratio ellipse_a ellipse_b ellipse_c] = ellipse_axis2(R_0, phi_w, phi_s, C, B, E,
beta,...
                                            beta_op, slope, U_wind, dt);

%calculate the tributory area at each node
floors = data(:,35);
unit_area = grid_distance_a * grid_distance_b;
```

fun_connect

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function should generate a connectivity matrix that shows the
% adjacent nodes
% Input
%       a - real map length
%       b - real map width
%       real_an - number of nodes in real grid distance in length
%       real_bn - number of nodes in real grid distance in width
%       m - number of nodes in mesh grid distance in length
%       n - number of nodes in mesh grid distance in width
%       nnodes - total number of nodes
%       node - [node_num x_co y_co]
%       velocity - velocity array contains ROS at each node
%       U_wind - wind speed at each node
%       slope - wind direction at each node
%       grid_distance_a - mesh grid distance in length
%       grid_distance_b - mesh grid distance in width
%       excel_grid_distance - survey grid distance from original data
%       ellipse_a - elliptical parameter
```

```
%      ellipse_b - elliptical parameter
%      ellipse_c - elliptical parameter
%      unit_area - tributory area for each node
%      imagename - testing image name

% Output
%      x - the x coordinate of all the nodes
%      y - the y coordinate of all the nodes
%      M - meshed M matrix
%      N - meshed N matirx
%      node - node matrix [node_num x_co y_co]
%      nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%      slope - interpolated wind direction respect to mesh size
%      ellipse_a - interpolated elliptical parameter
%      ellipse_b - interpolated elliptical parameter
%      ellipse_c - interpolated elliptical parameter
%
%      figure(1) - wind velocity vector plot
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [connectivity x y M N node nodeproperty slope ellipse_a ellipse_b ellipse_c]
= fun_connect(a, b, real_an, real_bn, m, n, ...

nnodes, node, velocity, U_wind, slope, grid_distance_a,grid_distance_b,...

excel_grid_distance, ellipse_a, ellipse_b, ellipse_c,...

unit_area, imagename)

% define variable matrix
temp = 0;
connectivity = zeros(nnodes,5);
velocity_mesh = zeros(real_bn,real_an);
U_wind_mesh = zeros(real_bn,real_an);
slope_mesh = zeros(real_bn,real_an);
ellipse_a_mesh = zeros(real_bn,real_an);
ellipse_b_mesh = zeros(real_bn,real_an);
ellipse_c_mesh = zeros(real_bn,real_an);

%generate node coordinates
for j = 0 : grid_distance_b : floor(b) - grid_distance_b
    for i = 0 : grid_distance_a : floor(a) - grid_distance_a
        temp = temp + 1;
        node(temp,1) = temp;
        node(temp,2) = i;
        node(temp,3) = j;
    end
end

x = node(:,2);
y = node(:,3);

%generate connectivity array
for temp = 1 : nnodes
```

```
    connectivity(temp,1) = temp;
    if temp <= m
        connectivity(temp,2) = NaN;
    else
        connectivity(temp,2) = temp - m;
    end
    if mod((temp - 1),m) == 0
        connectivity(temp,3) = NaN;
    else
        connectivity(temp,3) = temp - 1;
    end
    if mod(temp,m) == 0
        connectivity(temp,4) = NaN;
    else
        connectivity(temp,4) = temp + 1;
    end
    if temp >= (n-1)*m
        connectivity(temp,5) = NaN;
    else
        connectivity(temp,5) = temp + m;
    end
end

%interpolate velocity within grid
k = 1;
for i = 1:real_bn                                                          %real_an
and real_bn are mesh on real map
    for j = 1:real_an
        velocity_mesh(i,j) = velocity(k);
        U_wind_mesh(i,j) = U_wind(k);
        slope_mesh(i,j) = slope(k);

        ellipse_a_mesh(i,j) = ellipse_a(i+j-1);
        ellipse_b_mesh(i,j) = ellipse_b(i+j-1);
        ellipse_c_mesh(i,j) = ellipse_c(i+j-1);
        k = k+1;
    end
end

new_a = linspace(0,floor(a),a/excel_grid_distance);
new_b = linspace(0,floor(b),b/excel_grid_distance);
new_m = linspace(0,floor(a)-grid_distance_a,m);
new_n = linspace(0,floor(b)-grid_distance_b,n);

[mesh_A,mesh_B] = meshgrid(new_a,new_b);
[M,N] = meshgrid(new_m,new_n);

%convert real spread rate, wind velocity and slope into meshed size
new_velocity = interp2(mesh_A,mesh_B,velocity_mesh,M,N);
new_velocity(isnan(new_velocity)) = 0;
new_wind_velocity = interp2(mesh_A,mesh_B,U_wind_mesh,M,N);
new_wind_velocity(isnan(new_wind_velocity)) = 0;
new_slope = interp2(mesh_A,mesh_B,slope_mesh,M,N);
new_slope(isnan(new_slope)) = 0;
```

```
%convert ellipse_a ellipse_b ellipse_c into meshed size
new_ellipse_a = interp2(mesh_A,mesh_B,ellipse_a_mesh,M,N);
new_ellipse_a(isnan(new_ellipse_a)) = 0;
new_ellipse_b = interp2(mesh_A,mesh_B,ellipse_b_mesh,M,N);
new_ellipse_b(isnan(new_ellipse_b)) = 0;
new_ellipse_c = interp2(mesh_A,mesh_B,ellipse_c_mesh,M,N);
new_ellipse_c(isnan(new_ellipse_c)) = 0;

%reshape test
%a=[7 8 9;4 5 6;1 2 3]
%reshape((flip(a))',[9,1])

%reshape into one column
velocity = reshape((flip(new_velocity))',[nnodes,1]);
U_wind = reshape((flip(new_wind_velocity))',[nnodes,1]);
slope = reshape((flip(new_slope))',[nnodes,1]);

ellipse_a = reshape((flip(new_ellipse_a))',[nnodes,1]);
ellipse_b = reshape((flip(new_ellipse_b))',[nnodes,1]);
ellipse_c = reshape((flip(new_ellipse_c))',[nnodes,1]);


%plot wind velocity vector
U = new_wind_velocity .* cos(new_slope);
V = new_wind_velocity .* sin(new_slope);
M_sub = M(1:15:end, 1:15:end);
N_sub = N(1:15:end, 1:15:end);
U_sub = U(1:15:end, 1:15:end);
V_sub = V(1:15:end, 1:15:end);
%convert feet into mile
a_mile = a * 0.000189394;
b_mile = b * 0.000189394;
M_sub = M_sub * 0.000189394;
N_sub = N_sub * 0.000189394;
U_sub = U_sub * 0.000189394;
V_sub = V_sub * 0.000189394;

figure(1)
I = imread(imagename);
h = image(linspace(0, a_mile, a),linspace(0, b_mile, b),flipud(I));
set(gca,'YDir','normal');
xlabel('x Distance (mile)')
ylabel('y Distance (mile)')
title('wind velocity vector plot')
hold on
quiver(M_sub,N_sub,U_sub,V_sub,'w','AutoScale', 'off')
%axis tight
hold off



%velocity4dir = [velocity velocity*2 velocity velocity];

%change node property to test
%use following to find node number
```

```
%find(nodeproperty(:,2)==7200 & nodeproperty(:,3) == 6900)

type =
ones(nnodes,1);                                                    %%1=normal
fuel,2=water,3=barrier,4=concrete,5=steel,6=wood

%%assign random tributary_cost
unit_cost = randi([3,20], length(type), 1);
tributary_cost = unit_cost * unit_area;
nodeproperty = [node velocity type tributary_cost];

%%tests to change node property to structure
test_con_structure = [4000,20500; 6900,12700; 14400,20600; 6500,10700;
19100,17300]; %assign test coordinates
[~,idx] = ismember(test_con_structure, nodeproperty(:,2:3), 'rows');
nodeproperty(idx, 5) = 4;                                               %assign
structure type
concrete_floors = randi([2, 5]);                                        %assign
random floors
concrete_area = randi([2000, 5000], length(idx), 1) .* concrete_floors;     %assign
random area
concrete_unit_cost = randi([400, 600], length(idx), 1);                 %assign
random structure unit cost
concrete_initial_cost = concrete_area .* concrete_unit_cost;            %assign
test structure repair cost
nodeproperty(idx, 6) = concrete_initial_cost;                           %change
variable in nodeproperty

test_steel_structure = [2700,14500; 3600,18200; 4500,13400; 14700, 18700; 8700,9500;
11200,8200];
[~,idx] = ismember(test_steel_structure, nodeproperty(:,2:3), 'rows');
nodeproperty(idx,5) = 5;
steel_floors = randi([1, 5]);
steel_area = randi([1500, 4000], length(idx), 1) .* steel_floors;
steel_unit_cost = randi([300, 500], length(idx), 1);
steel_initial_cost = steel_area .* steel_unit_cost;
nodeproperty(idx, 6) = steel_initial_cost;

test_wood_structure = [2800,15600; 3900,14600; 4900,18000; 6200,17500; 6900,14600;
7400,13800; 7800,12200; 9200,13300; 7800,11800; 6900,9900; 10200,13100; 10200,11400;
10000,11300; 9900,8800];
[~,idx] = ismember(test_wood_structure, nodeproperty(:,2:3), 'rows');
nodeproperty(idx,5) = 6;
wood_floors = randi([1, 3]);
wood_area = randi([1000, 3000], length(idx), 1) .* wood_floors;
wood_unit_cost = randi([200, 400], length(idx), 1);
wood_initial_cost = wood_area .* wood_unit_cost;
nodeproperty(idx, 6) = wood_initial_cost;
```

image_preprocess1

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to find the structure damage condition at current
% time
% Input
%      a - real map length (matrix columns)
%      b - real map width (matrix rows)
%      m - mesh length
%      n - mesh width
%      nnodes - total number of nodes
%      nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%      grid_distance_a - mesh grid distance in x direction
%      grid_distance_b - mesh grid distance in y direction
%      imagename - testing image name (used for barrier testing)
%
% Output
%      nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%      water - water nodes [node number, x coordinate, y coordinate]
%      barrier - barrier nodes [node number, x coordinate, y coordinate]
%
%      figure(3) - map with identified fuels
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [nodeproperty,water,barrier] = image_preprocess1(a, b, m, n, nnodes,
nodeproperty, grid_distance_a, grid_distance_b, imagename)

[I,cmap] = imread(imagename);                                 %read the image (image
scans from top left corner
croppedImage = imresize(I, [n, m]);                                         %resize
the resolution of original image

%use matlab built-in function to find the coordinates for different fuel
%types (I used color thresholder)
[BW,masked] = findwater(croppedImage);
[BW2,masked2] = findbarrier(croppedImage);
% [BW_concrete,masked_concrete] = findconcrete(croppedImage);
% [BW_steel,masked_steel] = findsteel(croppedImage);
% [BW_wood,masked_wood] = findwood(croppedImage);

[row,col] = find(BW);
[row2,col2] = find(BW2);
% [row_concrete, col_concrete] = find(BW_concrete);
% [row_steel, col_steel] = find(BW_steel);
% [row_wood, col_wood] = find(BW_wood);

%change the coordinates (up-side down) into what I can use
%temp_a is related with the size of image, if the image keeps the same, then temp_a
is the same
temp_a = size(masked,1);
row = temp_a-row;                                                           %convert
the y-axis upside down
row2 = temp_a-row2;
% row_concrete = temp_a - row_concrete;
% row_steel = temp_a - row_steel;
```

```
% row_wood = temp_a - row_wood;

col = col;
col2 = col2;
% col_concrete = col_concrete;
% col_steel = col_steel;
% col_wood = col_wood;

%plot the assigned fuel type on each sub-image
% plot(col,row,'r.')
% plot(col2,row2,'k.')
% plot(col_concrete, row_concrete,'w.')
% plot(col_steel, row_steel,'w.')
% plot(col_wood, row_wood, 'y.')
% hold off

%The following if statements convert sub-image coordinates (local
%coordinates) into simulation coordinates (global coordinates) need to use
%new if statement if more fuel types needs to be identified

%find the node number for each fuel type
%findnodenum will return 3 columns, only column 1 is meaningful here
row = row * grid_distance_b;
row2 = row2 * grid_distance_b;
% row_concrete = row_concrete * grid_distance_b;
% row_steel = row_steel * grid_distance_b;
% row_wood = row_wood * grid_distance_b;

col = col * grid_distance_a;
col2 = col2 * grid_distance_a;
% col_concrete = col_concrete * grid_distance_a;
% col_steel = col_steel * grid_distance_a;
% col_wood = col_wood * grid_distance_a;

water = findnodenum(col,row,nodeproperty);
barrier = findnodenum(col2,row2,nodeproperty);
% concrete = findnodenum(col_concrete,row_concrete,nodeproperty);
% steel = findnodenum(col_steel,row_steel,nodeproperty);
% wood = findnodenum(col_wood,row_wood,nodeproperty);

% water = findnodenum(col(1:length(col)-1),row(1:length(row)-1),nodeproperty);
% barrier = findnodenum(col2(1:length(col2)-1),row2(1:length(row2)-1),nodeproperty);

%assign the fuel properties (1=normal
fuel,2=water,3=barrier,4=concrete,5=steel,6=wood)
if isempty(water) ~= 1
    water_reduced = all(water(:, 1) == 0, 2);
    water = water(~water_reduced, :);
    nodeproperty(water(:,1),5) = 2;
end
if isempty(barrier) ~= 1
    barrier_reduced = all(barrier(:, 1) == 0, 2);
    barrier = barrier(~barrier_reduced, :);
    nodeproperty(barrier(:,1),5) = 3;
end
```

```
% if isempty(concrete) ~= 1
%     concrete_reduced = all(concrete(:, 1) == 0, 2);
%     concrete = concrete(~concrete_reduced, :);
%     nodeproperty(concrete(:,1),5) = 4;
% end
% if isempty(steel) ~= 1
%     steel_reduced = all(steel(:, 1) == 0, 2);
%     steel = steel(~steel_reduced, :);
%     nodeproperty(steel(:,1),8) = 5;
% end
% if isempty(wood) ~= 1
%     wood_reduced = all(wood(:, 1) == 0, 2);
%     wood = wood(~wood_reduced, :);
%     nodeproperty(wood(:,1),5) = 6;
% end

%Change spread rate to 0 for certain types of fuel type (water, barrier)
for i = 1: nnodes
    nodeproperty(find(nodeproperty(:,5)==3),4) = 0;
    %test for barrier, barrier is using 2 which should be structure
    nodeproperty(find(nodeproperty(:,5)==2),4) = 0;
end

%Figure 13 plot the assemble image after assigning fuel types
figure(3)
I = imread(imagename);
h = image(linspace(0,a,a),linspace(0,b,b),flipud(I));
set(gca,'YDir','normal');
hold on
if isempty(water) ~= 1
    plot(water(:,2),water(:,3),'b.');
end
if isempty(barrier) ~= 1
    plot(barrier(:,2),barrier(:,3),'k.');
end
% if isempty(concrete) ~= 1
%     plot(concrete(:,2),concrete(:,3),'w.');
% end
% if isempty(steel) ~= 1
%     plot(steel(:,2),steel(:,3),'w.');
% end
% if isempty(wood) ~= 1
%     plot(wood(:,2),wood(:,3),'y.');
% end
hold off
```

fun_process3

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function should generate a connectivity matrix that shows the
% adjacent nodes
% Input
```

```
%      m - number of nodes in mesh grid distance in length
%      n - number of nodes in mesh grid distance in width
%      x - the x coordinate of all the nodes
%      y - the y coordinate of all the nodes
%      node - [node_num x_co y_co]
%      nnodes - total number of nodes
%      connectivity - node connectivity (neihbourhood nodes) (not used)
%      input_co - initial ignition points
%      time - total duration
%      dt - time step
%      grid_distance_a - mesh grid distance in length
%      grid_distance_b - mesh grid distance in width
%      slope - wind direction at each node
%      ellipse_a - elliptical parameter
%      ellipse_b - elliptical parameter
%      ellipse_c - elliptical parameter
%      nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%      water - water nodes [node number, x coordinate, y coordinate]
%      barrier - barrier nodes [node number, x coordinate, y coordinate]
%
% Output
%      firefront_x - x coordinate for firefront nodes
%      firefront_y - y coordinate for firefront nodes
%      firefront_xcheck - x coordinate for firefront nodes without ROS = 0
%      firefront_ycheck - y coordinate for firefront nodes without ROS = 0
%      result - consists all the affected nodes and its current state
%      Z - 3-D matrix stores affected time step for all nodes
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [firefront_x firefront_y firefront_xcheck firefront_ycheck result Z] =
fun_process3(m,n,x,y,node,nnodes,connectivity,input_co,time,dt,grid_distance_a,grid_d
istance_b,slope,ellipse_a, ellipse_b, ellipse_c, nodeproperty,water,barrier)

temp_dt = dt;                                                               %used
temp_Z = zeros(n,m);                                                        %Create Z
cell matrix to store results (time affected by fire)
Z = zeros(n,m,time/dt + 1);
firefront_x = [];                                                           %create
empty matrix for firefront coordinates x
firefront_y = [];                                                           %%create
empty matrix for firefront coordinates x

firefront_x = NaN(nnodes,time/dt + 1,size(input_co,1));                     %fill the
empty matrix with NaN, filling with 0 will impact nodes with spread rate = 0
firefront_y = NaN(nnodes,time/dt + 1,size(input_co,1));
firefront_xcheck = firefront_x;
firefront_ycheck = firefront_y;

aff_node = [];                                                              %create
empty matrix for affected nodes
result = zeros(nnodes,1);                                                   %create
temperory result matrix to store result at each time step

% help to plot ellipse at every time step
```

```
% figure(2)
% I = imread('satellite.png');
% h = image(linspace(0,a,a),linspace(0,b,b),flipud(I));
% set(gca,'YDir','normal');
% hold on

%This for loop is to find the first ellipse (firefront coordinates) generated by the
first set of
%ignition points located by input_co. This is necessary for simulation with
%multiple ignition locations, otherwise, the boundary at each timestep will
%mess up with different ignition locations.
for i = 1:size(input_co,1)
    node_num = find(node(:,2)==input_co(i,1) & node(:,3)==input_co(i,2));
%     % only for testing
%     for k = 1:length(input_co)
%         node_num(k) = find(node(:,2)==input_co(k,1) & node(:,3)==input_co(k,2));
%     end

%     [x_e y_e] =
generate_ellipse(input_co(i,1),input_co(i,2),ellipse_a(node_num),ellipse_b(node_num),
ellipse_c(node_num),slope(node_num));
%     [x_inbound y_inbound aff_node] =
boundedpoints(x,y,x_e,y_e,node,nnodes,aff_node);
    [x_e,y_e,x_inbound,y_inbound,aff_node] =
generate_ellipse2(x,y,node,aff_node,input_co(i,1),input_co(i,2),ellipse_a(node_num),e
llipse_b(node_num),ellipse_c(node_num),slope(node_num));

    result(aff_node) = result(aff_node) + 1;
    if length(water) ~= 0
        result(water(:,1)) = 0;
    end
    if length(barrier) ~= 0
        result(barrier(:,1)) = 0;
    end
    temp_result = result(aff_node);
    aff_coord = node(aff_node,:);

    firefront_x(1:length(x_inbound),1,i) = [x_inbound];
    firefront_y(1:length(y_inbound),1,i) = [y_inbound];

    for ii = 1:length(aff_node)
        row_index = round(aff_coord(ii,3)/grid_distance_a);
        col_index = round(aff_coord(ii,2)/grid_distance_b);
        if row_index > 0 && col_index > 0
%             Z{1}(row_index, col_index) = temp_result(ii);
            Z(row_index, col_index,1) = temp_result(ii);
        end
    end

end
%%can also work to set the input points as firefront
% for i = 1:size(input_co,1)
%     firefront_x{i} = input_co(:,1)
%     firefront_y{i} = input_co(:,2)
% end
```

```
% for i=1:2
%     plot(firefront_x{i},firefront_y{i},'ro')
% end


%This for loop is to find the firefront coordinates for the following
%timesteps, store the results and record the firefront coordinates,
%affected node coordinates
for j = 2:time/dt + 1
    for i = 1:size(firefront_x,3)
        %remove all NaN rows
        temp_ff_x = rmmissing(firefront_x(:,j - 1,i));
        temp_ff_y = rmmissing(firefront_y(:,j - 1,i));

        %for k = 1:sum(~isnan(firefront_x(:,j - 1,i)))

        for k = 1:length(temp_ff_x)
            node_num(k) = find(node(:,2)==temp_ff_x(k) & node(:,3)==temp_ff_y(k));
        end
%          node_num = transpose(find(ismember(node(:,2:3), [temp_ff_x, temp_ff_y],
'rows')));
        [x_e,y_e,x_inbound,y_inbound,aff_node] =
generate_ellipse2(x,y,node,aff_node,temp_ff_x,temp_ff_y,ellipse_a(node_num),ellipse_b
(node_num),ellipse_c(node_num),slope(node_num));
%          plot(x,y,'k.',x(aff_node),y(aff_node),'ro',x(aff_node2),y(aff_node2),'g*')

        result(aff_node) = result(aff_node) + 1;
        if length(water) ~= 0
            result(water(:,1)) = 0;
        end
        if length(barrier) ~= 0
            result(barrier(:,1)) = 0;
        end
        temp_result = result(aff_node);
        aff_coord = node(aff_node,:);

        firefront_x(1:length(x_inbound),j,i) = [x_inbound];
        firefront_y(1:length(y_inbound),j,i) = [y_inbound];

        firefront_xcheck(:,j,i) = firefront_x(:,j,i);
        firefront_ycheck(:,j,i) = firefront_y(:,j,i);

        %plot(x_inbound,y_inbound,'yo')

        for ii = 1:length(aff_node)
            % Calculate the row and column indices
            row_index = round(aff_coord(ii,3)/grid_distance_a);
            col_index = round(aff_coord(ii,2)/grid_distance_b);

            % Check that the indices are positive integers
            if row_index > 0 && col_index > 0
                Z(row_index, col_index, j) = temp_result(ii);
%              else
%                  disp('Warning: Index out of range.');
```

```
            end
        end

        %This for loop is to reset the affect time to 0 for nodes with
        %spread rate = 0
        for p = 1:sum(~isnan(firefront_x(:,j,i)))
            idx = find(nodeproperty(:,2) == firefront_x(p,j,i) & nodeproperty(:,3) ==
firefront_y(p,j,i));
            if nodeproperty(idx,4) == 0;
                firefront_x(p,j,i) = NaN;
                firefront_y(p,j,i) = NaN;
            end
        end


%     firefront_x{i} = nonzeros(firefront_x{i});
%     firefront_y{i} = nonzeros(firefront_y{i});
%     plot(firefront_x{i},firefront_y{i},'b')
    end
end
end
```

fun_postresults

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to find the structure damage condition at current
% time
% Input
%       result - consists all the affected nodes and its current state
%       nodeproperty - consists all the parameters of all nodes
%       time - total duration
%       dt - time step
%       M - horizontal mesh size
%       N - vertical mesh size
%       Z - 3-D matrix stores affected time step for all nodes
%       a - real map length
%       b - real map width
%       firefront_x - x coordinate for firefront nodes
%       firefront_y - y coordinate for firefront nodes
%       firefront_xcheck - x coordinate for firefront nodes without ROS = 0
%       firefront_ycheck - y coordinate for firefront nodes without ROS = 0
%       imagename - testing image name (used for barrier testing)
%
% Output
%       M - horizontal mesh size
%       N - vertical mesh size
%       Z - 3-D matrix stores affected time step for all nodes
%       nodeproperty - consists all the parameters of all nodes
%                   - [node_num, x_co, y_co, ROS, node_type, tributary_cost, affected
time, retension factor, repair cost]
%
%       figure(11) - firefront propogation
```

```
%      figure(12) - fire spread simulation
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [M N Z nodeproperty] = fun_postresults(result, nodeproperty, time, dt, M, N,
Z, a, b, firefront_x, firefront_y, firefront_xcheck, firefront_ycheck, imagename)


temp_dt = 0;
[structure_prop str_xco str_yco str_time] = fun_damagelevel(result, nodeproperty,
dt);  %figure7,8,9

%convert feet to mile for plot
a_mile = a * 0.000189394;
b_mile = b * 0.000189394;
M_mile = M .* 0.000189394;
N_mile = N .* 0.000189394;
firefront_x = firefront_x .* 0.000189394;
firefront_y = firefront_y .* 0.000189394;
firefront_xcheck = firefront_xcheck .* 0.000189394;
firefront_ycheck = firefront_ycheck .* 0.000189394;

%firefront plot
figure(11)
I = imread(imagename);
h = image(linspace(0,a_mile,a),linspace(0,b_mile,b),flipud(I));
set(gca,'YDir','normal');
xlabel('x Distance (mile)')
ylabel('y Distance (mile)')
hold on

for j = 1:time/dt + 1
    title(['Time = ' num2str(temp_dt * 60) ' minutes'])
    temp_dt=temp_dt+dt;

    for i = 1:size(firefront_x,3)
        ff_x = rmmissing(firefront_x(:,j,i));
        ff_y = rmmissing(firefront_y(:,j,i));
        ff_x2 = rmmissing(firefront_xcheck(:,j,i));
        ff_y2 = rmmissing(firefront_ycheck(:,j,i));
        figure(11)
        plot(ff_x,ff_y,'b',ff_x2,ff_y2,'g*')
    end
    pause(0.5)

end

hold off

%damage level plot                                                    %figure 3
figure()
%load background image
I = imread(imagename);
h = image(linspace(0,a_mile,a),linspace(0,b_mile,b),flipud(I));
```

```
% h = image(linspace(0,a,a),linspace(0,b,b),flipud(I));

set(gca,'YDir','normal');
xlabel('x Distance (mile)')
ylabel('y Distance (mile)')
hold on

%replace all 0s with nan in Z to show background figure
xlim([0 a])
ylim([0 b])
Z(Z == 0) = NaN;

% begin to write a video for simulation
aviobj=VideoWriter('test.avi');
open(aviobj);
ax = gca();

temp_dt = 0;
%X_MAX = time/dt;
for j = 1:time/dt + 1

%     h = surface(M,N,Z{j});
%     h = surf(M,N,Z{j});
    h = contourf(M_mile,N_mile,Z(:,:,j),'LineColor','none');
%     h = contourf(M,N,Z(:,:,j),'LineColor','none');


%     set(h, 'LineColor', 'none');
    xlim([0 a])
    ylim([0 b])
    shading interp

    drawnow();
    axis tight

    title(['Time = ' num2str(temp_dt * 60) ' minutes'])
    temp_dt=temp_dt+dt;


    %currFrame = getframe;
    writeVideo(aviobj,getframe(ax));
    pause(0.5)
end
hold off
close(aviobj);

structure_visual(str_time, structure_prop, str_xco, str_yco);
[nodeproperty] = economic_loss(nodeproperty, structure_prop, result);
```

Other functions

v_cal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function uses Rothermel's model to find the ROS on all the nodes
% Input
%       h - low heat content
%       S_T - Total mineral content
%       S_e - effective mineral content
%       rho_p - oven-dry particle density
%       sigma - surface-area-to-volume ratio
%       w0 - oven-dry fuel load
%       delta - fuel bed depth
%       M_x - dead fuel moisture of extinction
%       M_f - moisture content
%       U - wind velocity at midflame height
%       slope - wind direction
%
% Output
%       velocity - velocity array contains velocity at that node
%       R_0 - ROS with no wind
%       phi_w - wind factor
%       phi_s - topographic slope factor
%       C - parameter for wind factor
%       B - parameter for wind factor
%       E - parameter for wind factor
%       beta - packing ratio
%       beta_op - optimum packing ratio
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [velocity R_0 phi_w phi_s C B E beta beta_op] =
v_cal(h,S_T,S_e,rho_p,sigma,w0,delta,M_x,M_f,U,slope)

wn = w0 .* (1 - S_T);
rho_b = w0 ./ delta;

beta = rho_b ./ rho_p;
beta_op = 3.348 * sigma .^ -0.8189;
Gamma_maxp = sigma .^ 1.5 .* (495 + 0.0594 * sigma .^ 1.5) .^ -1;
A = 133 * sigma .^ -0.7913;
Gamma_p = Gamma_maxp .* (beta ./ beta_op) .^ A .* exp(A .* (1 - beta ./ beta_op));

r_M = min(M_f ./ M_x, 1);
ita_M = 1 - 2.59 * r_M + 5.11 * (r_M) .^ 2 - 3.52 * (r_M) .^ 3;
ita_S = min(0.174 * S_e .^ -0.19, 1);
I_R = Gamma_p .* wn .* h .* ita_M .* ita_S;
```

```
xi = (192 + 0.2595 * sigma) .^ -1 .* exp((0.792 + 0.681 * sigma .^ 0.5) .* (beta +
0.1));

C = 7.47 * exp(-0.133 * sigma .^ 0.55);
B = 0.02526 * sigma .^ 0.54;
E = 0.715 * exp(-3.59 * 10^-4 * sigma);
phi_w = C .* U .^ B .* (beta ./ beta_op) .^ -E;
phi_s = 5.275 * beta .^ -3 .* tand(slope);

epsilon = exp(-138 ./ sigma);

Q_ig = 250 + 1116 .* M_f;

velocity = I_R .* xi .* (1 + phi_w + phi_s) ./ (rho_b .* epsilon .* Q_ig);
R_0 = I_R .* xi ./ (rho_b .* epsilon .* Q_ig);
```

ellipse_axis2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function calculates the elliptical parameters at each node
% Input
%      R_0 - ROS with no wind
%      phi_w - wind factor
%      phi_s - topographic slope factor
%      C - parameter for wind factor
%      B - parameter for wind factor
%      E - parameter for wind factor
%      beta - packing ratio
%      beta_op - optimum packing ratio
%      slope - wind direction
%      U_wind - wind speed at each node
%      dt - time step
%
% Output
%      ratio - (ellipse_b + ellipse_c) / ellipse_a
%      ellipse_a - interpolated elliptical parameter
%      ellipse_b - interpolated elliptical parameter
%      ellipse_c - interpolated elliptical parameter
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ratio ellipse_a ellipse_b ellipse_c] = ellipse_axis2(R_0, phi_w, phi_s, C,
B, E, beta,...
                                            beta_op, slope, U_wind, dt)
t = dt * 60;
Ds = R_0 .* phi_s * t;
Dw = R_0 .* phi_w * t;
X = Ds + Dw .* cos(slope);
Y = Dw .* sin(slope);

D_h = sqrt(X .^ 2 + Y .^ 2);
```

```
R_h = R_0 + D_h / t;
%effective wind factor
phi_e = R_h ./ R_0 - 1;
%effective wind speed(in ft/min)
U_e = [phi_e .* (beta ./ beta_op) .^ E ./ C] .^ (-B);
%length-to-width ratio (in mi/h)
Z = 1 + 0.25 * U_wind * 0.0113636;
e = (Z .^ 2 - 1) .^ 0.5 ./ Z;
R_b = R_h .* (1 - e) ./ (1 + e);

D_b = R_b * t;

L = D_h + D_b;
W = L ./ Z;

ellipse_a = W / 2;
ellipse_c = D_h - (D_h + D_b) / 2;
ellipse_b = (D_h + D_b) / 2 + ellipse_c;
ratio = (ellipse_b + ellipse_c) ./ ellipse_a;
```

firebrand_distance3D

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function generates extra distance the one firebrand may cause on
% each node
%
% Input
%       nnodes - number of total nodes
%
% Output
%       fb_coord - [firebrand id, x distance, y distance]
%
%       figure(2) - sampling space for 3-D firebrand model
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [fb_coord] = firebrand_distance3D(nnodes)

rng='shuffle';

x = linspace(0,10,100);
y = linspace(-5,5,100);
[X,Y] = meshgrid(x,y);
sigmax = 0.47;
sigmay = 0.92;
mux = 0.88;
pX = 1./(sqrt(2*pi).*sigmax.*X) .* exp(-(log(X) - mux) .^ 2 ./ (2*sigmax^2));
pY = 1./(sqrt(2*pi)*sigmay) * exp(-Y.^2/(2*sigmay^2));
pXY = pX .* pY;
px = 1./(sqrt(2*pi).*sigmax.*x) .* exp(-(log(x) - mux) .^ 2 ./ (2*sigmax^2));
```

```
py = 1./(sqrt(2*pi)*sigmay) * exp(-y.^2/(2*sigmay^2));

figure(2)
surf(X,Y,pXY)
alpha 0.2
hold on

%generate random numbers                                             %check
unit
n = 10000;
xr = 10*rand(n,1);
yr = -5 + 10 .* rand(n,1);
zr = max(pXY(:)) * rand(n,1);
z = linspace(min(pXY(:)),max(pXY(:)),100);
scatter3(xr,yr,zr,'b.')

%filtering usable points
idx = 1;
for i=1:n
    if zr(i) <= (1./(sqrt(2*pi).*sigmax.*xr(i)) .* exp(-(log(xr(i)) - mux) .^ 2 ./
(2*sigmax^2))) * (1./(sqrt(2*pi)*sigmay) * exp(-yr(i).^2/(2*sigmay^2)))
        valid_brand(idx) = i;
        idx = idx + 1;
        scatter3(xr(i),yr(i),zr(i),'r.')
    end
end

title('Mass percentage of firebrand vs. distance')
xlabel('xDistance (m)')
ylabel('yDistance (m)')
zlabel('Mass percentage (%)')

brand_x = xr(valid_brand);
brand_y = yr(valid_brand);

%convert from meters to feet
brand_x = 3.28 * brand_x;
brand_y = 3.28 * brand_y;

%output
spot_co = [brand_x, brand_y];
hold off
fb_id = randi(length(spot_co(:,1)),nnodes,1);
fb_coord = spot_co(fb_id,:);
end
```

findwater

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function finds and marks the water nodes on the map
%
% Input
```

```
%      RGB - image RGB [rox by col by 3]
%
% Output
%      BW - segmentation mask
%      maskedRGBImage - composite of the mask and original RGB image
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [BW,maskedRGBImage] = findwater(RGB)
%createMask  Threshold RGB image using auto-generated code from colorThresholder app.
%  [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB using
%  auto-generated code from the colorThresholder app. The colorspace and
%  range for each channel of the colorspace were set within the app. The
%  segmentation mask is returned in BW, and a composite of the mask and
%  original RGB images is returned in maskedRGBImage.

% Auto-generated by colorThresholder app on 17-Oct-2022
%------------------------------------------------------


% Convert RGB image to chosen color space
I = rgb2hsv(RGB);

% Define thresholds for channel 1 based on histogram settings
channel1Min = 0.547;
channel1Max = 0.636;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 0.530;
channel2Max = 0.930;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 0.235;
channel3Max = 0.451;

% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
BW = sliderBW;

% Initialize output masked image based on input image.
maskedRGBImage = RGB;

% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

end
```

findbarrier

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function finds and marks the barrier nodes on the map
%
% Input
%       RGB - image RGB [rox by col by 3]
%
% Output
%       BW - segmentation mask
%       maskedRGBImage - composite of the mask and original RGB image
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [BW,maskedRGBImage] = findbarrier(RGB)
%createMask  Threshold RGB image using auto-generated code from colorThresholder app.
%  [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB using
%  auto-generated code from the colorThresholder app. The colorspace and
%  range for each channel of the colorspace were set within the app. The
%  segmentation mask is returned in BW, and a composite of the mask and
%  original RGB images is returned in maskedRGBImage.

% Auto-generated by colorThresholder app on 17-Oct-2022
%------------------------------------------------------


% Convert RGB image to chosen color space
I = RGB;

% Define thresholds for channel 1 based on histogram settings
channel1Min = 223.000;
channel1Max = 255.000;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 24.000;
channel2Max = 54.000;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 45.000;
channel3Max = 71.000;

% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
BW = sliderBW;

% Initialize output masked image based on input image.
maskedRGBImage = RGB;

% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

end
```

generate_ellipse2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to find the structure damage condition at current
% time
% Input
%       x_global - global x coordinate for all nodes
%       y_global - global y coordinate for all nodes
%       node - [node_num x_co y_co]
%       aff_node - nodes with time step > 0 at current time step
%       center_x - firefront x coordinate at current step
%       center_y - firefront y coordinate at current step
%       ellipse_a - elliptical parameter
%       ellipse_b - elliptical parameter
%       ellipse_c - elliptical parameter
%       slope - wind direction
%
% Output
%       x_e - sub-ellipse x coordinate at next time step
%       y_e - sub-ellipse y coordinate at next time step
%       x_inbound - x coordinate of nodes encircled by the boundary of sub-ellipses
%       y_inbound - y coordinate of nodes encircled by the boundary of sub-ellipses
%       aff_node - nodes with time step > 0 at next time step
%
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [x_e,y_e,x_inbound,y_inbound,aff_node] =
generate_ellipse2(x_global,y_global,node,aff_node,center_x,center_y,ellipse_a,ellipse
_b,ellipse_c,slope)

x_e = [];
y_e = [];
n = 1000;
for i = 1:length(center_x)
    s = linspace(0,2*pi, n + 1);
    s = s(1:end-1);
    a = ellipse_a(i);                                                       %minor
ellipse axis
    b = ellipse_b(i);                                                       %major
ellipse axis
    c = ellipse_c(i);                                                       %distance
from center
    alpha = slope(i);
    % center_x = 0;
    % center_y = 0;
    % dt = 1;
    % xs = 2;
    % ys = 1;

    %original ellipse
    x = b*cos(s);                                                           %x&y used
to draw the new ellipse
    y = a*sin(s);
```

```
    xp = c*cos(alpha);                                                      %xp&yp
used to displace the new ellipse according to local coordinates
    yp = c*sin(alpha);
    %rotate an ellipse
    %alpha = pi/6;
    R  = [cos(alpha) -sin(alpha);                                           %rotating
matrix
          sin(alpha)  cos(alpha)];
    rCoords = R*[x ; y];                                                    %used to
rotate all the points on the ellipse
    xr = rCoords(1,:)';
    yr = rCoords(2,:)';
%     x_e = [x_e; NaN];
%     y_e = [y_e; NaN];
    x_e = [x_e; xr + center_x(i) + xp];                                     %collect
global coordinates for all points on the sub-ellipse
    y_e = [y_e; yr + center_y(i) + yp];
end

kk = boundary(x_e,y_e,1);

% [in,on] = inpolygon(x_global,y_global,x_e(kk),y_e(kk));
% x_out = [x_global(in); x_global(on)];
% y_out = [y_global(in); y_global(on)];

[in,on] = inpoly2([x_global y_global], [x_e(kk) y_e(kk)]);
x_out = [x_global(in); x_global(on)];
y_out = [y_global(in); y_global(on)];


for i = 1:length(x_out)
    aff_node(i) = find(node(:,2) == x_out(i) & node(:,3) == y_out(i));
end

k = boundary(x_out,y_out,1);
% x_inbound = x_e(k);
% y_inbound = y_e(k);
if isempty(k) ~= 1
    x_inbound = x_out(k);
    y_inbound = y_out(k);
else
    x_inbound = x_out;
    y_inbound = y_out;
end

x_e = x_e(~isnan(x_e));
y_e = y_e(~isnan(y_e));

% figure(32)
% plot(x_e,y_e,'r.',x_e(kk),y_e(kk),'g+',x_out(k),y_out(k),'b*',x_out,y_out,'k.')
% axis equal
% hold on

end
```

inpoly2

```
function [STAT,BNDS] = inpoly2(varargin)
%INPOLY2 compute "points-in-polygon" queries.
%   [STAT] = INPOLY2(VERT,NODE,EDGE) returns the "inside/ou-
%   tside" status for a set of vertices VERT and a polygon
%   {NODE,EDGE} embedded in a two-dimensional plane. General
%   non-convex and multiply-connected polygonal regions can
%   be handled. VERT is an N-by-2 array of XY coordinates to
%   be tested. STAT is an associated N-by-1 logical array,
%   with STAT(II) = TRUE if VERT(II,:) is an interior point.
%   The polygonal region is defined as a piecewise-straight-
%   line-graph, where NODE is an M-by-2 array of polygon ve-
%   rtices and EDGE is a P-by-2 array of edge indexing. Each
%   row in EDGE represents an edge of the polygon, such that
%   NODE(EDGE(KK,1),:) and NODE(EDGE(KK,2),:) are the coord-
%   inates of the endpoints of the KK-TH edge. If the argum-
%   ent EDGE is omitted it assumed that the vertices in NODE
%   are connected in ascending order.
%
%   [STAT,BNDS] = INPOLY2(..., FTOL) also returns an N-by-1
%   logical array BNDS, with BNDS(II) = TRUE if VERT(II,:)
%   lies "on" a boundary segment, where FTOL is a floating-
%   point tolerance for boundary comparisons. By default,
%   FTOL = EPS ^ 0.85.
%
%   See also INPOLYGON

%   This algorithm is based on a "crossing-number" test, co-
%   unting the number of times a line extending from each
%   point past the right-most region of the polygon interse-
%   cts with the polygonal boundary. Points with odd counts
%   are "inside". A simple implementation requires that each
%   edge intersection be checked for each point, leading to
%   O(N*M) complexity...
%
%   This implementation seeks to improve these bounds:
%
% * Sorting the query points by y-value and determining can-
%   didate edge intersection sets via binary-search. Given a
%   configuration with N test points, M edges and an average
%   point-edge "overlap" of H, the overall complexity scales
%   like O(M*H + M*LOG(N) + N*LOG(N)), where O(N*LOG(N))
%   operations are required for sorting, O(M*LOG(N)) operat-
%   ions required for the set of binary-searches, and O(M*H)
%   operations required for the intersection tests, where H
%   is typically small on average, such that H << N.
%
% * Carefully checking points against the bounding-box asso-
%   ciated with each polygon edge. This minimises the number
%   of calls to the (relatively) expensive edge intersection
%   test.

%   Darren Engwirda : 2017 --
```

```
%   Email           : d.engwirda@gmail.com
%   Last updated    : 19/12/2020

%---------------------------------------------- extract args
    node = []; edge = []; vert = [];

    fTOL = eps ^ .85;

    if (nargin>=+1), vert = varargin{1}; end
    if (nargin>=+2), node = varargin{2}; end
    if (nargin>=+3), edge = varargin{3}; end
    if (nargin>=+4), fTOL = varargin{4}; end

%---------------------------------------------- default args
    nnod = size(node,1) ;
    nvrt = size(vert,1) ;

    if (isempty(edge))
        edge = [(1:nnod-1)',(2:nnod)'; nnod,1];
    end

%---------------------------------------------- basic checks
    if ( ~isnumeric(node) || ...
         ~isnumeric(edge) || ...
         ~isnumeric(vert) || ...
         ~isnumeric(fTOL) )
        error('inpoly2:incorrectInputClass' , ...
            'Incorrect input class.') ;
    end

%---------------------------------------------- basic checks
    if (ndims(node) ~= +2 || ...
        ndims(edge) ~= +2 || ...
        ndims(vert) ~= +2 || ...
        numel(fTOL) ~= +1 )
        error('inpoly2:incorrectDimensions' , ...
            'Incorrect input dimensions.');
    end
    if (size(node,2)~= +2 || ...
        size(edge,2)~= +2 || ...
        size(vert,2)~= +2 )
        error('inpoly2:incorrectDimensions' , ...
            'Incorrect input dimensions.');
    end

%---------------------------------------------- basic checks
    if (min([edge(:)]) < +1 || ...
            max([edge(:)]) > nnod)
        error('inpoly2:invalidInputs', ...
            'Invalid EDGE input array.') ;
    end

    STAT = false(size(vert,1),1) ;
    BNDS = false(size(vert,1),1) ;
```

```
%----------------------------------- prune points using bbox
    nmin = min(node,[],1);
    nmax = max(node,[],1);
    ddxy = nmax - nmin ;

    lbar = sum(ddxy) / 2.;

    veps = fTOL * lbar ;

    mask = ...
    vert(:,1) >= min (node(:,1)) - veps & ...
    vert(:,1) <= max (node(:,1)) + veps & ...
    vert(:,2) >= min (node(:,2)) - veps & ...
    vert(:,2) <= max (node(:,2)) + veps ;

    vert = vert(mask, :) ;

    if (isempty(vert)), return; end

%-------------- flip to ensure the y-axis is the "long" axis
    vmin = min(vert,[],1);
    vmax = max(vert,[],1);
    ddxy = vmax - vmin ;

    if (ddxy(1) > ddxy(2))
    vert = vert(:,[2,1]) ;
    node = node(:,[2,1]) ;
    end

%---------------------------------- sort points via y-value
    swap = ...
       node(edge(:,2),2) ...
     < node(edge(:,1),2) ;

    edge(swap,[1,2]) = ...
        edge(swap,[2,1]) ;

   [~,ivec] = ...
        sort(vert(:,+2)) ;
    vert = vert (ivec,:) ;

    if (exist( ...
        'OCTAVE_VERSION','builtin')  > +0)

    if (exist('inpoly2_oct','file') == +3)

    %-- delegate to the compiled version of the code if it's
    %-- available

       [stat,bnds] = ...
            inpoly2_oct( ...
                vert,node,edge,fTOL,lbar) ;

    else
```

```
    %-- otherwise, just call the native m-code version

       [stat,bnds] = ...
            inpoly2_mat( ...
                vert,node,edge,fTOL,lbar) ;

    end

    else

    %-- MATLAB's JIT is generally smart enough these days to
    %-- run this efficiently

       [stat,bnds] = ...
            inpoly2_mat( ...
                vert,node,edge,fTOL,lbar) ;

    end

    stat(ivec) = stat ;
    bnds(ivec) = bnds ;

    STAT(mask) = stat ;
    BNDS(mask) = bnds ;

end
```

fun_damagelevel

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to assess the damage of structure components
%
% Input
%       result - consists all the affected nodes and its affected time step
%       nodeproperty - consists all the parameters of all nodes
%       dt - time step
%
% Output
%       structure_prop - consists the parameters for all structure nodes
%       str_xco - x coordinate for all structures
%       str_yco - y coordinate for all structures
%       str_time - affected time step for all structures
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [str_property str_xco str_yco str_time] = fun_damagelevel(result,
nodeproperty, dt)

str_idx = find(nodeproperty(:,5) == 4 | nodeproperty(:,5) == 5 | nodeproperty(:,5) ==
6);
str_type = nodeproperty(str_idx,5);
```

```
str_xco = nodeproperty(str_idx,2);                                   %extract
structure coordinates
str_yco = nodeproperty(str_idx,3);
str_time = result(str_idx);

%retrieve percent failure model
[p_concrete, p_steel, T, p_wood, T_wood] = probabilistic_model;
%from time to compartment temperature
%use the ISO 834 standard fire curve to estimate the temperature
T0 = 20;
T_curve = T0 + 345 * log10(8 * dt * str_time + 1);
percent_failure = zeros(length(str_idx),1);
for i = 1:length(str_idx)
    if str_type(i) == 4
        percent_failure(i) = interp1(T, p_concrete, T_curve(i));
    elseif str_type(i) == 5
        percent_failure(i) = interp1(T, p_steel, T_curve(i));
    elseif str_type(i) == 6
        if T_curve(i) > 350
            percent_failure(i) = 1;
        else
            percent_failure(i) = interp1(T_wood, p_wood, T_curve(i));
        end
    end
end

str_property = [str_idx str_xco str_yco str_type str_time
percent_failure];                %assemble structure node and ignition time
```

probabilistic_model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to generate the probabilistic model and fragility
% curves for different material
%
% Input
%
% Output
%      p_concrete - probabilistic model for concrete
%      p_steel - probabilistic model for steel
%      T - time for steel and concrete probabilistic model
%      p_wood - probabilistic model for wood
%      T_wood - time for wood probabilistic model
%
%      figure(7) - concrete strength reduction probabilistic model
%      figure(8) - steel strength reduction probabilistic model
%      figure(9) - wood strength reduction probabilistic model
%      figure(10) - fragility curves for the three materials
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generate temperature-fragility curve
function [p_concrete, p_steel, T, p_wood, T_wood] = probabilistic_model
```

```
[k_concrete, T_concrete] = cal_concrete_function;
[k_steel, T_steel] = cal_steel_function;
[k_wood, T_wood] = cal_wood_function;

T = linspace(20,1000,981);

%generate random normal distribution at each T
%parabola equation, y = a(x-h)^2 + k, (h,k) is the vertex
b_concrete = 0.0002;
std_concrete = (-0.12-1000*b_concrete)/1000000 * T .^ 2 + b_concrete*T + 0.15;
b_steel = 0.00004;
std_steel =(-0.12-1000*b_concrete)/1000000 * T .^ 2 + b_concrete*T + 0.15;
b_wood = 0.00004;
std_wood = -0.8375*10^-6*T_wood .^ 2 + b_wood * T_wood + 0.15;
%plot(T,std)

% std = linspace(0.15, 0.03, 1000);
for i = 1:1:10
    r_concrete(i,:) = normrnd(k_concrete,std_concrete);
    r_steel(i,:) = normrnd(k_steel,std_steel);
    r_wood(i,:) = normrnd(k_wood,std_wood);
%     r_c(i,:) = normrnd(k_cconcrete,std);

    upper_bound_concrete = k_concrete + 2 * std_concrete;
    lower_bound_concrete = k_concrete - 2 * std_concrete;
    upper_bound_steel = k_steel + 2 * std_steel;
    lower_bound_steel = k_steel - 2 * std_steel;
    upper_bound_wood = k_wood + 2 * std_wood;
    lower_bound_wood = k_wood - 2 * std_wood;
%     upper_bound_c = k_cconcrete + 2 * std;
%     lower_bound_c = k_cconcrete - 2 * std;
end

figure(7)
scatter(T,r_concrete(1,:),'.')
hold on
plot(T, k_concrete,'k-', T, upper_bound_concrete,'b--', T, lower_bound_concrete, 'b--
')
title("Concrete strength reduction probabilistic model")
xlabel('Temperature(C)')
ylabel('Reduction factor')
hold off

figure(8)
scatter(T,r_steel(1,:),'.')
hold on
plot(T, k_steel,'k-', T, upper_bound_steel,'b--', T, lower_bound_steel, 'b--')
title("Steel strength reduction probabilistic model")
xlabel('Temperature(C)')
ylabel('Reduction factor')
hold off

figure(9)
scatter(T_wood,r_wood(1,:),'.')
hold on
```

```
plot(T_wood, k_wood,'k-', T_wood, upper_bound_wood,'b--', T_wood, lower_bound_wood,
'b--')
title("wood strength retention factors (mass loss)")
xlabel('Temperature(C)')
ylabel('Retention factor')
hold off

%generate fragility curve
ls = 0.5;
p_concrete = normcdf(ls,k_concrete,std_concrete);
% p_c = normcdf(ls,k_cconcrete,std);
p_steel = normcdf(ls,k_steel,std_steel);
p_wood = normcdf(ls,k_wood,std_wood);

figure(10)
plot(T,p_steel,T,p_concrete,T_wood,p_wood)
legend("steel", "concrete","wood")
title("strength retention factor")
xlabel("Temperature (C)")
ylabel("percentage failure (%)")
```

cal_concrete_function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to generate the regression curve for concrete
% probabilistic model
%
% Input
%
% Output
%       k_concrete - regressed concrete strength retention factor
%       T_concrete - temperatures for concrete strength retention factor
%
%       figure(4) - summary plot for concrete retention factor curves
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [k_concrete, T_concrete] = cal_concrete_function
%Shuna_siliceous_concrete (Weibull)
x1=[21.04097453
    42.08194906
    65.33776301
    91.9158361
    116.2790698
    138.427464
    158.3610188
    182.7242525
    203.765227
    222.5913621
    245.8471761
    266.8881506
    284.606866
```

```
    301.2181617
    320.0442968
    339.9778516
    362.1262458
    379.8449612
    398.6710963
    417.4972315
    436.3233666
    454.0420819
    473.9756368
    492.8017719
    511.627907
    527.1317829
    545.9579181
    561.461794
    575.8582503
    594.6843854
    611.2956811
    634.551495
    657.807309
    676.6334441
    696.5669989
    718.7153931
    735.3266888
    747.5083056
    763.0121816
    780.730897
    799.5570321
    813.9534884
    830.5647841
    846.06866
    856.0354374
    873.7541528
    887.0431894
    895.9025471
    914.7286822
    933.5548173
    953.4883721
    973.4219269];
y1=[1.014256645
    1.001013289
    0.987757475
    0.972607973
    0.95934593
    0.946096346
    0.930984219
    0.915847176
    0.898853821
    0.881872924
    0.86299211
    0.844123754
    0.827149086
    0.815805648
    0.795074751
    0.772462625
```

```
    0.75358804
    0.736613372
    0.714007475
    0.691401578
    0.668795681
    0.651821013
    0.629208887
    0.60660299
    0.587747093
    0.567034884
    0.546303987
    0.519966777
    0.503010797
    0.4841549
    0.457811462
    0.429555648
    0.393799834
    0.367443937
    0.333581811
    0.301582226
    0.273363787
    0.254545266
    0.230083056
    0.207483389
    0.175502492
    0.147296512
    0.126578073
    0.102115864
    0.079559801
    0.055085133
    0.021260382
    0.004335548
    0.004229651
    0.000373754
    0.000261628
    0.000149502];

%Shuna_calcareous_concrete (Weibull)
x2=[11.20350764
    25.6946126
    43.28967746
    62.95838256
    80.55344743
    100.2205274
    126.0939021
    148.8665671
    166.4583817
    187.1557814
    204.7475961
    221.2993404
    241.9999902
    258.5517346
    275.1083542
    295.8008785
    314.4278882
```

```
    329.9428124
    343.3905968
    355.8048113
    371.3229857
    384.7675199
    405.4584191
    419.9381483
    435.4546976
    445.7920217
    464.4109058
    479.9242049
    495.4326286
    510.9443026
    526.4543515
    542.9995953
    556.4425044
    572.9861232
    588.4929218
    606.0701105
    619.5097693
    636.0533881
    649.4897967
    664.9982205
    679.4714493
    690.8423432
    702.2148623
    722.9008862
    738.4044346
    751.8424683
    765.2805021
    778.7250363
    792.1695705
    810.780329
    828.3591428
    842.8372469
    856.2801559
    872.8253998
    896.6218841
    915.2423933
    930.7573175
    945.2402969
    958.6897064
    973.1743109
    988.6859849
    1000.068255];
y2=[0.98898988
    0.987078751
    0.983277944
    0.983240892
    0.979440085
    0.977519205
    0.969935142
    0.968008412
    0.96043995
    0.952865638
```

```
    0.945297176
    0.932079182
    0.928272524
    0.915054531
    0.907488019
    0.894262224
    0.886691813
    0.871591941
    0.860263625
    0.850821087
    0.83948887
    0.8243929
    0.809283278
    0.794185357
    0.780969313
    0.763995365
    0.747005816
    0.730022118
    0.707386937
    0.688519411
    0.667768058
    0.647014754
    0.630034956
    0.607397825
    0.582878817
    0.558355909
    0.537608456
    0.514971325
    0.490456217
    0.467821036
    0.445187806
    0.42632808
    0.409352182
    0.388591078
    0.360304416
    0.337673135
    0.315041855
    0.299945884
    0.284849913
    0.258441227
    0.235802146
    0.218820398
    0.201840599
    0.181087296
    0.165971824
    0.150866102
    0.135766231
    0.124435965
    0.114991476
    0.105545037
    0.086677512
    0.081004578];

%high_performance_concrete_CRC
x3=[18.50285326
```

```
    97.4477592
    201.1108627
    299.3038318
    497.9750056
    698.5650105
    1000];
y3=[1.003435181
    1.271711146
    1.338544041
    1.205969947
    0.904382759
    0.390551279
    0];

%high_performance_concrete_C70
x4=[18.50285326
    99.51991642
    200.0172868
    298.1426121
    498.4575319
    598.3438526
    698.0272417
    1000];
y4=[1.003435181
    0.986573068
    1.15844825
    1.058033179
    0.674981727
    0.437360085
    0.296215513
    0];

%high_performance_concrete_RPC-AF
x5=[18.50285326
    98.86940787
    199.6463729
    298.9193892
    395.2792095
    598.9143159
    699.6360384
    1000];
y5=[1.003435181
    0.945835674
    0.984786894
    1.038740396
    1.077675832
    0.866152323
    0.581366744
    0];

%high_performance_concrete_C60-SF
x6=[18.50285326
    97.24370019
    200.1525745
    397.5001832
```

```
    499.509394
    599.2739558
    699.4996233
    1000];
y6=[1.003435181
    1.018724199
    1.094130203
    0.721787903
    0.524908916
    0.345173516
    0.296220775
    0];

%generate spline and interpolation
x_spline = linspace(20,1000,981);
y1_spline = spline(x1,y1,x_spline);
y2_spline = spline(x2,y2,x_spline);
y3_spline = spline(x3,y3,x_spline);
y4_spline = spline(x4,y4,x_spline);
y5_spline = spline(x5,y5,x_spline);
y6_spline = spline(x6,y6,x_spline);

%find average
for i = 1:length(x_spline)
    avg_concrete(i) = 1/6*(y1_spline(i) + y2_spline(i) + y3_spline(i) + y4_spline(i)
+ y5_spline(i) + y6_spline(i));
end

%plot
figure(4)
plot(x1,y1,':',x2,y2,':',x3,y3,':*',x4,y4,':^',x5,y5,':+',x6,y6,':o')
hold on
plot(x_spline,avg_concrete)

h = polyfit(x_spline,avg_concrete,4);
y_val = polyval(h,x_spline);
plot(x_spline,y_val,'k-')

title('Reduction factor for concrete')
xlabel('Temperature(C)')
ylabel('Reduction factor')
legend('Shuna siliceous concrete (Weibull)','Shuna calcareous concrete (Weibull)',...
    'high performance concrete CRC','high performance concrete C70',...
    'high performance concrete RPC-AF','high performance concrete C60-SF',...
    'avg concrete','concrete fitted avg')
k_concrete = avg_concrete;
T_concrete = x_spline;
```

cal_steel_function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to generate the regression curve for steel
```

```
% probabilistic model
%
% Input
%
% Output
%      k_steel - regressed steel strength retention factor
%      T_steel - temperatures for steel strength retention factor
%
%      figure(5) - summary plot for steel retention factor curves
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [k_steel, T_steel] = cal_steel_function
%steel_EC3
x1=[28.99761337
    96.65871122
    151.4319809
    207.2792363
    273.8663484
    379.1169451
    403.8186158
    419.928401
    431.7422434
    459.6658711
    494.0334129
    529.4749403
    559.5465394
    587.4701671
    612.1718377
    634.725537
    665.8711217
    691.646778
    708.8305489
    755.0119332
    794.7494033
    848.4486874
    897.8520286
    1000];
y1=[1
    1.002713704
    1.002713704
    1.002713704
    1.004070556
    1.001356852
    0.981004071
    0.949796472
    0.921302578
    0.865671642
    0.789687924
    0.689280868
    0.591587517
    0.506105834
    0.439620081
    0.382632293
```

```
    0.312075984
    0.256445047
    0.215739484
    0.165535957
    0.111261872
    0.082767978
    0.05834464
    0];

%steel_wang
x2=[22.27163336
    54.10170556
    91.2341739
    129.6955111
    160.2019462
    190.7031495
    221.2069687
    241.1007638
    284.8644972
    348.5115622
    401.5302919
    455.8752747
    499.5866904
    533.9907921
    569.7132992
    610.678037
    643.7192634
    672.7869624
    717.7749294
    760.1182379
    789.2251753
    846.2122005
    897.9046772
    1000];
y2=[0.949007011
    0.948856336
    0.946313697
    0.946131631
    0.948354086
    0.945842838
    0.945698441
    0.945604269
    0.94303024
    0.930894632
    0.902241289
    0.873581668
    0.823670608
    0.752501831
    0.674226222
    0.539121063
    0.434822643
    0.33527676
    0.240389243
    0.152614837
    0.088571728
```

```
    0.050432144
    0.021785079
    0];

%steel_Outinen
x3=[19.93840022
    100.8554087
    200.7078799
    300.5629392
    399.5574133
    501.2772896
    600.4154103
    701.2747014
    800.2665873
    900.9861142
    950.9142909
    999.9792942];
y3=[0.997894737
    0.967218045
    0.932932331
    0.895037594
    0.857142857
    0.618947368
    0.38075188
    0.142556391
    0.108270677
    0.064962406
    0.045112782
    0.02887218];

%find EC3 retention factor
EC3_points = [63.51999999999998, 1.003264942450914
100.97142857142853, 1.0033536899119835
140.1904761904762, 1.0034466260437822
181.0666666666666, 1.0035434890543893
221.11428571428564, 1.0036383886255922
264.3657142857143, 1.0037408801624914
327.99999999999994, 1.0038916723087339
373.29523809523806, 1.0039990069961633
425.77142857142854, 0.9471836154366959
474.6571428571428, 0.83931029113067
516.9142857142856, 0.737311577522004
553.3714285714284, 0.6195916046039267
576.5714285714284, 0.5450358835477318
621.3142857142857, 0.42340866621530093
673.790476190476, 0.2900191378921235
724.0571428571427, 0.20178348002708169
778.0799999999999, 0.13711799593771135
829.1199999999999, 0.09718477995937702
870.8800000000001, 0.07607859174001308
912.6399999999999, 0.05889928232904529
934.5142857142857, 0.05502423832092074
957.5999999999999, 0.04492322274881477
982.8, 0.036452132701421114
1010.4000000000001, 0.033673933649288834
```

```
1047.6000000000001, 0.028074881516587435
1077.6, 0.025302369668246172
1108.8, 0.016845497630331607
1140, 0.014075829383886074
];



%generate spline and interpolation
x_spline = linspace(20,1000,981);
y1_spline = spline(x1,y1,x_spline);
y2_spline = spline(x2,y2,x_spline);
y3_spline = spline(x3,y3,x_spline);
%x_spline = [20:1:1000];
EC3_r = spline(EC3_points(:,1),EC3_points(:,2),x_spline);
T = linspace(20,1000,981);
epsilon = 0;
%Ts = T - 20;
temp0 = exp(-1/2 * (T/590).^5.708-1/2*(T/919));


%find average
for i = 1:length(x_spline)
    avg_steel(i) = 1/3*(y1_spline(i) + y2_spline(i) + y3_spline(i));
end

%plot
figure(5)
plot(x1,y1,':*',x2,y2,'^:',x3,y3,':o')
hold on
plot(x_spline,avg_steel,'k-')

h = polyfit(x_spline,avg_steel,4);
y_val = polyval(h,x_spline);
% plot(x_spline,y_val,'k--')
plot(EC3_points(:,1),EC3_points(:,2),x_spline,EC3_r)
plot(T,temp0)

title('Reduction factor for steel')
xlabel('Temperature(C)')
ylabel('Reduction factor')
legend('steel EC3','steel wang','steel Outinen','steel avg',"EC3 data", "EC3 interp",
"NIST model")

k_steel = avg_steel;
T_steel = x_spline;
```

cal_wood_function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to generate the regression curve for wood
% probabilistic model
%
```

```
% Input
%
% Output
%      k_wood - regressed wood strength retention factor
%      T_wood - temperatures for wood strength retention factor
%
%      figure(6) - summary plot for wood retention factor curves
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [k_wood, T_wood] = cal_wood_function
%first part of wood reduction
%No treatment
x1=[0
40.0454
80.0908
120.1362
160.1816
200.2273031
205.3004874
210.8572192
217.8629678
227.2838525
235.0128086
240.0853425
246.1233456
253.6106906
259.8893293
266.409254
272.445306
276.7907301
282.100973
289.3421542
295.8575264
300.1993735
304.5412206
307.6759874
311.2920254
315.1490243
318.5247519
321.1756459
323.5846037
326.7187201
329.3679882
332.2565913
334.1816763
336.349348
338.5150686
339.9588824
343.0903973
345.4977291
348.6298945
352.4894948
356.3523469
```

```
362.3890493
367.7031944
373.7418478];
y1=[1
0.989
0.978
0.967
0.956
0.945492779
0.943104308
0.941930396
0.938343625
0.931139189
0.921500757
0.91668967
0.908251164
0.898611106
0.886540303
0.873259818
0.857553466
0.844258347
0.824913194
0.798313202
0.768074409
0.741454906
0.714835403
0.691841694
0.66158339
0.628904096
0.603489397
0.57806982
0.55143731
0.526020985
0.49454487
0.454591227
0.425532849
0.40011002
0.367419346
0.345625562
0.310518776
0.277829727
0.245145556
0.222156724
0.211280969
0.197997232
0.193187772
0.187171881];

%2% Na2B4O7.10H2O
x2=[0
40.0454
80.0908
120.1362
160.1816
200.4689143
```

```
203.6098596
210.8575444
219.0710236
227.0432169
235.9792533
243.9491702
250.7113565
256.98967
264.2331275
270.0275683
276.7861775
282.0954449
286.1963312
289.81367
294.1545415
296.8044601
299.9366254
303.3097515
306.1993301
308.8466472
311.2556049
313.4200248
316.5512146
320.409189
326.9274877
334.1728963
341.1773441
348.4237283
356.3939704
364.3658384];
y2=[1
0.989
0.978
0.967
0.956
0.945494405
0.945515542
0.943141704
0.938351754
0.934771486
0.921507261
0.909447839
0.898591595
0.885309484
0.867188646
0.851480668
0.827300039
0.804320963
0.78012245
0.754709376
0.72445595
0.69540245
0.662718279
0.627613119
0.591293398
```

```
0.552549437
0.525916927
0.488381021
0.452062927
0.423017556
0.403680533
0.392827541
0.384395538
0.37717647
0.366328356
0.36153678];

%2% NaCl
x3=[0
40.0454
80.0908
120.1362
160.1816
200.2276283
205.0582259
215.4442546
223.4148219
229.9357221
236.9398447
243.9436421
249.9800193
255.7744601
261.5682506
264.9462544
269.5310135
273.3896382
276.5237546
279.4169103
282.7913371
285.6809157
288.3301839
291.9439456
294.5932138
297.7240784
300.8565689
304.2293698
306.3947653
309.2866202
317.4968475
324.016447
330.5376725
337.7827559
346.4781567
353.0000325];
y3=[1
0.989
0.978
0.967
0.956
0.946704087
```

```
0.940680067
0.928636904
0.919000098
0.909353536
0.899710226
0.888855608
0.874360563
0.858652585
0.840521991
0.823586446
0.8018138
0.775191045
0.74977472
0.726779385
0.696519455
0.660199735
0.62872362
0.589986162
0.558510046
0.520980644
0.489507781
0.453191313
0.41928933
0.391448764
0.374545737
0.360053944
0.35161869
0.339554391
0.329922462
0.323909823];

%2% KHCO3
x4=[0
40
80
120
160
199.9847164
206.023695
216.1677873
227.0354125
234.0382343
240.7981443
246.8338711
253.1089328
258.6604617
264.2103647
268.069965
272.651147
276.2684858
280.1258098
283.2569996
286.1459278
290.0013007
294.0992604
```

```
296.7488537
299.8819946
308.3289554
315.5714373
324.5068234
332.7170508
339.2382762
348.8997966
357.8364834];
y4=[1
0.988
0.977
0.9652
0.9536
0.94185723
0.937052647
0.923796551
0.905700102
0.891211561
0.871876163
0.854958503
0.829563315
0.80900848
0.782397107
0.759408275
0.724311244
0.698898171
0.667430185
0.631112091
0.592369755
0.553633923
0.51853364
0.488268833
0.459218585
0.424147568
0.402392807
0.386705966
0.369802939
0.361367685
0.350530952
0.339689342];

%2% AlCl3.6H2O
x5=[0
40.0431
80.0861
120.1292
160.1722
200.2152713
205.771678
212.7771013
218.0912464
226.0621389
234.0327062
240.5526309
```

```
248.0390003
255.5230934
262.0407419
269.0412874
273.6260465
277.9691943
281.3439464
284.4764369
287.8479371
291.462024
295.3170718
297.7224525
301.3362142
304.9506263
306.8760365
310.2501382
315.317144
320.3867513
327.1463361
332.6998161
341.8764882
344.7748467];
y5=[1
0.9904
0.9808
0.9712
0.9616
0.900674394
0.898289175
0.893491095
0.888681635
0.880256136
0.87061933
0.857338845
0.844064863
0.822311728
0.800552089
0.777584394
0.755811748
0.734037476
0.704988853
0.67351599
0.632354291
0.594828141
0.554881001
0.514924106
0.476186648
0.439871806
0.412024736
0.380553498
0.355150181
0.339437325
0.318890619
0.305603631
0.288707107
```

```
0.285092695];


%generate spline and interpolation
x_spline = linspace(20,400,500);
y1_spline = spline(x1,y1,x_spline);
y2_spline = spline(x2,y2,x_spline);
y3_spline = spline(x3,y3,x_spline);
y4_spline = spline(x4,y4,x_spline);
y5_spline = spline(x5,y5,x_spline);

%find average
for i = 1:length(x_spline)
    avg_wood_1(i) = 1/5*(y1_spline(i) + y2_spline(i) + y3_spline(i) + y4_spline(i) +
y5_spline(i));
end

%plot
figure(6)
plot(x1,y1,':',x2,y2,':',x3,y3,':',x4,y4,':',x5,y5,':')
hold on
plot(x_spline,avg_wood_1,'r-')


%%
%second part of wood reduction
%Janssens (1997)
data1 = [20.60085837      1.002606511
53.64806867  0.966289822
81.11587983  0.95675704
109.8712446  0.92772253
137.7682403  0.915754213
162.2317597  0.898879933
190.5579399  0.874719983
218.0257511  0.845675006
225.751073   0.762810984
233.4763948  0.655556719
248.0686695  0.504455843
259.6566524  0.367964688
271.2446352  0.253424753
282.4034335  0.126686207
293.1330472  0.00482222];

%White et al. (1993)
data2 = [19.74248927      1.002599532
29.61373391  1.012435884
43.34763948  1.010108517
55.36480687  0.998011096
70.81545064  0.995697687
83.26180258  0.990920828
96.99570815  0.981276388
109.0128755  0.942349698
117.167382   0.908269654
128.3261803  0.869335985
142.4892704  0.798719425
```

```
150.2145923  0.774391989
158.3690987  0.742750968
169.9570815  0.703820789
181.9742489  0.672211173
190.5579399  0.643012666
203.4334764  0.611410028
213.304721   0.562709794
228.7553648  0.460396385
236.0515021  0.423870337
244.6351931  0.360525489
250.6437768  0.341062145];

%Ostman (1985)
data3 = [19.74248927      1.002599532
28.32618026  0.985596148
42.91845494  0.956446492
51.93133047  0.941885621
67.38197425  0.912742943
81.97424893  0.888471335
94.84978541  0.871502844
109.4420601  0.854548309
123.6051502  0.84002931
139.9141631  0.820649709
153.6480687  0.80612722
170.3862661  0.796507205
183.2618026  0.779538714
194.8497854  0.76012073
207.7253219  0.740713214
219.3133047  0.728612303
225.751073   0.72134757
230.472103   0.692117659
236.0515021  0.650699606
241.2017167  0.61903416
246.7811159  0.592250253
252.360515   0.563027321];

%ASCE (1992)
data4 = [20.17167382      1.005042046
30.90128755  0.990495132
38.62660944  0.983240867
50.21459227  0.976018005
66.52360515  0.97127255
84.97854077  0.956788443
96.13733906  0.944684043
108.1545064  0.932586622
120.1716738  0.927806274
132.1888412  0.918147877
142.4892704  0.908475523
152.7896996  0.901242193
160.5150215  0.891548903
169.0987124  0.884301616
175.9656652  0.869723298
182.4034335  0.855141491
187.9828326  0.833235633
194.4206009  0.80401968
```

```
200.4291845  0.772361213
204.72103    0.738249764
209.8712446  0.696828222
216.3090129  0.635904951
220.6008584  0.587159357
224.0343348  0.543284832
227.8969957  0.492096724
230.9012876  0.450657734
235.193133   0.409229212
240.3433476  0.355612548
244.6351931  0.319062075
247.639485   0.275184061
250.2145923  0.24349768];

%Schaffer (1984)
data5 = [19.31330472      1.000157019
25.32188841  0.980693674
35.19313305  0.958822708
42.06008584  0.944244391
51.93133047  0.937007572
60.51502146  0.929760285
67.8111588   0.920063505
75.10729614  0.91280575
80.68669528  0.903095014
89.27038627  0.898286751
97.85407725  0.895917513
105.5793991  0.893541296
112.8755365  0.883844517
123.1759657  0.881489235
132.6180258  0.871809903
140.7725322  0.864559126
148.9270386  0.85730835
156.223176   0.852489619
163.0901288  0.852545448
170.3862661  0.845287693
178.5407725  0.835597892
183.2618026  0.811246031
186.695279   0.779566628
190.5579399  0.750329739
193.9914163  0.718650337
197.4248927  0.682092885
202.1459227  0.638228829
206.0085837  0.594357793
210.7296137  0.552932761
213.7339056  0.509054747
217.167382   0.470058271
220.1716738  0.445692453
224.4635193  0.396946858
227.0386266  0.360382428
230.0429185  0.316504414
234.3347639  0.277514917
239.4849785  0.243410447
243.776824   0.211738023
247.2103004  0.170302523];
```

```
%generate spline and interpolation
x_spline_2 = linspace(20,240,500);
y21_spline = spline(data1(:,1),data1(:,2),x_spline_2);
y22_spline = spline(data2(:,1),data2(:,2),x_spline_2);
y23_spline = spline(data3(:,1),data3(:,2),x_spline_2);
y24_spline = spline(data4(:,1),data4(:,2),x_spline_2);
y25_spline = spline(data5(:,1),data5(:,2),x_spline_2);

%find average
for i = 1:length(x_spline_2)
    avg_wood_2(i) = 1/5 * (y21_spline(i) + y22_spline(i) + y23_spline(i)...
        + y24_spline(i) + y25_spline(i));
end
%extend avg_2 to 0 and keeps 0 until 400 C
temp = linspace(280,400,300);
avg_wood_2 = [avg_wood_2,zeros(1,length(temp))];
x_spline_2 = [x_spline_2, temp];

%plot
plot(data1(:,1),data1(:,2),'-.',data2(:,1),data2(:,2),'-.',...
    data3(:,1),data3(:,2),'-.',data4(:,1),data4(:,2),'-.',...
    data5(:,1),data5(:,2),'-.')

plot(x_spline_2,avg_wood_2,'r-')

% x = [x_spline, x_spline_2];
% y = [avg_wood_1, avg_wood_2];

% h = polyfit(x,y,3);
% y_val = polyval(h,x_spline);
% plot(x_spline,y_val,'k-')
avg_wood_1_spline = spline(x_spline, avg_wood_1, x_spline_2);
y_avg = (avg_wood_1_spline + avg_wood_2) ./ 2;
plot(x_spline_2, y_avg, 'k-')

title('Mass loss for wood')
xlabel('Temperature(C)')
ylabel('Mass loss')
legend('No treatment','2% Na2B4O7.10H2O',...
    '2% NaCl','2% KHCO3    ',...
    '2% AlCl3.6H2O', 'average_1',...
    'Janssens (1997)','White et al. (1993)',...
    'Ostman (1985)', 'ASCE (1992)', ...
    'Schaffer (1984)', 'average_2',...
    'wood fitted avg')
hold off

T_wood = linspace(20,400,380);
k_wood = spline(x_spline_2, y_avg, T_wood);
```

structure_visual

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to plot the damage state and location of
% structures on top of the simulation
%
% Input
%       str_time - affected time step for all structures
%       structure_prop - consists the parameters for all structure nodes
%       str_xco - x coordinate for all structures
%       str_yco - y coordinate for all structures
%
% Output
%       figure(12) - add a layer of structures to the simulation
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function structure_visual(str_time, structure_prop, str_xco, str_yco)

%convert feet into mile
str_xco = str_xco .* 0.000189394;
str_yco = str_yco .* 0.000189394;

width = ones(length(str_xco), 1) * 300 * 0.000189394;
height = ones(length(str_xco), 1) * 220 * 0.000189394;
position = [str_xco - width/2, str_yco - height/2, width, height];

%define the color array
color = (str_time - min(str_time)) ./ (max(str_time) - min(str_time));
% Initialize color array
color_array = zeros(length(color), 3);

% Define the mid-point for the transition between red and dark red
mid_point = mean(color);

for i = 1:length(color)
    if mid_point == 0
        color_array = zeros(length(color), 3);
    else
        if color(i) <= mid_point
            % Interpolate from blue (0, 0, 1) to red (1, 0, 0)
            interp_factor = color(i) / mid_point;
            color_array(i, :) = [interp_factor, 0, 1 - interp_factor];
        else
            % Interpolate from red (1, 0, 0) to dark red (0.5, 0, 0)
            interp_factor = (color(i) - mid_point) / (1 - mid_point);
            color_array(i, :) = [(1 - interp_factor*0.5), 0, 0];
        end
    end
end

% Plot rectangles
figure(12)
hold on
for i = 1:length(str_xco)
    if structure_prop(i,4) == 4
        box_color = 'black';
```

```
    elseif structure_prop(i,4) == 5
        box_color = 'magenta';
    elseif structure_prop(i,4) == 6
        box_color = 'white';
    end
    h(i) = rectangle('Position', position(i,:), 'EdgeColor', box_color, 'FaceColor',
color_array(i, :));
end
hold off
```

economic_loss

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to find the economic loss for each iteration
%
% Input
%      nodeproperty - consists all the parameters of all nodes
%                   - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%      structure_prop - consists the parameters for all structure nodes
%      result - consists all the affected nodes and its affected time step
%
% Output
%      nodeproperty - consists all the parameters of all nodes
%                   - [node_num, x_co, y_co, ROS, node_type, tributary_cost, affected
time, retension factor, repair cost]
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [nodeproperty] = economic_loss(nodeproperty, structure_prop, result)

str_cost_ratio = 0.4;
nonstr_cost_ratio = 0.6;
percent_failure = structure_prop(:,6);
nodeproperty = [nodeproperty result zeros(length(result),
2)];               %initialize repair cost ratio and repair cost at each node

structure_repair_ratio = 1 - (str_cost_ratio * percent_failure +
nonstr_cost_ratio); %find repair cost ratio for structure nodes
nodeproperty(find(any(result~=0,2)), 8) = 1;                              %set
repair cost ratio for unoccupied nodes (affected) = 1
affected_structure_idx =
find(structure_prop(:,5)~=0);                              %find affected structures
(unaffected structure will not have any damage)
nodeproperty(affected_structure_idx, 8) =
structure_repair_ratio(affected_structure_idx);              %set repair cost ratio
for structure nodes
nodeproperty(:,9) = nodeproperty(:,6) .*
nodeproperty(:,8);                 %calculate the repair cost at each node
```

Functions not used for this case study

imageprocess

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to find the node type based on sub-images
% Input
%       a - real map length (matrix columns)
%       b - real map width (matrix rows)
%       m - mesh length
%       n - mesh width
%       nnodes - total number of nodes
%       nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%       grid_distance_a - mesh grid distance in x direction
%       grid_distance_b - mesh grid distance in y direction
%
% Output
%       nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%       water - water nodes [node number, x coordinate, y coordinate]
%       barrier - barrier nodes [node number, x coordinate, y coordinate]
%
%       figure(13) - map with identified fuels
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [nodeproperty,water,bridge] =
imageprocess(a,b,m,n,nnodes,nodeproperty,grid_distance_a,grid_distance_b)
imagenum = ['1','2','3','4','5','6','7','8','9'];                             %So far
the large image is cut into 9 pieces and used to find the fuel type on each piece.
m1 = floor(m/3);                                                              %we have
3 images in x-axis and ignore the nodes on the most right side
n1 = floor(n/3);                                                              %we have
3 images in y-axis and ignore the nodes on the most top side
nnodes1 = m1*n1;                                                              %find
total number of nodes in a sub-image

%create empty matrix to save coordinates for different fuel types
water = [];
bridge = [];

%This for loop will loop through all the sub-images and change the fuel
%type then save the node number and coordinates into the empty matrix
%[nodeproperty,water,barrier] = image_preprocess1(m, n, nnodes, nodeproperty);
for i = 1:9
    [nodeproperty,temp_water,temp_bridge] = image_preprocess2(m1, n1, nnodes1,
nodeproperty, grid_distance_a, grid_distance_b, imagenum(i));
    water = [water; temp_water];
    bridge = [bridge; temp_bridge];
end

%Figure 13 plot the assemble image after assigning fuel types
figure(13)
I = imread('satellite2.png');
%I = imread('satellite2.png');
h = image(linspace(0,a,a),linspace(0,b,b),flipud(I));
set(gca,'YDir','normal');
```

```
hold on
if isempty(water) ~= 1
    plot(water(:,2),water(:,3),'r.');
end
if isempty(bridge) ~= 1
    plot(bridge(:,2),bridge(:,3),'k.');
end
hold off
```

image_preprocess2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is going to find the node type on each sub-image
% Input
%       m - mesh length
%       n - mesh width
%       nnodes - total number of nodes
%       nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%       grid_distance_a - mesh grid distance in x direction
%       grid_distance_b - mesh grid distance in y direction
%       imagenum - image number defined by user
%
% Output
%       nodeproperty - [node_num, x_co, y_co, ROS, node_type, tributary_cost]
%       water - water nodes [node number, x coordinate, y coordinate]
%       barrier - barrier nodes [node number, x coordinate, y coordinate]
%
% <04,17,2024> {Xitong Zhou}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [nodeproperty,water,barrier] = image_preprocess2(m, n, nnodes, nodeproperty,
grid_distance_a, grid_distance_b, imagenum)

[I,cmap] = imread(strcat(imagenum,'.jpg'));                                 %read the
image (image scans from top left corner
figure(3+str2double(imagenum))
h = image(linspace(0,m,m),linspace(0,n,n),flipud(I));                       %change
the image resolution into the mesh size for a sub-image and flip the y axis
title(imagenum)
hold on
set(gca,'YDir','normal');                                                   %set the
ydirection to normal ((0,0) at bottom left corner)
croppedImage = imresize(I, [n, m]);                                         %resize
the resolution of original image

%use matlab built-in function to find the coordinates for different fuel
%types (I used color thresholder)
[BW,masked] = findwater(croppedImage);
[BW2,masked2] = findbarrier(croppedImage);

[row,col] = find(BW);
[row2,col2] = find(BW2);
```

```
%change the coordinates (up-side down) into what I can use
a = size(masked,1);
row = a-row;                                                         %convert the
y-axis upside down
row2 = a-row2;
col = col;
col2 = col2;

%plot the assigned fuel type on each sub-image
plot(col,row,'r.')
plot(col2,row2,'k.')
hold off

%The following if statements convert sub-image coordinates (local
%coordinates) into simulation coordinates (global coordinates) need to use
%new if statement if more fuel types needs to be identified

%water
if contains('456',imagenum)
    row = (row + n) * grid_distance_b;
elseif contains('789',imagenum)
    row = (row + 2*n) * grid_distance_b;
elseif contains('123',imagenum)
    row = row * grid_distance_b;
end
if contains('258',imagenum)
    col = (col + m) * grid_distance_a;
elseif contains('369',imagenum)
    col = (col + 2*m) * grid_distance_a;
elseif contains('147',imagenum)
    col = col * grid_distance_a;
end

%bridge
if contains('456',imagenum)
    row2 = (row2 + n) * grid_distance_b;
elseif contains('789',imagenum)
    row2 = (row2 + 2*n) * grid_distance_b;
elseif contains('123',imagenum)
    row2 = row2 * grid_distance_b;
end
if contains('258',imagenum)
    col2 = (col2 + m) * grid_distance_a;
elseif contains('369',imagenum)
    col2 = (col2 + 2*m) * grid_distance_a;
elseif contains('147',imagenum)
    col2 = col2 * grid_distance_a;
end

%find the node number for each fuel type
%findnodenum will return 3 columns, only column 1 is meaningful here
water = findnodenum(col,row,nodeproperty);
barrier = findnodenum(col2,row2,nodeproperty);
```

```
%assign the fuel properties (1=normal
fuel,2=water,3=barrier,4=concrete,5=steel,6=wood)
if isempty(water) ~= 1
    water_reduced = all(water(:, 1) == 0, 2);
    water = water(~water_reduced, :);
    nodeproperty(water(:,1),5) = 2;
end
if isempty(barrier) ~= 1
    barrier_reduced = all(barrier(:, 1) == 0, 2);
    barrier = barrier(~barrier_reduced, :);
    nodeproperty(barrier(:,1),5) = 3;
end

%Change spread rate to 0 for certain types of fuel type
for i = 1: nnodes
    nodeproperty(find(nodeproperty(:,8)==3),4) = 0;
    %test for barrier, barrier is using 2 which should be structure
    nodeproperty(find(nodeproperty(:,8)==2),4) = 0;
end
```

Reference

Albini, F., (1976). Computer-based models of wildland fire behavior: A user's manual. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and range Experiment Station. 68 p.

Albini, F., (1976). Estimating wildfire behavior and effects. Rep. No. GTR INT-30. USDA, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

Albini, F., (1979). Spot fire distance from burning trees-a predictive model. General Technical Report INT-GTR-56. Ogden, UT: USDA Forest Service, Intermountain Forest and Range Experiment Station. 73 p.

Alexander, M., and Cruz, M., (2013). Limitations on the accuracy of model predictions of wildland fire behaviour: A state-of-the-knowledge overview. The forestry chronicle vol.89

Alfonseca, K., (2023). Timeline: How the deadly wildfires took over Maui day by day. abcNEWS. Retrieved on 6/1/2024, from: https://abcnews.go.com/US/timeline-deadly-wildfires-maui-day-day/story?id=102253075

Alipour, M., Puma, I., Picotte, J., Shamsaei, K., Rowell, E., Watts, A., Kosovic, B., Ebrahimian, H., Taciroglu, E., (2023). A Multimodal Data Fusion and Deep Learning Framework for Large-Scale Wildfire Surface Fuel Mapping. Fire 2023.6.36. https://doi.org/10.3390/fire6020036

Alloza, J., Baeza, M., Riva, J., Duguy, B., Echeverria, M., Ibarra, P., Llovet, J., Perez-Cabello, F., Rovira, P., Vallejo, V., (2006). A model to evaluate the ecological vulnerability to forest fires in Mediterranean ecosystems. Forest Ecology and Management 234, S203

Anderson, D., Catchpole, E., Mestre, N., Parkes, T., (1982). Modeling the Spread of Grass Fires. J. Austral. Math.Soc. (Series B) 23(1982), 451-466

Anderson, H., (1983). Predicting Wind-Driven Wild Land Fire Size and Shape. Rep. No. GTR INT-305. USDA, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

Andrews, P. (2018). The Rothermel surface fire spread model and associated developments: A comprehensive explanation. Gen. Tech. Rep. RMRS-GTR-371. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 121 p.

Andrews, P., (1986). BEHAVE: Fire behavior prediction and fuel modeling system. Burn subsystem. Part 1. Rep. No. GTR INT-194. USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

Andrews, P., (1986). BEHAVE: Fire behavior prediction and fuel modeling system. Burn subsystem. Part 1. Rep. No. GTR INT-194. USDA Forest Service, Inter Mountain Forest and Range Experiment Station, Ogden, UT

Andrews, P., Cruz, M., Rothermel, R., (2013). Examination of the wind speed limit function in the Rothermel surface fire spread model. International Journal of Wildland Fire. 22(7): 959-969

Arroyo, L., Healey, S., Cohen, W., Cocero, D., (2006). Using object-oriented classification and high-resolution imagery to map fuel types in a Mediterranean region. Journal of Geophysical Research 111, G04S04 doi:10.1029/ 2005JG000120.

BBC News, (2017). California wildfires: Flames seen from space. Retrieved on 6/1/2024, from https://www.bbc.com/news/world-us-canada-42278027

Beitel, J. and Iwankiw, N., (2002). Analysis of Needs and Existing Capabilities for Full-Scale Fire Resistance Testing. National Institute of Standards and Technology (NIST). NIST GCR 02-843

Berlad, A., (1970). Fire spread in solid fuel arrays. Combustion and Flame, State University of New York, Stony Brook, New York

Burgan, R. and Rothermel, R., (1984). BEHAVE: Fire Behavior Prediction and Fuel Modeling System—FUEL Subsystem. Rep. No. GTR INT-167. Intermountain Forest and Range Experiment Station, Ogden, UT.

CAL FIRE. (n.d.). Remembering the Camp Fire. State of California. Retrieved on 6/1/2024, from *https://www.fire.ca.gov/our-impact/remembering-the-camp-fire*

Carmel, Y., Paz, S., Jahashan, F., Schoshany, M., (2009). Assessing Fire Risk Using Monte Carlo Simulations of Fire Spread. Forestt Ecology and Management 257(2009) 370-377. DOI: 10.1016/j.foreco.2008.09.039

Choe, L., Zhang, C., Luecke, W., Gross, J., Varma, A., (2017). Influence of material models on predicting the fire behavior of steel columns. Fire Technol. 53 (1) (2017) 375-400

Deeming, J., Burgan, R., Cohen, J., (1977). The National Fire Danger Rating System. Rep. No. GTR INT-39. USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT.

Deeming, J., Lancaster, J., Fosberg, M., et al., (1972). National Fire Danger Rating System. Res. Pap. RM-84. Fort Collins, CO: USDA, Forest Service, Rocky Mountain Forest and Range Experiment Station. 165p

Fernandez, C., Vega, J., Fonturbel, T., Perez-Gorostiaga, P., Jimenez, E., Madrigal, J., (2007). Effects of wildfire, salvage logging and slash treatments on soil degradation. Land Degradation & Development 18, 591-607

Fernandez-Pello, A., (2017). Wildland Fire Spot Ignition by Sparks and Firebrand. Fire Safety Journal 91(2017)2-10

Finney, M., (1998). FARSITE: fire area simulator-model development and valuation. Rep. No. Paper RMRS-RP-4. USDA Forest Service, Ogden, UT.

Finney, M., (1999). Mechanistic Modeling of Landscape Fire Patterns. Spatial Modeling of Forest LandSCAPE Change: Approaches and Applications, Chapter 8, Missoula Fire Lab

Finney, M., (2004). FARSITE: Fire Area Simulatior – Model Development and Evaluation. USDA, Research Paper RMRS-RP-4 Revised

Frandsen, W. (1971). Fire spread through porous fuels from the conservation of energy. Combustion and Flame. 16(1): 9-16

Franssen, J., and Real, P., (2015) Fire Design of Steel Structures, 2nd edition, Eurocode 3 : Design of steel structures Part1-2 – General rules – Structural fire design. ISBN(ECCS):978-3-433-03143-8

Gollner, M., Hakes, R., Caton S., and Kohler, K. (2015). Pathways for building fire spread at the wildland Urban Interface. Report to Fire Protection Research Foundation. College Park, Maryland, USA.

Himoto, K. and Tanaka, T., (2008). Development and Validation of a Physics-based Urban Fire Spread Model. Fire Safety Journal 43 (2008) 477-494

Hough, W. and Albini, F., (1978). Predicting fire behavior in palmetto-gallberry fuel complexes. Res. Pap. SE-174. Asheville, NC: USDA, Forest Service, Southeastern Forest Experiment Station. 44p

Huygens, C., (1962). Treatise on light (S.P. Thompson, translator). Dover Publ., New York. (Original work published 1912.) 128p

IBHS, 2014. IBHS website. http://www.disastersafety.org.

Janssens, R., White, R., (1994). Temperature profile in wood members exposed to fire, Fire Mater, 18(4) (1994) 263-265

Juliano, T., Bardales, F., Lareau, N., Shamsaei, K., Kosovic, B., Khorasani, N., James, E., Ebrahimian, H., (2024). Brief communication: The Lahaina Fire disaster – how models can be used to understand and predict wildfires. Nat. Hazards Earth Syst. Sci., 24, 47-52, 2024

Khorasani, N., Ebrahimian, H., Buja, L., Cutter, S., Kosovic, B., Lareau, N., Meacham, B., Rowell, E., Taciroglu, E., Thompson, M., Watts, A., (2022). Conceptualizing a probabilistic risk and loss assessment framework for wildfires. Natural Hazards (2022) 114:1153-1169

Khorasani, N., Gardoni, P., Garlock, M., (2015). Probabilistic Fire Analysis: Material Models and Evaluation of Steel Structural Members. ASCE, DOI: 10.1061/(ASCE)ST.1943-541X.0001285

Khoury, G., (2000). Effect of fire on concrete and concrete structures. Prog. Struct. Engng Mater. 2000; 2: 429-447. Imperial College, London, UK.

Kochanski, A., Jenkins, M., Mandel, J., Beezley, J., Clements, C., Krueger, S. (2013). Evaluation of WRF-SFIRE performance with field observations from the FireFlux experiment, Model Dev., 6, 1109–1126

Lennon, T., and Moore, D., (2003). The natural fire safety concrpt – full-scale tests at Cardington. Fire Safety Journal 38 (2003) 623 – 643

Mahmoud, H. and Chulahwat, A., (2016). A probabilistic cellular automata framework for assessing the impact of WUI fires on communities. Urban Transitions Conference, Shanghai.

Mahmoud, H. and Chulahwat, A., (2018). Unraveling the complexity of wildland urban interface fires. Scientific Reports. DOI: 10.1038/s41598-018-27215-5

Mahmoud, H., Ellingwood, B., and Memari, M., (2015). Challenges and alternative approaches for simulating the response of steel structures exposed to fire. Second International Conference on Performance-based and Life-cycle Structural Engineering, Brisbane, Australia.

Mak, J., Brewbaker, P., Haas, F., (2023). Economic perspective of Maui's Devastating wildfires. UHERO, Retrieved on 6/1/2024, from https://uhero.hawaii.edu/economic-perspective-of-mauis-devastating-wildfires/#:~:text=Moody's%20RMS%20estimates%20that%20insured,%242.5%20billion%20to%20%244%20billion.

Manzello, S., Maranghides, A., Mell, W., (2007). Firebrand generation from burning vegetation. Int. J. Wildl. Fire 16, 458. doi:10.1071/WF06079

Manzello, S., Maranghides, A., Shields, J., Mell, W., Hayashi, Y., Nii, D., (2009). Mass and size distribution of firebrands generated from burning Korean pine (Pinus koraiensis) trees. Fire Mater. 33, 21–31. doi:10.1002/fam.977

Manzello, S., Suzuki, S., Gollner, M., Fernandez-Pello, A., (2020). Role of Firebrand Combustion in Large Outdoor Fire Spread. Progress in Energy and Combustion Science 76 (2020 100801)

Maselli, F., Rodolfi, A., Bottai, L., Romanelli, S., Conese, C., (2000). Classification of Mediterranean vegetation by TM and ancillary data for the evaluation of fire risk. International Journal of Remote Sensing 17, 3303-3313

Merrill, D., Alexander, M., (1987). Glossary of Forest Fire Management Terms, fourth edn. National Research Council of Canada, Canadian Committee on Forest Fire Management, Ottawa, Ontario

Nelson, R., (2000) Prediction of diurnal change in 10-h fuel stick moisture content. Canadian Journal of Forest Research **30**, 1071–1087.

Ni, S., and Gernay, T., (2021). A framework for Probabilistic Fire Loss Estimation in Concrete Building Structures. Structural Safety 88(2021)102029

Ostman, B., (1985). Wood tensile strength at temperatures and moisture contents simulating fire conditions. Wood Sci. Technol. 19 (2) (1985) 103-116

Outinen, J. and Makelainen, P., (2004). Mechanical properties of structural steel at elevated temperatures and after cooling down. Fire Mater. 28 (2-4) (2004) 237-251

Pacific Disaster Center and the Federal Emergency Management Agency releases Fire Damage. County of Maui, Hawaii, Posed on: August 12, 2023, Retrieved on6/1/2024, from https://www.mauicounty.gov/CivicAlerts.aspx?AID=12683

Public Safety Canada. (n.d.). Event Details: Wildfire - Fort McMurray AB, April 30 to June 1, 2016. Canadian Disaster Database. Retrieved on 6/1/2024, from *https://cdd.publicsafety.gc.ca/dtprnt-eng.aspx?cultureCode=en-Ca&eventTypes=%27WF%27&normalizedCostYear=1&dynamic=false&eventId=1135&prnt=both*

Qureshi, R., Ni, S., Khorasani, N., Coile, R., Hopkin, D., Gernay, T., (2020). Probabilistic models for temperature-dependent strength of steel and concrete. ASCE. DOI: 10.1061/(ASCE)ST.1943-541X.0002621

Rothermel, R., (1972). A mathematical model for predicting fire spread in wildland fuels. Rep. No. RP INT-115. SDA Forest Service, Ogden, UT.

Rothermel, R., (1983). How to predict the spread and intensity of forest and range fires. Gen. Tech. Rep. INT-143. Ogden, UT: USDA, Forest Service, Intermountain Forest and Range Experiment Station. 161p

Schaffer, E., (1984). Structural Fire Design: Wood. Forest Products Laboratory, Madison, Wis

Shaik, R., Alipour, M., Rowell, E., Balaji, B., Watts, A., Taciroglu, E., (2024). FUELVISION: A Multimodal Data Fusion and Multimodel Ensemble Algorithm for Wildfire Fuels Mapping. arXiv: 2403.15462v1

Shaik, R., Alipour, M., Taciroglu, E., (2024). A bibliometric analysis of artificial intelligence-based solutions to challenges in wildfire fuel mapping. IGARSS 2023 – 2023 IEEE International Geoscience and Remote Sensing Symposium

Shamsaei, K., Juliano, T., Roberts, M., Ebrahimian, H., Kosovic, Branko., Lareau, N., Taciroglu, E., (2023). Coupled fire-atmosphere simulation of the 2018 Camp Fire using WRF-Fire. International Journal of Wildland Fire 32(2) 195-221

Solano, J., Montano, T., Maldonado-Correa, J., Ordonez, A., Pesantez, M., (2021). Correlation Between the Wind Speed and the Elevation to Evaluate the Wind Potential in the Southern Region of Ecuador. 6th International Conference on Advances on Clean Energy Research , ICACER 2021 April 15-17, 2021, Barcelona, Spain

Sousa, F., Reis, R., and Pereira, J., (2012). Simulation of surface fire fronts using fireLib and GPUs. Environmental Modelling & Software. DOI: 10.1016/j.envsoft.2012.06.006

Structural Fire Protection – ASCE Manuals and Reports on Engineering. Practice No.78, 1992

Sullivan, A, (2009a). Wildland surface fire spread modeling, 1990-2007. Empirical and quasi-physical models. International Journal of Wildland Fire. 18(4): 349-368

Sullivan, A, (2009b). Wildland surface fire spread modeling, 1990-2007. 2: Empirical and quasi-empirical models. International Journal of Wildland Fire. 18(4): 369-386

Sullivan, A, (2009c). Wildland surface fire spread modeling, 1990-2007. 3: Simulation and mathematical analogue models. International Journal of Wildland Fire. 18(4): 387-403

Suzuki, S., Brown, A., Manzello, S., Suzuki, J., (2014). Firebrands Generated from a Full-scale Structure Burning under Well-controlled Laboratory Conditions. Fire Safety Journal 63(2014)43-51

Tang, W., and Eickner, H., (1968). Effect of inorganic salts on pyrolysis of wood, cellulose, and lignin determined by differential thermal analysis. Forest Projects Laboratory, Forestry Service, USDA, Madison

Tymstra, C., Bryce, R., Wotton, B., Taylor, S., Armitage, O., (2010). Development and Structure of Prometheus: The Canadian Wildland Fire Growth Simulation Model. Canadian Forest Service Northern Forestry Center

Wagner, V., (1977). Conditions for the start and spread of crownfire. Can. J. For. Res. 7:23-34.

Wang, W., Zhang, Y., Xu, L., Li, X., (2020). Mechanical properties of high-strength Q960 steel at elevated temperature. Fire Safety Journal 114 (2020) 103010

Waterman, T., (1969). Experimental study of firebrand generation, IIT Research Institute, Project J6130, Chicago, IL, 1969.Weir, J.R., 2004. Probability of Spot Fires During Prescribed Burns. Fire Manag. Today 64, 24–26.

White, R. and Nordheim, E., (1992). Charring rate of wood for ASTM E 119 exposure. Fire Technol. 28(1) (1992) 5-30

White, R., Dietenberger, M., 2010. Fire Safety of Wood Construction. In: Centennial (Ed.), Wood Handbook Wood as an Engineering Material. Madison, WI: USDA, FS, Forest Products Laboratory.