

2021-05

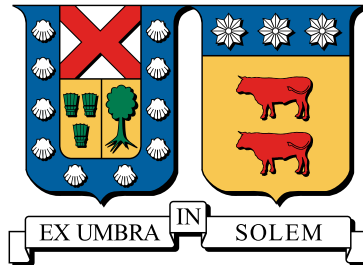
Open-source Framework for Chilean Wildfire Spreading and Effects Analysis

San Martín Reyes, Daniel Eduardo

<https://hdl.handle.net/11673/54317>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



Open-source Framework for Chilean Wildfire Spreading and Effects Analysis

Daniel Eduardo San Martín Reyes

Thesis presented in partial fulfillment of the requirements for the degree of
Magíster en Ciencias de la Ingeniería Informática

Advisor: Claudio Torres, Ph.D.
Internal Co-referent: Andrés Fuentes, Ph.D.
Internal Co-referent: Christopher Cooper, Ph.D.
External Co-referent: Orlando Ayala, Ph.D.
Committee Chair: Marcelo Mendoza, Ph.D.

May 13, 2021

THESIS TITLE:

Open-source Framework for Chilean Wildfire Spreading and Effects Analysis

AUTHOR:

Daniel Eduardo San Martín Reyes

Thesis presented in partial fulfillment of the requirements for the degree of Magister en Ciencias de la Ingeniería Informática of the Universidad Técnica Federico Santa María.

Claudio Torres

Advisor

Andrés Fuentes

Internal Co-referent

Christopher Cooper

Internal Co-referent

Orlando Ayala

External Co-referent

Marcelo Mendoza

Committee Chair

Valparaíso, Chile.
May 13, 2021.

For someone somewhere...

Acknowledgements

This work was funded by ANID PIA/APOYO AFB180002, Centro Científico Tecnológico de Valparaíso - CCTVal, and with the support of the Dirección de Postgrado y Programas Universidad Técnica Federico Santa María, Valparaíso, Chile.

Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02).

Resumen

Los incendios forestales son un problema de interés mundial que frecuentemente afectan a nuestro país, quemando miles de hectáreas de bosques, destruyendo la flora y fauna, afectando el aire, los ciclos del agua y el ecosistema. Estos daños han generado pérdidas económicas, ambientales y sociales significativas. Según estadísticas de la *CONAF* el 99,7% de los incendios en Chile son provocados por negligencia humana y desde 1963 a 2020 se han producido más de 250 mil incendios forestales con más de 3 millones de hectáreas quemadas.

Debido a la problemática que genera este tipo de fenómenos es que se han desarrollado diversas formas de abordar el problema, las cuales pueden ser clasificadas en: evaluación de riesgo, propagación y efecto. Nuestro interés se centra en los modelos de propagación, que se pueden sub-clasificar en físicos, empíricos e híbridos. Como ejemplo, dos de los enfoques altamente explorados durante las últimas décadas se encuentran los modelos discretos, como autómatas celulares y los modelos continuos basados en ecuaciones diferenciales. El desarrollo de estos modelos ha permitido la creación de una amplia variedad de software, pero solo algunos de estos son de código abierto.

El presente proyecto de tesis propone el desarrollo de un framework de código abierto, basado en ecuaciones diferenciales, para apoyar el estudio de la propagación de incendios forestales incorporando características geográficas y meteorológicas de Chile. Dada la estructura del algoritmo, este será implementado en *GPU* para reducir los tiempos de simulación numérica en la generación de mapas de riesgo.

Además, este trabajo presenta un método numérico eficiente para el cómputo del modelo de propagación de incendios y la implementación del algoritmo asociado. El algoritmo obtiene resultados satisfactorios respecto a propiedades de estabilidad, convergencia y complejidad computacional. Por otra parte, la implementación en *GPU* del algoritmo propuesto, supera a 3 implementaciones en *CPU*. Estos resultados positivos permiten concluir que el framework de código abierto es un instrumento válido para el estudio de los incendios forestales.

Palabras clave: Modelamiento de incendios forestales, Computación Científica, Métodos Numéricos, *GPU*.

Abstract

Wildfires are a problem of global interest that frequently affect our country, burning thousands of hectares of forests, destroying flora and fauna, affecting the air, water cycles and the ecosystem. These damages have generated significant economic, environmental and social losses. According to *CONAF* statistics, in Chile 99.7% of fires are caused by human negligence and from 1963 to 2020 more than 250 thousand wildfires have been occurred with more than 3 million hectares burned.

Due to the problems generated by this type of phenomena, several approaches have been developed to address the problem, which can be classified into: risk assessment, propagation and effect. Our focus is on propagation models, which can be sub classified into physical, empirical and hybrid. As an example, two of the highly explored approaches during the last decades are discrete models, such as cellular automata and continuous models based on differential equations. The development of these models has allowed the creation of a wide variety of software, but only few of these are open-source.

This thesis project proposes the development of an open-source framework, based on differential equations, to support the study of the wildfire spreading by including Chilean geographical and meteorological characteristics. By the algorithm structure, it will be implemented in *GPU* to reduce the numerical simulation times in the generation of risk maps.

Additionally, this work presents an efficient numerical method for the computation of the fire propagation model and the implementation of the associated algorithm. The algorithm presents positive properties of stability, convergence, and complexity. Also, the *GPU* implementation of the proposed algorithm outperforms 3 different *CPU* implementations. These satisfactory results lead to the conclusion that the open-source framework is a valid instrument for the study of wildfires.

Keywords: Wildfire Modeling, Scientific Computing, Numerical Methods, *GPU*.

Contents

Resumen	III
Abstract	IV
Contents	VII
List of Figures	IX
List of Tables	X
List of Algorithms	XI
1 Introduction	1
1.1 Objectives	3
1.2 Structure	3
2 State of the Art	4
2.1 Wildfire Models	4
2.1.1 Partial Differential Equations	4
2.1.2 Cellular Automata	6
2.2 Wildfire Software	9
3 Wildfire Mathematical Model	13
3.1 Model Derivation	13
3.1.1 Fire Reaction rate	15
3.1.2 Balance equation of fuel supply	15
3.1.3 Conservation of energy	16
3.1.4 Final Model	20
3.2 Model analysis	20
3.2.1 Parameters	20
3.2.2 Nondimensionalization	22
4 Proposed Algorithm	26
4.1 Working Hypothesis	27
4.2 Simplified Two-dimensional Model	27

CONTENTS

5	Numerical Algorithm	30
5.1	Method of Lines	30
5.2	Spatial Approximation	30
5.2.1	Differentiation Matrices	31
5.3	Time Approximation	34
5.3.1	Euler Method	35
5.3.2	Fourth Order-Runge Kutta Method	36
5.4	Algorithm Description	36
5.5	Stability Analysis	38
5.6	Risk Maps Generation Algorithm	41
6	Numerical Implementation	42
6.1	<i>CPU</i> Version	42
6.2	<i>GPU</i> Version	44
6.2.1	<i>CUDA</i> description	44
6.2.2	<i>GPU</i> algorithm	46
7	Numerical Results	51
7.1	Algorithm Analysis	51
7.1.1	Stability	52
7.1.2	Convergence	53
7.1.3	Complexity	53
7.2	Numerical Simulation	55
7.2.1	Asensio & Ferragut output	55
7.2.2	Controlled fire	55
7.2.3	Wind effect	58
7.3	Risk Maps	58
7.4	<i>CPU-GPU</i> Performance	64
8	Open-source Framework	67
8.1	<i>Python</i>	67
8.1.1	Prerequisites	67
8.1.2	Installation	67
8.1.3	Usage	68
8.2	<i>CUDA</i>	69
8.2.1	Prerequisites	69
8.2.2	Installation	69
8.2.3	Usage	69
8.3	Data	70
8.3.1	How to use	70
8.3.2	Example files	70
9	Conclusions	72
9.1	Future Work	73

References	74
A Appendix	83
A.1 Mathematical Stuff	83
A.1.1 Reynolds theorem for an incompressible flow	83
A.1.2 Taylor expansion for radiation term	84
A.2 Kernel Memory experiment	84
A.2.1 Experiments	84
A.2.2 Times	85
A.2.3 Experimental bandwidths	85
A.2.4 Analysis	86
A.2.5 Summary	86
A.2.6 Comments	86
A.3 Figure Reuse Licenses	87

List of Figures

1.1	Historical statistics of wildfires in Chile from 1963 to 2020.	2
3.1	Wildfire components sketch. Images from Freepik.com	14
3.2	Fuel consumption.	16
3.3	Sketch of the components for thermal energy dynamics formalization.	17
3.4	Sketch of the thermal energy generated by a source.	19
4.1	Diagram of risk map generation.	27
4.2	Sketch of model components (San Martín & Torres, 2018d).	29
6.1	<i>Python</i> implementation of risk maps generation algorithm.	43
6.2	<i>C</i> implementation of risk maps generation algorithm.	44
6.3	<i>OpenMP</i> implementation of risk maps generation algorithm.	45
6.4	<i>CUDA</i> software mapping.	46
6.5	Programming tools provided by <i>CUDA</i> platform (Nobile, Cazzaniga, Besozzi, Pescini, & Mauri, 2014).	47
6.6	<i>CUDA</i> implementation of risk maps generation algorithm.	48
6.7	Description of the GPU implementation.	48
6.8	Single dimensional array of <i>CUDA</i> threads.	48
7.1	Eigenvalues of \mathbf{J}_Φ for first (left plot) and last (right plot) time step using the <i>Euler Method</i>	52
7.2	Eigenvalues of \mathbf{J}_Φ for first (left plot) and last (right plot) time step using the <i>RK4</i> method.	53
7.3	Analysis of the algorithm convergence.	54
7.4	Analysis of the algorithm complexity.	54
7.5	Numerical simulation to achieve Asensio and Ferragut (2002) output. Left column presents the nondimensional temperature and right column presents the fuel fraction. Times for each row are: $t = \{0, 10, 15, 20\}$	56

7.6	Numerical simulation output adopted from Asensio and Ferragut (2002, p. 155). Left column presents the temperature and right column presents the fuel fraction. Republished with permission of John Wiley & Sons - Books, from On a wildland fire model with radiation, Asensio, M. I.; Ferragut, L., 54, 1, 2002; permission conveyed through Copyright Clearance Center, Inc.	57
7.7	Experiment of controlled fire at $t = \{0, 10, 15, 20\}$. Left column presents the nondimensional temperature and right column presents the fuel fraction.	59
7.8	Numerical simulation output adopted from Mell, Charney, Jenkins, Cheney, and Gould (2013, p.11). Republished with permission of Springer, from Numerical Simulations of Grassland Fire Behavior from the LANL-FIRETEC and NIST-WFDS Models, Mell W., Charney J., Jenkins M.A., Cheney P., Gould J, 2013; permission conveyed through Copyright Clearance Center, Inc.	60
7.9	Wind effect experiment at $t = \{0, 10, 30, 50\}$. Left column presents the nondimensional temperature and right column presents the fuel fraction.	61
7.10	Examples of <i>risk maps</i> generated from multiple numerical simulations. Top plot shows the initial conditions for the experiment. Middle plot shows the time that 10% of fuel was burnt. The darkest red denotes a more dangerous fire ignition zone. Bottom plot shows the percentage of fuel burned at end of numerical simulation. The yellow zones indicates a more dangerous fire ignition zone.	63
7.11	Burnt rate map example. Blue values indicates a faster fuel consumption rate.	64
7.12	CPU-GPU performance comparison.	66

List of Tables

2.1	Numerical methods summary.	7
2.2	Components of presented model approaches summary.	10
2.3	Software summary.	12
3.1	Metric units in wildfire context.	21
3.2	Value range of the wildfire model's physical parameters (Asensio & Ferragut, 2002).	21
4.1	Physical parameters used by this thesis.	28
7.1	Average execution time [s] of the algorithm for a different number of threads and blocks.	64
7.2	Algorithm implementations average times in [s].	65
A.1	Tesla V100 memory description.	85
A.2	Times [ms] for each matrix size using different <i>kernels</i>	85
A.3	Experimental bandwidths [GB/s] computed for each matrix size. . . .	86
A.4	Estimated bandwidth [GB/s]	86
A.5	NVIDIA V100 summary.	86

List of Algorithms

1	Main algorithm.	36
2	Right hand side Φ approximation.	37
3	4th Order Runge-Kutta.	37
4	Φ approximation using C implementation.	43
5	$RK4$ <i>host</i> implementation.	49
6	$CUDA$ implementation of vector a plus scalar times vector c	49
7	$RK4$ scheme implementation in $CUDA$	49
8	$CUDA$ implementation of $\Phi(\mathbf{Y})$	50

Chapter 1

Introduction

“A wildfire is a fire that, whatever its origin and with the possibility of damaging people, property, or the environment, spreads uncontrolled in rural areas, through the woody, bushy or herbaceous living or dead vegetation. In other words, it is an unjustified and out-of-control fire in which the fuels are vegetation and which, in its propagation, can destroy everything in its path” (CONAF, 2021).

Wildfires are ubiquitous phenomena of global interest since they may occur in almost every zone of the world (Doerr & Santin, 2017), and have the potential to damage large extensions of land (Archibald, Lehmann, Gómez-Dans, & Bradstock, 2013). Fire can start for different causes, including natural causes such as lightning, spontaneous combustion, and volcanic eruption, but about 90% of worldwide fires are caused by humans, either deliberately or by negligence (Costafreda-Aumedes, Comas, & Vega-Garcia, 2017). The seasons of peak occurrence happen in summer and frequently in areas of Mediterranean climate, conditions that favor the uncontrolled spread of fire (Úbeda & Sarricolea, 2016). The main effects of wildfires include flora, fauna and soil damages, air pollution, human lives, and economic losses (Doerr & Santin, 2017). An important study zone is a wildland-urban interface (WUI), zones that combine both unoccupied and populated areas, which have an obvious latent danger in a wildfire event (Doerr & Santin, 2017).

In the last decades, Chile has been affected by a significant number of fire episodes. According to CONAF (CONAF, 2021), from 1963 to 2020, more than 250 thousand wildfires have occurred damaging more than 3 million hectares. Regarding the start of the fires, 99.7% are associated with human negligence. Figure 1.1 shows the historical number of wildfires and burnt hectares in Chile. According to Peña-Fernández and Valenzuela-Palma (2004), until 2004, the increase in these statistics is closely related to the planting of highly flammable species like Eucalyptus and Pines. A more extensive review of the wildfires in Chile can be found at Úbeda and Sarricolea (2016).

Due to the devastating effects, and to mitigate the consequences of wildfires, it is necessary to be able to study, analyze and model the fire behavior. Several researchers from different areas have studied wildfires for many years, making mathematical models to describe their dynamics (Weber, 1991a; Pastor, Zárata, Planas,

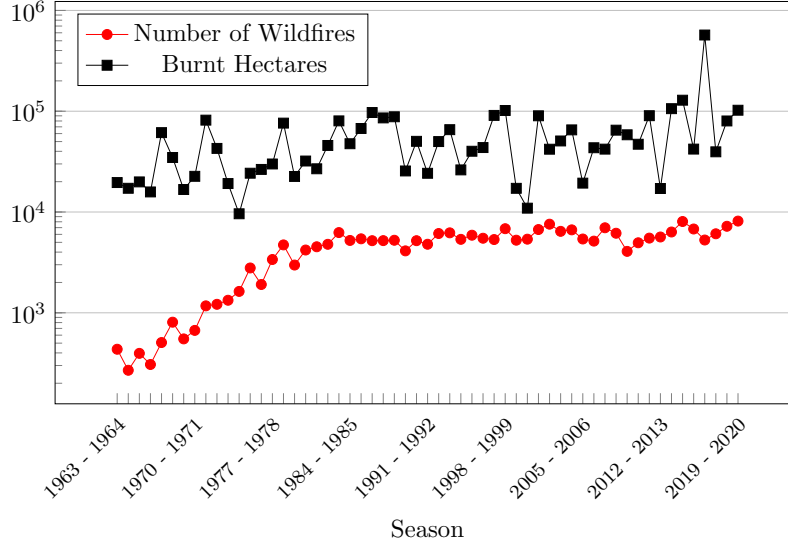


Figure 1.1: Historical statistics of wildfires in Chile from 1963 to 2020.

& Arnaldos, 2003).

According to Preisler and Ager (2014) fire models can be grouped into three categories: risk assessment, propagation, and effect models. The first models are associated with the quantification of possible fire episodes. The second category studies the propagation of fire, including physical up to regression approaches. Finally, an effect model is a high-level analysis, including fuel management planning, tree mortality up to analysis of ecosystems, among others. These categories are not mutually exclusive and often interact with each other. Focusing on the spreading models and according to the nature of equations, they can be sub-classified into three types: empirical, semi-empirical, and physical models (Pastor et al., 2003). Empirical ones use a statistical approach to describe the fire behavior. The works of McArthur (1966, 1967) are very cited since the author designed fire danger metrics using statistical correlations from experimental burns (Pastor et al., 2003). Semi-empirical models combine an empirical approach with some physical laws such as conservation of energy and mass. The most important work is referred to by Rothermel (1972), in which the author uses a theoretical framework and experimental data to propose a surface fire spread model. Finally, the physical models are based exclusively on physical laws. The work of Asensio and Ferragut (2002) is a notable reference where the authors derived a non-linear system of equations to simulate the fire spreading behavior.

As a result of these researches, computer software has been developed to support the phenomena under study, helping to decrease the damage caused by fire. Many of this software is not open-source, so it is not possible to study the implementation of these. Considering the development of this kind of computational tools, the main motivation of this work is the implementation of an *Open-source Framework* that explore the use of *Graphics Processing Units* to speed up the computation of fire

dynamics numerical simulations using *Partial Differential Equations*-based models.

1.1 Objectives

There are two main objectives associated with this thesis project. The first one is to design and implement a numerical algorithm derived from a mathematical model of wildfire spreading using physical laws and based on *Partial Differential Equations*. The second aim is to provide an *Open-source Framework* for support in the study of wildfire effects in Chile. Specific objectives defined to achieve the thesis purposes are:

1. Develop a numerical algorithm for *CPU*.
2. Develop a numerical algorithm for *GPU*.
3. Build risk maps with the wildfire vulnerability zones employing numerical simulations on several fire scenarios.
4. Model framework components using Chilean geographical and weather data.

1.2 Structure

The present work is structured as follows. Chapter 2 provides a literature review of the most important approaches in fire modeling. This chapter contains a selection of the relevant research in continuous and discrete world to address wildfire simulations, with a comparative review between *Partial Differential Equations* and *Cellular Automata* models. Additionally, it includes the leading software used in forest fire study. Chapter 3 presents the general ideas behind physical wildfire modeling, the mathematical formalization, and the derivation of the simplified two-dimensional model used in this work. The thesis proposal is presented in Chapter 4 with its implications and the products associated with the development of the working hypothesis. Chapter 5 presents the numerical algorithm used to approximate the wildfire mathematical model. The chapter includes the theoretical concepts to build the proposed algorithm. Also, it includes the description of numerical methods used in the development of the proposed algorithm, in particular, the use of *Method of Lines* which is widely used in the numerical solution of *Partial Differential Equations*. Computational implementation of the numerical algorithm is presented in Chapter 6. This work considers the implementation of the algorithm in both *CPU* and *GPU*. The aim is to demonstrate experimentally that the proposed algorithm is suitable for a *GPU* implementation and has a better performance than a *CPU* version. Chapter 7 presents results of the algorithm and the applications that can be performed using it. The description of how to use the open-source framework developed is presented in Chapter 8. Finally, Chapter 9 compiles the conclusions of the thesis and defines possible lines of future work.

Chapter 2

State of the Art

2.1 Wildfire Models

During the last decade, two of the most used approaches in Wildfire Modeling are *Partial Differential Equations (PDE)* and *Cellular Automata (CA)* based models. The first class of models is based on physical laws, describing physical and chemical processes using differential equations (Pastor et al., 2003). In the second kind of model, fire dynamic is studied on a landscape grid following transition rules. These rules determine the state of each cell and they are based on theoretical and semi-empirical mathematical fire behavior models (Pastor et al., 2003).

This work will be focused on the first approach, but to provide a comparison with the second one, the succeeding section presents a selection of relevant literature using both techniques.

2.1.1 Partial Differential Equations

PDE-based models attempt to predict the fire spread employing fundamental physical laws and mathematical equations (Weber, 1991a). These equations produce a system of *PDE* which in many cases are impossible to solve analytically, it is necessary to use numerical approximations.

Weber (1991b) presents a theoretical model based on physical processes involved in wildfires. Processes included chemical kinetics and heat transfer mechanisms, becoming into a one-dimensional *PDE* with advection, radiation, and reaction. The author concludes that a 2D version of the model has solutions with an elliptical shape and presents a set of parametric equations to describe the geometry of the fire front. Richards (1995) presents a mathematical model for two-dimensional fire spreading based on the fire perimeter. The model is a system of *PDE* that describes the fire growth in terms of the rate of spread. The mathematical formulation includes factors such as homogeneous fuel and wind speed conditions. Grishin (1996) presented a 3D surface and crown wildfire spread model, including heat and mass transfer, combustible pyrolysis, and the effect over a near-ground atmospheric layer. The formulation takes into account turbulence models and radiation models. In Clark, Jenkins,

Coen, and Packham (1996), the authors describe a 3D coupled atmosphere-fire model for the wildfire simulation model, based on equations of motion and thermodynamics. The model includes atmosphere and fuel dynamics to avoid non-physical solutions. This work also includes a fire effects study of the main components of the proposed model. Montenegro, Plaza, Ferragut, and Asensio (1997) proposed a 2D continuous mathematical model based on the fuel type, moisture content, wind velocity, topography, among others. The model is solved numerically using the *Euler Method* for time, and *Finite Differences Method (FDM)*, and adaptive *Finite Element Method (FEM)* for space. In Santoni and Balbi (1998) and Balbi, Santoni, and Dupuy (2000) the authors developed a 2D reaction-diffusion equation for wildfire spread over a fuel bed regardless of wind and terrain conditions. The model allows estimating the acceleration of spread for a point-ignition fire at the initial stage and steady-state phase. The numerical scheme for the simulation used *FEM* and central difference for space and forward difference for time. Asensio and Ferragut (2002) presented a simplified 2D wildfire model based on conservation laws and using radiation as the dominant heat transfer mechanism. The convection is modeled by the effect of wind and terrain slope, moreover, it includes information of moisture content. They developed a numerical method using the *Euler Method*, *FDM*, and *FEM*. Ferragut, Asensio, and Monedero (2007b, 2007a) present a 2D model including moisture content and radiation. Combustible is modeled using a simplified energy conservation equation. Vegetation behavior is incorporated using a multi-valued function representing Enthalpy. The model also includes wind and slope effects. The numerical solution is performed using semi-implicit *Euler* and *FEM*. Mandel et al. (2008) present a 2D coupled system of *PDE* that includes a *Kalman filter* to assimilate temperatures into wildfire numerical simulation. The model is similar to the proposed in Asensio and Ferragut (2002). Numerical methods used were the *Euler method* for time and central difference *FDM* for space. In Eberle (2014) and Eberle, Freeden, and Matthes (2015) the authors derived a nonlinear system of equations, similar to Asensio and Ferragut (2002), considering heat and mass transfer mechanism. They proposed a numerical method to solve the nonlinear system of equations, using a space discretization based on a collocation method with Radial Basis Function (*RBF*) and a Crank-Nicolson discretization for time. Baranovskiy and Nee (2015) present a 3D simplified mathematical model of heat transfer to simulate surface forest fires. The model considers temperature and fuel dynamics, paying attention to a two-layer soil structure. The model is solved numerically using *FDM*. Ferragut, Asensio, Cascón, and Prieto (2015) used a simplified two-dimensional fire model with three-dimensional effects. The model coupled topography, fuel, and meteorological information using a geographic information system. The model also includes data assimilation for the temperature of the solid fuel and the mass fraction of fuel at a certain position in the domain. According to the authors, this approach improves the predictions of fire spread. Numerical results were performed using *FEM* and *Euler method*. Guelpa, Sciacovelli, Verda, and Ascoli (2016) present a technique to reduce computational time for physical modeling of wildfire spreading. The technique used

is called proper orthogonal decomposition. The mathematical model used is a system of *PDE* similar to the one developed in Weber (1991b); Santoni and Balbi (1998); Balbi et al. (2000), and the proposed method is applied to 1D equations. Lopes, Ribeiro, Viegas, and Raposo (2017) present a method to model the effect of wind in forest fire spreading simulations. The model couples Rothermel’s surface fire spread model (Rothermel, 1972) and Navier-Stokes equations with the $k - \varepsilon$ turbulence model. Fečkan and Pačuta (2018) established the existence of a solution for a 2D fire spread model that studies the geometry of fire. They do not present an analytical solution and just indicate that was solved numerically. The model presented is similar to Richards (1995). In Ambroz, Balažovjeh, Medl’a, and Mikula (2018) authors propose a new approach to modeling the spread of forest fires. They used the evolution of a 3D surface curve to represent fire perimeter on the topography. The model also includes climatic variables, terrain, and fuel dynamics to calculate the perimeter of the fire. The numerical scheme is based on *FEM*. Perminov and Soprunenko (2016); Perminov (2017, 2018) present a coupled model for crown fire initiation and spreading using a 3D physical approach using conservation of mass, momentum, fuel, and energy equations. The model includes the atmosphere interaction. Numerical simulations are performed using the *Finite Volume Method (FVM)*. Marzaeva and Galtseva (2018) present a numerical method to solve the initiation and spread of surface and crown fires using the interaction between fire and atmospheric models similar to Perminov (2018). The numerical solution was performed using the *Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)* algorithm (S. V. Patankar & Spalding, 1972; S. Patankar, 2018). Toivanen et al. (2019) present an atmospheric model coupled with a fire model to simulate a UK fire. It is a surface and spotting wildfire spreading. Fire spread is calculated by combining the level set method and McArthur’s empirical formulas (McArthur, 1966, 1967). The level set *PDE* is solved by *Runge-Kutta* of second-order and *Godunov* upwind methods. The model also includes a fuel-burning model similar to the one developed by Albini (Albini, Brown, Reinhardt, & Ottmar, 1995). Grasso and Innocente (2020) present a physical-based model as an extension of the work of Asensio and Ferragut (2002), including further detail on the combustion process and chemical species. The paper aims is to present a real-time simulation simplifying some assumptions of current complex forest fire simulators. In Bürger, Gavilán, Inzunza, Mulet, and Villada (2020b, 2020a) implement the Asensio and Ferragut (2002) model using implicit-explicit *Runge-Kutta* and weighted essentially non-oscillatory methods to handle the non-linear terms of the equations. Both papers explore topography and firebreak effects in fire dynamics.

Table 2.1 presents the numerical methods used by *PDE* models described before. Besides, a summary of model components is included in Table 2.2.

2.1.2 Cellular Automata

According to Weisstein (2002), a cellular automaton is a collection of “colored” cells on a grid of specified shape that evolves through several discrete time steps depending

Table 2.1: Numerical methods summary.

Reference	Discretization	
	Space	Time
Montenegro et al. (1997)	<i>FEM</i>	<i>Implicit Euler</i>
Santoni and Balbi (1998)	<i>FDM (CD)</i>	<i>FDM (FD)</i>
Balbi et al. (2000)	<i>FDM (CD)</i>	<i>FDM (FD)</i>
Asensio and Ferragut (2002)	<i>FEM</i>	<i>Euler</i>
Ferragut et al. (2007b)	<i>FEM</i>	<i>Semi-implicit Euler</i>
Ferragut et al. (2007a)		
Mandel et al. (2008)	<i>FDM (CD)</i>	<i>Euler</i>
Eberle (2014)	<i>RBF</i>	<i>Crank-Nicolson</i>
Eberle et al. (2015)		
Baranovskiy and Nee (2015)	<i>FDM</i>	<i>FDM</i>
Ferragut et al. (2015)	<i>FEM</i>	<i>Euler</i>
Perminov and Soprunenko (2016)	<i>FVM</i>	<i>SIMPLE Algorithm</i>
Perminov (2017)		
Perminov (2018)		
Toivanen et al. (2019)	<i>FVM</i>	<i>RK2</i>
Grasso and Innocente (2020)	<i>FDM (CD)</i>	<i>RK4</i>
Bürger et al. (2020b)	<i>FDM (WENO)</i>	<i>IMEX-RK</i>
Bürger et al. (2020a)		

on a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many time steps as desired. Due to its interesting results, *CA* have been widely used in different applications, one of these is the wildfire simulation. Below there is a brief review of work using this approach.

Hansen (1993) developed a generic parallel approach of a *CA* for multicomputers. The model is adapted for the numerical simulation of a forest fire where every cell represents a tree either alive, burning or dead. The system evolves using probabilistic rules that only depend on the domain fuel. Karafyllidis and Thanailakis (1997) present a model for fire spreading in a uniform and not uniform distribution of trees respectively, with an easy way to include weather conditions and land topography. The authors model the rates of fire spread at every point at a certain time. The fire front is determined by wind speed and direction combined with topography. Another model is proposed by Quartieri, Mastorakis, Iannone, and Guarnaccia (2010) introducing an “ignition probability” in the transition rules. The probability depends on surrounding cells, weather conditions, land, and fuel topology. With the stochastic feature, this work allows simulating irregular shapes in the fire front. Almeida and Macau (2011) proposed another stochastic approach under flat and no-wind conditions to model a wildfire. The dynamic of fire spreading is described using three possible states: vegetation, burning, and burnt cells. The changes depend on probabilities computed using the vegetation information of neighborhood cells. The work in (Iudin, Sergeyev, & Hayakawa, 2015) shows the connection between a percolation model and *CA* to describe forest fire dynamics. It is a crown fire spreading model that only uses fuel information. To avoid distortion in fire shapes, Ghisu, Arca, Pellizzaro, and Duce (2015) developed a fire model based on the Rothermel equation (Rothermel, 1972) and the *CA* approach. The model includes fuel, wind, and topography information. This work also presents a correction method using a modified *Tabu Search* to find optimal parameters in wildfire simulation. Eskandari (2016) presents a *CA* implementation of surface fire spreading to simulate fire front. The model uses the information of altitude and vegetation characteristics and tries to predict the fire front in the Hyrcanian forest of Iran. Wang et al. (2017) analyzed the use of *CA* to model surface fire spreading implemented as a system using MATLAB and including components such as meteorological factors, terrain, and combustible material. Mulyani, Putri, and Gunawan (2017) conduct a performance analysis of *OpenMP* to simulate surface fire spreading. The model is based on rules defined in Karafyllidis and Thanailakis (1997). Topography and wind are the main components used in this work. Fernandez-Anez, Christensen, and Rein (2017) proposed a two-dimensional multi-layer *CA* developed by the study of smoldering. The work includes three layers: fuel, heat, and oxygen, each with its state definition. The system evolves through the interaction between layers. Lei and Zhiqiang (2017) combine triangular fuzzy numbers with *CA* and *Monte Carlo* method to simulate wildfire dynamics. The states of *CA* represent if fuel is available, burning, or burnt. The model includes information of fuel and wind. In Liu, Liu, Zhou, and Sun (2018) a semi-empirical model to simulate surface and crown fires in real-time is presented.

This proposal includes physical equations to improve the quality of simulations, including reaction and radiation models. These models take into account fuel, wind, and topography information. Gouveia Freire and Castro Dacamura (2019) present a *CA* approach to help fire management. The model uses real data of topography, vegetation characteristics, and the wind data is collected from the *Weather Research and Forecasting (WRF) model* (National Center of Atmospheric Research, n.d.). This work tries to simulate the Algarve wildfire. Propagator (Trucchia et al., 2020) is a stochastic *CA* model developed for a rapid method for fire risk assessment. This model makes independent simulations in time-discrete steps to build a map with probabilities of affected zones by fire. Probabilities depend on fuel moisture, vegetation type, slope, and wind. The *CA* model in Mutthulakshmi et al. (2020) simulates forest fire and fire-fighting. Transition rules depend on wind effect, vegetation type, geography, and fire-fighting control strategies. Finally, Green, Deluca, and Kaiser (2020) combine a *CA* approach with *Genetic Programming (GP)*. *GP* is used to learn the transition rules of the *CA* model. The model is named *FireGen* and it is presented as an alternative to physics-based models. The model takes into account the topography elevation, wind, humidity, and homogeneous fuel. As in the previous subsection, a summary of model components using the *CA* approach is included in Table 2.2.

2.2 Wildfire Software

The construction of mathematical models has allowed the development of several computational tools, which depending on the models' complexity, can be executed from portable computers to high-performance clusters. This section presents a short brief of the current leading software, available for the study of forest fires.

BehavePlus

BehavePlus fire modeling system is a Windows-based computer program that can be used for any fire management application that involves modeling fire behavior and some fire effects. The system is composed of a collection of mathematical models that describe fire behavior and the fire environment. The program simulates the rate of fire spread, spotting distance, scorch height, tree mortality, fuel moisture, wind adjustment factor, as well as other variables; so it is used to predict fire behavior in multiple situations (FRAMES, n.d.; Andrews, 2014).

FARSITE

FARSITE is a fire growth simulation modeling system that uses spatial information on topography and fuels along with weather and wind files. It incorporates existing models for surface fire, crown fire, spotting, post-frontal combustion, and fire acceleration into a 2-dimensional fire growth model (*FARSITE — Fire, Fuel, and*

Table 2.2: Components of presented model approaches summary.

Approach	Model	Components		
		Fuel	Wind	Topography
PDE	Weber (1991b)	✓	✓	×
	Richards (1995)	✓	✓	×
	Grishin (1996)	✓	✓	✓
	Clark et al. (1996)	✓	✓	✓
	Montenegro et al. (1997)	✓	✓	×
	Balbi et al. (2000)	✓	×	×
	Asensio and Ferragut (2002)	✓	✓	✓
	Ferragut et al. (2007b)	✓	✓	✓
	Ferragut et al. (2007a)	✓	✓	✓
	Mandel et al. (2008)	✓	✓	✓
	Eberle (2014)	✓	✓	×
	Eberle et al. (2015)	✓	✓	✓
	Baranovskiy and Nee (2015)	✓	×	×
	Ferragut et al. (2015)	✓	✓	✓
	Guelpa et al. (2016)	✓	✓	×
	Perminov and Soprunenko (2016)	✓	✓	×
	Lopes et al. (2017)	✓	✓	✓
	Perminov (2017)	✓	✓	×
	Fečkan and Pačuta (2018)	✓	✓	✓
	Perminov (2018)	✓	✓	×
	Ambroz et al. (2018)	✓	✓	✓
	Marzaeva and Galtseva (2018)	✓	✓	×
	Toivanen et al. (2019)	✓	✓	✓
	Grasso and Innocente (2020)	✓	✓	✓
	Bürger et al. (2020a)	✓	✓	✓
	Bürger et al. (2020b)	✓	✓	✓
CA	Hansen (1993)	✓	×	×
	Karafyllidis and Thanailakis (1997)	✓	✓	✓
	Quartieri et al. (2010)	✓	✓	✓
	Almeida and Macau (2011)	✓	×	×
	Iudin et al. (2015)	✓	×	×
	Ghisu et al. (2015)	✓	✓	✓
	Eskandari (2016)	✓	✓	✓
	Wang et al. (2017)	✓	✓	✓
	Mulyani et al. (2017)	✓	✓	✓
	Fernandez-Anez et al. (2017)	✓	✓	×
	Lei and Zhiqiang (2017)	✓	✓	×
	Liu et al. (2018)	✓	✓	✓
	Gouveia Freire and Castro Dacamara (2019)	✓	✓	✓
	Trucchia et al. (2020)	✓	✓	✓
	Mutthulakshmi et al. (2020)	✓	✓	✓
	Green et al. (2020)	✓	✓	✓

Smoke Science Program, n.d.; Finney, 1998). Currently, it is included in *FlamMap* Software.

FLAMMAP

FlamMap is a fire analysis desktop application that runs in a 64-bit Windows Operating System environment. It can simulate potential fire behavior characteristics (spread rate, flame length, fire line intensity, etc.), fire growth and spread, and conditional burn probabilities under constant environmental conditions (weather and fuel moisture). With the inclusion of *FARSITE* it can now compute wildfire growth and behavior for longer periods under heterogeneous conditions of terrain, fuels, fuel moisture, and weather) (*FlamMap — Fire, Fuel, and Smoke Science Program*, n.d.).

MesoNH/Forefire

ForeFire is open-source software designed for large-scale wildfire simulations. Code is composed of a simulation engine that may be integrated with other scientific environments such as *Python*, *Fortran*, *C++*, and *Java* (Filippi, n.d.). Currently, exists a coupled version with Meso-NH for mesoscale non-hydrostatic atmospheric to improve *ForeFire* simulation capabilities (Filippi, 2018).

CAWFE

The *CAWFE* modeling system combines a numerical weather prediction (*NWP*) model that predicts how the weather varies in time and space even in complex terrain with wildland fire behavior modules. These components are connected in two directions such that the evolving wind, along with fuel properties and terrain slope, directs where the fire grows and how fast, while heat released by the fire modifies its atmospheric environment thereby creating its weather (e.g., fire-induced winds) (J. Coen, n.d.). The model is described by Clark, Coen, and Latham (2004); J. L. Coen (2005), The software documentation is available in (J. L. Coen, 2013). Currently this model is re-implemented in *WRF-Fire*.

WRF-Fire

WRF-Fire combines the *Weather Research and Forecasting (WRF)* model with a fire code implementing a surface fire behavior model, called *SFIRE*, based on semi-empirical formulas that calculate the rate of spread of the fire line (the interface between burning and unignited fuel) based on fuel properties, wind velocities from *WRF*, and terrain slope. The fire spread is implemented by the level set method. The heat release from the fire line as well as post-frontal heat release feeds back into *WRF* dynamics, affecting the simulated weather in the vicinity of the fire. The fire code is written in *Fortran 90* following *WRF* coding conventions. It is integrated as a physics option, called from *WRF* as a subroutine (Beezley & Mandel, n.d.). Details

of models and system development can be accessed in (Mandel, Beezley, Coen, & Kim, 2009; Mandel, Beezley, & Kochanski, 2011).

Fire Dynamic Simulator

Fire Dynamics Simulator (FDS) is a *large-eddy simulation (LES)* code for low-speed flows, with an emphasis on smoke and heat transport from fires (*FDS-SMV*, n.d.; McGrattan, Hostikka, Floyd, McDermott, & Vanella, 2019). *FDS* is a *computational fluid dynamics (CFD)* model of fire-driven fluid flow. It solves numerically a form of the Navier-Stokes equations appropriate for low-speed, thermally-driven flow with an emphasis on smoke and heat transport from fires. The formulation of the equations and the numerical algorithm are contained in the *FDS* Technical Reference Guide (McGrattan et al., 2019).

Wildfire Analyst

Wildfire Analyst (WFA) is software that provides real-time analysis of wildfire behavior and simulates the spread of wildfires. Simulations are completed quickly, in seconds, to support real-time decision-making. For wildland fire, time is of the essence, and Wildfire Analyst was specifically architected to support initial attack situations, giving the Fire Chief and Incident Commander the critical intelligence needed to support resource allocation decision making (Tecnosylva, n.d.). A detailed description of the system is presented in (Ramírez, Monedero, & Buckley, 2011).

Table 2.3 presents a summary of OS compatibility and code accessibility of the software listed before. It is important to point out that these tools contain all the model components indicated in Table 2.2.

Table 2.3: Software summary.

Software	OS Compatibility	Open-source
BehavePlus	Windows	×
FlamMap	Windows	✓ ^(*)
ForeFire	UNIX	✓
WRF-Fire	Linux, macOS	✓
FDS	Windows, macOS, Linux	✓
Wildfire Analyst	Android, iOS, Windows, macOS, Linux	×

(*) Only *FARSITE* model.

Chapter 3

Wildfire Mathematical Model

“Wildland fire mathematical models are generally composed of a collection of equations whose solution gives numerical values for the spatial/temporal evolution of one or more variables, such as rate of spread, flame height, ignition risk or fuel consumption” (Pastor et al., 2003). A remarkable model, based on the previous definition, was developed by Asensio and Ferragut (2002), and a similar one was derived by Eberle et al. (2015) introducing important assumptions and considerations in the model description. Many other similar models have been built by Weber (1991b), Montenegro et al. (1997), and Mandel et al. (2008).

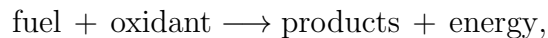
The main idea behind this theoretical phenomenon representation is to describe the behavior of temperature through diffusion, convection, and a reaction process. Diffusion represents the propagation of the temperature from a high to a low zone of heat, convection is the heat transfer induced by the effect of a fluid, such as the wind, and the reaction process is an exothermic chemical reaction between the temperature and vegetable fuel giving heat to the system. Also, the fuel dynamics are described by a nonlinear relation based on a modified version of the *Arrhenius equation* (Asensio & Ferragut, 2002).

In the following chapter, the theoretical foundation and the derivation of the model will be presented.

3.1 Model Derivation

Wildfire dynamics depend on various factors, which can be grouped into the combustion process and the heat/mass transfer mechanism (Eberle et al., 2015).

The combustion is related to the ignition process and fuel combustion. In this process, the fuel is determined by wood and forest vegetation, and the combustion depends on the chemical components of fuel. This phenomenon can be described by the following simplified reaction equation,



where fuel, composed of cellulose, reacts with oxygen to form dioxide and monoxide

carbon. At certain temperatures, many of these kinds of reactions are exothermic, generating and diffusing heat towards their surroundings.

The heat and mass transfer mechanism describes how the fire propagates by the interaction between particles. The mechanisms involved are:

- Advection or convection: it is a heat transport mechanism from one location to another by a fluid, such as streaming gases. Both terms are used as synonyms but technically the first one is the movement of something on the fluid and the second one is the movement of fluid in response to heat.
- Conduction or diffusion: it is understood as the heat transfer between different bodies by direct contact.
- Radiation: it is the transfer of energy by the emission of electromagnetic radiation. Associated with this process is the absorption caused by electromagnetic energy, which is transformed into internal energy absorbing thermal energy.

A general sketch of the involved mechanisms in a wildfire is presented in Figure 3.1.

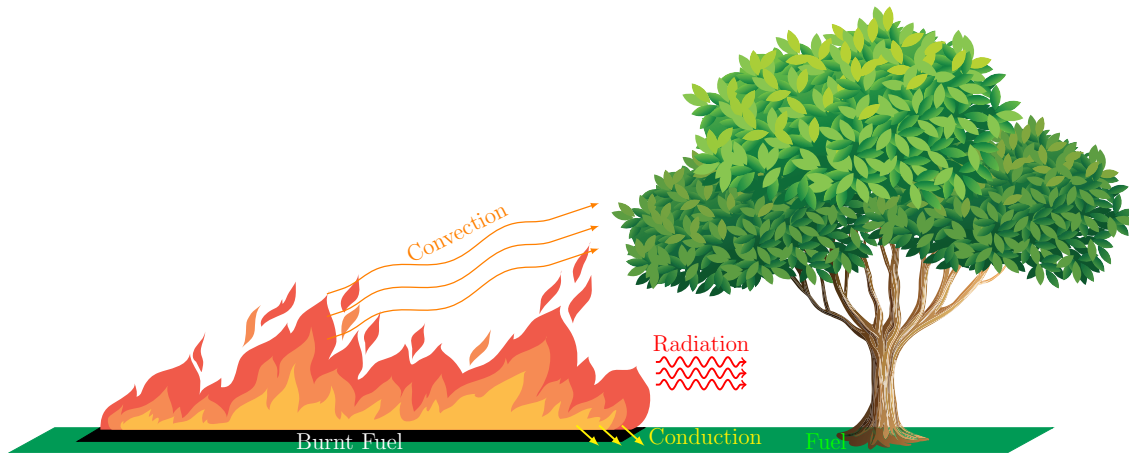


Figure 3.1: Wildfire components sketch. Images from Freepik.com

Considering the factors above, the fire generation time phases may be summarized as:

1. The ignition starts when a flame has contact with any kind of fuel depending on the composition of it.
2. Propagation and increasing of flames begin.
3. A fire ring is generated, where the center contains the burnt fuel with lower temperature.

4. Weather conditions and topography promote the growth of fire and the wildfire goes on.

From a theoretical point of view, Asensio and Ferragut (2002), Mandel et al. (2008), and Eberle et al. (2015) present a physical model based on the following assumptions:

- Fire reaction rate, to describe the speed of combustion.
- Balance equation of fuel supply, to describe the fuel consumption.
- Conservation of energy, to describe how the fuel energy is converted to thermal energy.

The next subsections present the mathematical formalization of the physical model. For all descriptions, let $V = V(t)$ be a bounded, simply connected set the arbitrary domain of consideration with a boundary $S = \partial V$, $\mathbf{x} \in V$ the spatial variable, and $t \in \mathbb{R}_0^+$ a temporal variable.

3.1.1 Fire Reaction rate

Both Asensio and Ferragut (2002) and Eberle et al. (2015) used the *Arrhenius equation* to model the fire reaction rate. This is an empirical relationship used to determine the rate of a chemical reaction and is defined as,

$$r = A \exp \left(-\frac{E_A}{RT} \right), \quad (3.1)$$

where $R = 8.31446261815324 \text{ [J}\cdot\text{K}^{-1}\cdot\text{mol}^{-1}]$ is the universal gas constant, $T = T(\mathbf{x}, t)$ the absolute temperature in [K], E_A represents the activation energy in $[\text{J}\cdot\text{mol}^{-1}]$ and A is the pre-exponential factor for the reaction in $[\text{s}^{-1}]$. This expression is important since it is used in the temperate and fuel dynamics that will be described below.

3.1.2 Balance equation of fuel supply

The following expression represents the mass of fuel m ,

$$\int_V \rho dV = m, \quad (3.2)$$

where ρ represents the density of fuel and dV is the volume element. Conservation of mass implies that the total derivative $\frac{d}{dt}$ of (3.2) is zero,

$$\frac{d}{dt} \left(\int_V \rho dV \right) = \frac{d}{dt} m = 0.$$

Eberle et al. (2015) introduces the mass change at the balance volume to describe the mass fraction consumption when fuel is burning. Assuming that m_i is a mass element contained in V , i the number of mass element, and let Y_i the mass fuel fraction concerning the i -th mass element, the following equation describes the dynamics of the mass element in terms of fuel fraction consumption:

$$\frac{\partial m_i}{\partial t} = \rho dV \frac{\partial Y_i}{\partial t}, \quad (3.3)$$

Figure 3.2 describes this process. Now, the focus is centered on the dynamic of the

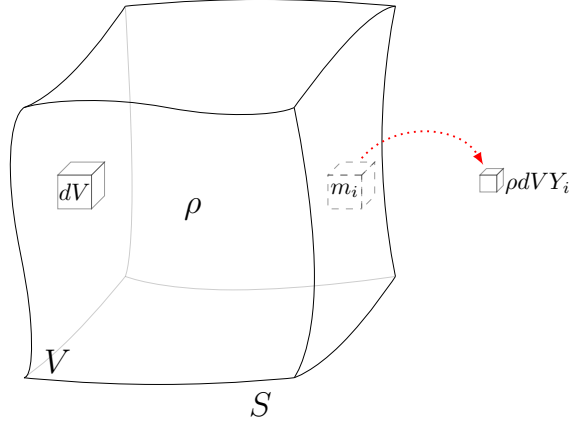


Figure 3.2: Fuel consumption.

mass fuel fraction term $\partial Y_i / \partial t$, considering $Y = Y(\mathbf{x}, t)$, the function which takes into account all the combustible. The balance equation of fuel supply states that the mass fuel fraction dynamics depends on the existing fuel fraction times the reaction rate r , that is,

$$\frac{\partial Y}{\partial t} = Y r.$$

Replacing r with *Arrhenius equation*, described in subsection 3.1.1, and taking into account that there is a fuel disappearance when it is burning, the balance equation for fuel supply is defined by:

$$\frac{\partial Y}{\partial t} = -Y A \exp\left(-\frac{E_A}{RT}\right). \quad (3.4)$$

3.1.3 Conservation of energy

Thermal energy is the main energy involved in the wildfire context. To represent its dynamics, conservation of energy is used considering the following three elements:

1. The thermal energy inside V , described by the following expression:

$$\int_V \rho c T dV,$$

where c is the heat capacity in $[\text{m}^2 \cdot \text{K}^{-1} \cdot \text{s}^{-2}]$, and $T = T(\mathbf{x}, t)$ is the temperature in $[\text{K}]$.

2. The thermal energy flowing through the boundary S , which is represented by

$$\int_S \mathbf{j} \cdot \hat{\mathbf{n}} dS,$$

with \mathbf{j} the thermal flux, $\hat{\mathbf{n}}$ the outer unit field at each point of S , and dS the normal surface element.

3. The thermal energy generated by a heat source s in V at t defined as

$$\int_V s dV.$$

Figure 3.3 presents the components used by the thermal energy conservation. Eberle

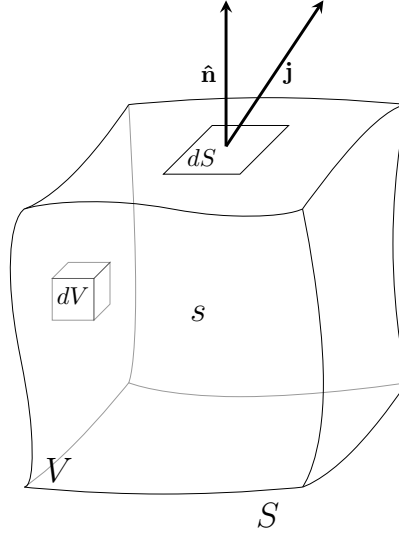


Figure 3.3: Sketch of the components for thermal energy dynamics formalization.

et al. (2015) establish the following integral equation to describe the thermal energy dynamics using the components defined above:

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = - \int_S \mathbf{j} \cdot \hat{\mathbf{n}} dS + \int_V s dV. \quad (3.5)$$

Expression (3.5) indicates that the rate of change of thermal energy is defined by the energy generated minus the energy that flows out. Applying the *Divergence Theorem*, equation (3.5) becomes to

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = - \int_V \nabla \cdot \mathbf{j} dV + \int_V s dV. \quad (3.6)$$

From the Reynolds transport theorem, the left-hand side of (3.6) turns into

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = \int_V \frac{\partial}{\partial t} (\rho c T) dV + \int_S \rho c T \mathbf{u} \cdot \hat{\mathbf{n}} dS, \quad (3.7)$$

where \mathbf{u} denotes the flux velocity where thermal energy is moving. Applying the *Divergence Theorem* to equation 3.7,

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = \int_V \frac{\partial}{\partial t} (\rho c T) dV + \int_V \nabla \cdot (\rho c T \mathbf{u}) dV. \quad (3.8)$$

Assuming ρ, c constant and \mathbf{u} an incompressible flow, the new expression for equation 3.8 is

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = \int_V \rho c \frac{\partial T}{\partial t} dV + \int_V \rho c \mathbf{u} \cdot \nabla T dV. \quad (3.9)$$

Details in Appendix A.1.1. Putting (3.9) into (3.6), the new expression of conservation of energy is

$$\int_V \left(\rho c \frac{\partial T}{\partial t} + \rho c \mathbf{u} \cdot \nabla T \right) dV = - \int_V \nabla \cdot \mathbf{j} dV + \int_V s dV. \quad (3.10)$$

Rearranging equation (3.10),

$$\int_V \left(\rho c \frac{\partial T}{\partial t} + \rho c \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{j} - s \right) dV = 0, \quad (3.11)$$

since the integration is over the same domain. Finally, integral can be removed because V is an arbitrary domain, therefore, the following *Partial Differential Equation (PDE)* can be obtained as conservation of thermal energy

$$\rho c \frac{\partial T}{\partial t} + \rho c \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{j} = s. \quad (3.12)$$

The next step is to find a detailed description of flow \mathbf{j} and source s . First, the source component includes a released heat Q and a cooling term to describe the natural convection as shown in Figure 3.4. This kind of convection represents the vertical heat loss associated with the effect of gravity caused by the distinct temperatures. Thus, the source component is represented by

$$s = -Q - h(T - T_\infty), \quad (3.13)$$

where T_∞ is a reference temperature, like ambient temperature, and h is the convective coefficient in $[\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}]$. Both Asensio and Ferragut (2002) and Eberle et al. (2015) model the energy released as an exothermic reaction which is represented by

$$Q = H S_f, \quad (3.14)$$

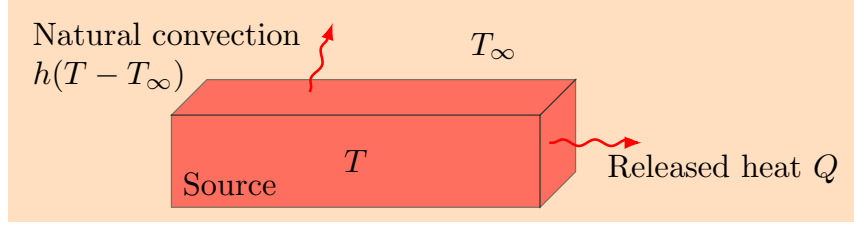


Figure 3.4: Sketch of the thermal energy generated by a source.

with H the heat of combustion expressed in $[\text{kJ} \cdot \text{kg}^{-1}]$. The term S_f is known as descent of educts, and it is defined as how a substance is separated from a mixture in which it already existed. Assuming that there is a sufficient mass amount of oxidant in the air Y_o comparing with the fuel Y_f , that is $Y_f \ll Y_o$, the reaction is controlled by Y_f , so $Y = Y_f$. Ignoring the conservation law for the oxidant, S_f is defined by

$$S_f = -r\rho Y = -A \exp\left(-\frac{E_A}{RT}\right) \rho Y, \quad (3.15)$$

where ρ is the density of the fuel.

On the other hand, a general manner to understand the total flow of the expression (3.12) is the following relationship:

$$\mathbf{j}_{\text{total}} = \mathbf{j}_{\text{convective}} + \mathbf{j}_{\text{diffusive}} + \mathbf{j}_{\text{radiative}} \quad (3.16)$$

Convective flow $\mathbf{j}_{\text{convective}}$ is a bulk motion of hot gas or diffusion by whirlwinds described by

$$\mathbf{j}_{\text{convective}} = \rho c \mathbf{u},$$

where ρ is density and c the specific heat of the fluid. In terms of modeling, \mathbf{u} includes the effect of wind and terrain slope. Convective flow is already included in equation (3.12), so let $\mathbf{j} = \mathbf{j}_{\text{diffusive}} + \mathbf{j}_{\text{radiative}}$. To describe the diffusive flow, Fourier's law is used. This law says that the heat flux density is equal to the product of thermal conductivity k and the negative local temperature gradient, that is

$$\mathbf{j}_{\text{diffusive}} = -k \nabla T. \quad (3.17)$$

Asensio and Ferragut (2002) use the Stefan-Boltzmann law for heat flux transfer by radiation between objects as,

$$\mathbf{j}_{\text{radiation}} = -\sigma (T^4(x + \delta) - T^4(x)),$$

where $\sigma = 5.670374419 \dots \times 10^{-8} [\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-4}]$ is the Stefan-Boltzmann constant, δ is the optical path length for radiation through the fuel and T the absolute temperature. When $\delta < dx$ and by truncating high order terms in Taylor series of $T^4(x + \delta)$, detailed in Appendix A.1.2, the thermal flux radiation is defined by

$$\mathbf{j}_{\text{radiation}} = -4\sigma \delta T^3 \nabla T. \quad (3.18)$$

Taking equations (3.17) and (3.18), the flux \mathbf{j} is:

$$\begin{aligned}\mathbf{j} &= \mathbf{j}_{\text{diffusive}} + \mathbf{j}_{\text{radiative}} \\ &= -k\nabla T - 4\sigma\delta T^3\nabla T \\ &= -(k + 4\sigma\delta T^3)\nabla T\end{aligned}\tag{3.19}$$

3.1.4 Final Model

Pyrolysis of wood is a complex phenomenon, but for the scope of this thesis and in terms of modelling there are two phases in the process: the endothermic or solid phase and exothermic or gaseous phase. To include these phases in the model, the following phase change function is introduced in both temperature and fuel dynamics equations:

$$H_{pc}(T) = \begin{cases} 1 & \text{if } T \geq T_{pc} \\ 0 & \text{otherwise.} \end{cases}\tag{3.20}$$

where T_{pc} is the phase change temperature that represents both phases in the same system. According to Asensio and Ferragut (2002), the change of phases occurs approximately in $T_{pc} = 550$ [K].

Putting together the expressions obtained in subsections 3.1.1, 3.1.2 and 3.1.3, the simplified model is

$$\begin{aligned}\rho c \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) - \nabla \cdot ((4\sigma\delta T^3 + k)\nabla T) &= H_{pc}(T)HA \exp\left(-\frac{E_A}{RT}\right) \rho Y - h(T - T_\infty), \\ \frac{\partial Y}{\partial t} &= -H_{pc}(T)YA \exp\left(-\frac{E_A}{RT}\right).\end{aligned}\tag{3.21}$$

This model is defined in a two-dimensional domain $(x, y) \in \Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega = \Gamma$ large enough to assume that temperature T and the mass fraction of fuel Y do not change on the boundary during $t \in [0, t_{\max}] \subseteq \mathbb{R}_0^+$. This assumption allows the use of Dirichlet or Neumann boundary conditions. The initial conditions are $T(x, y, 0) = T_0 \geq T_\infty$ and $Y(x, y, 0) = Y_0 \geq 0$.

3.2 Model analysis

3.2.1 Parameters

The physical validity of computational results requires a brief analysis of the model parameters, for instance, temperatures consistent with the real-world. To introduce the nature of the parameters, Table 3.1 presents the metrics units used in the wildfire context. Also, a summary of physical parameters and the values reported by Asensio and Ferragut (2002) are presented in Table 3.2.

Table 3.1: Metric units in wildfire context.

Unit	Symbol	SI base
Amount of substance	mol	
Distance	m	
Energy	J	$\text{kg} \cdot \text{m}^2 \cdot \text{s}^{-2}$
Mass	kg	
Temperature	K	
Time	s	
Power	W	$\text{kg} \cdot \text{m}^2 \cdot \text{s}^{-3}$

Table 3.2: Value range of the wildfire model's physical parameters (Asensio & Ferragut, 2002).

Parameter	Symbol	Units	Typical Values
Thermal conductivity	κ	$\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$	Maple/oak 0.059, sadwust 1.17, air 0.024.
Optical path length	δ	m	No reported.
Density	ρ	$\text{kg} \cdot \text{m}^{-3}$	Fir 420, yellow pine 640, air 1.1774 ^(*) .
Specific heat	c	$\text{m}^2 \cdot \text{K}^{-1} \cdot \text{s}^{-2}$	Maple 2400, yellow pine 2800, air 1005.7.
Natural convection	h	$\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$	No reported.
Activation energy	E_A	$\text{J} \cdot \text{mol}^{-1}$	Estimated value 83680
Heat of combustion	H	$\text{kJ} \cdot \text{kg}^{-1}$	Cellulose 66525.6 ^(**) .
Pre-exponential factor	A	s^{-1}	Admissible value 10^9 .
Reference temperature	T_∞	K	Estimated value 300
Phase change temperature	T_{pc}	K	Estimated value 550

^(*)At atmospheric pressure and ambient temperature.

^(**)Realistic cases may consider smaller values because there are other components in fuel with lower heats of combustion.

3.2.2 Nondimensionalization

To reduce the number of parameters of (3.21), Asensio and Ferragut (2002) and Eberle et al. (2015) used the Frank–Kamenetskii change of variable. Let t_0, l_0 characteristic magnitudes for temporal and spatial variables, the non-dimensional change of variables is

$$\xi = \frac{x}{l_0}, \quad \eta = \frac{y}{l_0}, \quad \tau = \frac{t}{t_0}, \quad u = \frac{T - T_\infty}{\varepsilon T_\infty}, \quad \beta = \frac{Y}{Y_0}, \quad \mathbf{v} = \frac{l_0}{t_0} \mathbf{u} \quad (3.22)$$

where $\varepsilon = \frac{RT_\infty}{E_A}$ is the inverse of activation energy. Also (Asensio & Ferragut, 2002) defines the non-dimensional reaction heat $q = \frac{HY_0}{cT_\infty}$, and set $t_0 = \frac{\varepsilon \exp(1/\varepsilon)}{qA}$, $l_0 = \sqrt{\frac{t_0 k}{\rho c}}$. Now, it is necessary to replace the new variables into each member of (3.21). For first member,

$$\begin{aligned} \rho c \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) &= \rho c \left(\frac{\partial(\varepsilon T_\infty u + T_\infty)}{\partial t} + \frac{l_0}{t_0} \mathbf{v} \cdot \left(\frac{\partial(\varepsilon T_\infty u + T_\infty)}{\partial x}, \frac{\partial(\varepsilon T_\infty u + T_\infty)}{\partial y} \right) \right) \\ &= \rho c \left(\varepsilon T_\infty \frac{\partial u}{\partial t} + \frac{l_0}{t_0} \mathbf{v} \cdot \left(\varepsilon T_\infty \frac{\partial u}{\partial x}, \varepsilon T_\infty \frac{\partial u}{\partial y} \right) \right) \\ &= \rho c \varepsilon T_\infty \left(\frac{\partial u}{\partial \tau} \frac{d\tau}{dt} + \frac{l_0}{t_0} \mathbf{v} \cdot \left(\frac{\partial u}{\partial \xi} \frac{d\xi}{dx}, \frac{\partial u}{\partial \eta} \frac{d\eta}{dy} \right) \right). \end{aligned}$$

From (3.22) is known that $\frac{d\tau}{dt} = \frac{1}{t_0}$, $\frac{d\xi}{dx} = \frac{1}{l_0}$ and $\frac{d\eta}{dy} = \frac{1}{l_0}$, so,

$$\begin{aligned} \rho c \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) &= \rho c \varepsilon T_\infty \left(\frac{\partial u}{\partial \tau} \frac{1}{t_0} + \frac{l_0}{t_0} \mathbf{v} \cdot \left(\frac{\partial u}{\partial \xi} \frac{1}{l_0}, \frac{\partial u}{\partial \eta} \frac{1}{l_0} \right) \right) \\ &= \rho c \varepsilon T_\infty \left(\frac{1}{t_0} \frac{\partial u}{\partial \tau} + \frac{l_0}{t_0} \mathbf{v} \cdot \frac{1}{l_0} \left(\frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \eta} \right) \right) \\ &= \rho c \varepsilon T_\infty \left(\frac{1}{t_0} \frac{\partial u}{\partial \tau} + \frac{1}{t_0} \mathbf{v} \cdot \left(\frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \eta} \right) \right) \quad (3.23) \\ &= \frac{\rho c \varepsilon T_\infty}{t_0} \left(\frac{\partial u}{\partial \tau} + \mathbf{v} \cdot \left(\frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \eta} \right) \right) \\ &= \frac{\rho c \varepsilon T_\infty}{t_0} (u_\tau + \mathbf{v} \cdot \bar{\nabla} u) \end{aligned}$$

The operator $\bar{\nabla} = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta} \right)$ is introduced to differentiate it from $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$. For the next member,

$$\begin{aligned}
 \nabla \cdot ((4\sigma\delta T^3 + k)\nabla T) &= \nabla \cdot \left((4\sigma\delta(\varepsilon T_\infty u + T_\infty)^3 + k) \frac{\varepsilon T_\infty}{l_0} \left(\frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \eta} \right) \right) \\
 &= \frac{\varepsilon T_\infty}{l_0} \nabla \cdot \left((4\sigma\delta T_\infty^3(\varepsilon u + 1)^3 + k) \left(\frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \eta} \right) \right) \\
 &= \frac{\varepsilon T_\infty}{l_0} \nabla \cdot \left(k \left(\frac{4\sigma\delta T_\infty^3}{k}(\varepsilon u + 1)^3 + 1 \right) \bar{\nabla} u \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0} \nabla \cdot \left(\left(\frac{4\sigma\delta T_\infty^3}{k}(\varepsilon u + 1)^3 + 1 \right) \bar{\nabla} u \right).
 \end{aligned}$$

Defining $K(u) = \bar{\kappa}(\varepsilon u + 1)^3 + 1$ and $\bar{\kappa} = \frac{4\sigma\delta T_\infty^3}{k}$,

$$\begin{aligned}
 \nabla \cdot ((4\sigma\delta T^3 + k)\nabla T) &= \frac{\varepsilon T_\infty k}{l_0} \nabla \cdot (K(u) \bar{\nabla} u) \\
 &= \frac{\varepsilon T_\infty k}{l_0} \nabla \cdot \left(K(u) \left(\frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \eta} \right) \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0} \nabla \cdot \left(K(u) \frac{\partial u}{\partial \xi}, K(u) \frac{\partial u}{\partial \eta} \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0} \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right) \cdot \left(K(u) \frac{\partial u}{\partial \xi}, K(u) \frac{\partial u}{\partial \eta} \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0} \left(\frac{\partial}{\partial x} \left(K(u) \frac{\partial u}{\partial \xi} \right) + \frac{\partial}{\partial y} \left(K(u) \frac{\partial u}{\partial \eta} \right) \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0} \left(\frac{\partial}{\partial \xi} \left(K(u) \frac{\partial u}{\partial \xi} \right) \frac{d\xi}{dx} + \frac{\partial}{\partial \eta} \left(K(u) \frac{\partial u}{\partial \eta} \right) \frac{d\eta}{dy} \right) \quad (3.24) \\
 &= \frac{\varepsilon T_\infty k}{l_0} \left(\frac{\partial}{\partial \xi} \left(K(u) \frac{\partial u}{\partial \xi} \right) \frac{1}{l_0} + \frac{\partial}{\partial \eta} \left(K(u) \frac{\partial u}{\partial \eta} \right) \frac{1}{l_0} \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0^2} \left(\frac{\partial}{\partial \xi} \left(K(u) \frac{\partial u}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left(K(u) \frac{\partial u}{\partial \eta} \right) \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0^2} \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta} \right) \cdot \left(K(u) \frac{\partial u}{\partial \xi}, K(u) \frac{\partial u}{\partial \eta} \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0^2} \bar{\nabla} \cdot \left(K(u) \frac{\partial u}{\partial \xi}, K(u) \frac{\partial u}{\partial \eta} \right) \\
 &= \frac{\varepsilon T_\infty k}{l_0^2} \bar{\nabla} \cdot (K(u) \bar{\nabla} u).
 \end{aligned}$$

The right hand side of temperature in (3.21) is

$$\begin{aligned}
 H_{pc}(T)HA \exp\left(-\frac{E_A}{RT}\right) \rho Y &= H_{pc}(u) \frac{cT_\infty q}{Y_0} \frac{\varepsilon \exp(1/\varepsilon)}{t_0 q} \exp\left(-\frac{E_A}{R(\varepsilon T_\infty u + T_\infty)}\right) \rho Y_0 \beta \\
 &= H_{pc}(u) \frac{cT_\infty \varepsilon \exp(1/\varepsilon)}{t_0} \exp\left(-\frac{E_A}{RT_\infty(\varepsilon u + 1)}\right) \rho \beta \\
 &= H_{pc}(u) \frac{\rho c \varepsilon T_\infty}{t_0} \beta \exp\left(\frac{1}{\varepsilon}\right) \exp\left(-\frac{1}{\varepsilon(\varepsilon u + 1)}\right) \\
 &= H_{pc}(u) \frac{\rho c \varepsilon T_\infty}{t_0} \beta \exp\left(\frac{1}{\varepsilon} - \frac{1}{\varepsilon(\varepsilon u + 1)}\right) \\
 &= H_{pc}(u) \frac{\rho c \varepsilon T_\infty}{t_0} \beta \exp\left(\frac{u}{\varepsilon u + 1}\right),
 \end{aligned} \tag{3.25}$$

and,

$$h(T - T_\infty) = h\varepsilon T_\infty u. \tag{3.26}$$

Replacing equations (3.23), (3.24), (3.25), and (3.26) into temperature dynamics equation,

$$\begin{aligned}
 \frac{\rho c \varepsilon T_\infty}{t_0} (u_\tau + \mathbf{v} \cdot \bar{\nabla} u) + \frac{\varepsilon T_\infty k}{l_0^2} \bar{\nabla} \cdot (K(u) \bar{\nabla} u) &= H_{pc}(u) \frac{\rho c \varepsilon T_\infty}{t_0} \beta \exp\left(\frac{u}{\varepsilon u + 1}\right) - h\varepsilon T_\infty u \\
 u_\tau + \mathbf{v} \cdot \bar{\nabla} u + \frac{t_0 k}{l_0^2 \rho c} \bar{\nabla} \cdot (K(u) \bar{\nabla} u) &= H_{pc}(u) \beta \exp\left(\frac{u}{\varepsilon u + 1}\right) - \frac{t_0 h}{\rho c} u \\
 u_\tau + \mathbf{v} \cdot \bar{\nabla} u + \frac{l_0^2}{l_0^2} \bar{\nabla} \cdot (K(u) \bar{\nabla} u) &= H_{pc}(u) \beta \exp\left(\frac{u}{\varepsilon u + 1}\right) - \alpha u \\
 u_\tau + \mathbf{v} \cdot \bar{\nabla} u + \bar{\nabla} \cdot (K(u) \bar{\nabla} u) &= H_{pc}(u) \beta \exp\left(\frac{u}{\varepsilon u + 1}\right) - \alpha u,
 \end{aligned} \tag{3.27}$$

with $\alpha = \frac{t_0 h}{\rho c}$. For fuel dynamics, the first term

$$\begin{aligned}
 \frac{\partial Y}{\partial t} &= \frac{\partial(Y_0 \beta)}{\partial t} \\
 &= Y_0 \frac{\partial \beta}{\partial t} \\
 &= Y_0 \frac{\partial \beta}{\partial \tau} \frac{d\tau}{dt} \\
 &= Y_0 \frac{\partial \beta}{\partial \tau} \frac{1}{t_0} \\
 &= \frac{Y_0}{t_0} \frac{\partial \beta}{\partial \tau} \\
 &= \frac{Y_0}{t_0} \beta_\tau,
 \end{aligned} \tag{3.28}$$

and the second term,

$$\begin{aligned}
 H_{pc}(T)Y A \exp\left(-\frac{E_A}{RT}\right) &= H_{pc}(u)Y_0\beta \frac{\varepsilon \exp(1/\varepsilon)}{t_0q} \exp\left(-\frac{E_A}{R(\varepsilon T_\infty u + T_\infty)}\right) \\
 &= H_{pc}(u) \frac{Y_0\varepsilon\beta \exp(1/\varepsilon)}{t_0q} \exp\left(-\frac{E_A}{RT_\infty(\varepsilon u + 1)}\right) \\
 &= H_{pc}(u) \frac{Y_0\varepsilon}{t_0q} \beta \exp\left(\frac{1}{\varepsilon}\right) \exp\left(-\frac{1}{\varepsilon(\varepsilon u + 1)}\right) \\
 &= H_{pc}(u) \frac{Y_0\varepsilon}{t_0q} \beta \exp\left(\frac{1}{\varepsilon} - \frac{1}{\varepsilon(\varepsilon u + 1)}\right) \\
 &= H_{pc}(u) \frac{Y_0\varepsilon}{t_0q} \beta \exp\left(\frac{u}{\varepsilon u + 1}\right)
 \end{aligned} \tag{3.29}$$

Replacing equations (3.28) and (3.29) into fuel balance equation,

$$\begin{aligned}
 \frac{Y_0}{t_0}\beta_\tau &= -H_{pc}(u) \frac{Y_0\varepsilon}{t_0q} \beta \exp\left(\frac{u}{\varepsilon u + 1}\right) \\
 \beta_\tau &= -H_{pc}(u) \frac{\varepsilon}{q} \beta \exp\left(\frac{u}{\varepsilon u + 1}\right)
 \end{aligned} \tag{3.30}$$

Finally, putting together equations (3.27) and (3.30), the non-dimensional version of the model (3.21) is defined as

$$\begin{aligned}
 u_\tau + \mathbf{v} \cdot \bar{\nabla} u &= \bar{\nabla} \cdot (K(u)\bar{\nabla} u) + f(u, \beta) \\
 \beta_\tau &= g(u, \beta)
 \end{aligned} \tag{3.31}$$

where

$$\begin{aligned}
 K(u) &= \bar{\kappa}(1 + \varepsilon u)^3 + 1, \\
 f(u, \beta) &= H_{pc}(u)\beta \exp\left(\frac{u}{1 + \varepsilon u}\right) - \alpha u, \\
 g(u, \beta) &= -H_{pc}(u) \frac{\varepsilon}{q} \beta \exp\left(\frac{u}{1 + \varepsilon u}\right), \\
 H_{pc}(u) &= \begin{cases} 1 & \text{if } u \geq u_{pc} \\ 0 & \text{other} \end{cases}, \quad \text{with } u_{pc} = \frac{T_{pc} - T_\infty}{\varepsilon T_\infty}
 \end{aligned} \tag{3.32}$$

To simplify the notation, the use of typical variables is adopted, thus, t instead of τ for time, and x, y instead of ξ, η for space. The main difference between the model presented by Asensio and Ferragut (2002) with the version of Eberle et al. (2015), is the definition of $K(u)$, where the second version set this function as a suitable constant κ . The importance of nondimensionalization, in addition to the considerable reduction of parameters, allows linking the reference values described in Table 3.2 with the data used by the model. This makes it possible to connect real-world data with the data to be used by the model in the numerical simulations.

Chapter 4

Proposed Algorithm

The proposal of this work is the development of an open-source framework to complement the analysis of forest fires in Chile, based on the physical spreading models of Asensio and Ferragut (2002) and Eberle et al. (2015). Emphasis is placed on the construction of a numerical method considering appropriate convergence properties for the model approximation used.

The main application of this work is the generation of vulnerability zones or risk maps through multiple independent numerical simulations. To generate this output, it is proposed the scheme in Figure 4.1, where, given a certain configuration of initial conditions and different ignition points, this map is generated considering criteria such as amount of fuel affected, fuel consumption times, among others. It can be seen that this process requires a high computer consumption, which is directly related to the efficiency in the implementation of the algorithm and numerical method associated with the model. Also, it can be noted that it is possible to obtain better results by increasing the domain resolution and number of different initial conditions, which increases the computing times. To address this excessive computation, it is proposed to design and implement an algorithm that is compatible with high-performance computing techniques such as the use of Graphics Processing Units.

Topographic and wind data used by the model will be loaded into the framework as input in a format to be defined. The numerical simulation scale of the fires corresponds to hourly propagation, no more than one day, so the effect of day/night wind will be reflected in the data used. Regarding the seasonal effect of the data, they are included in the parameter settings of the mathematical model that will be used.

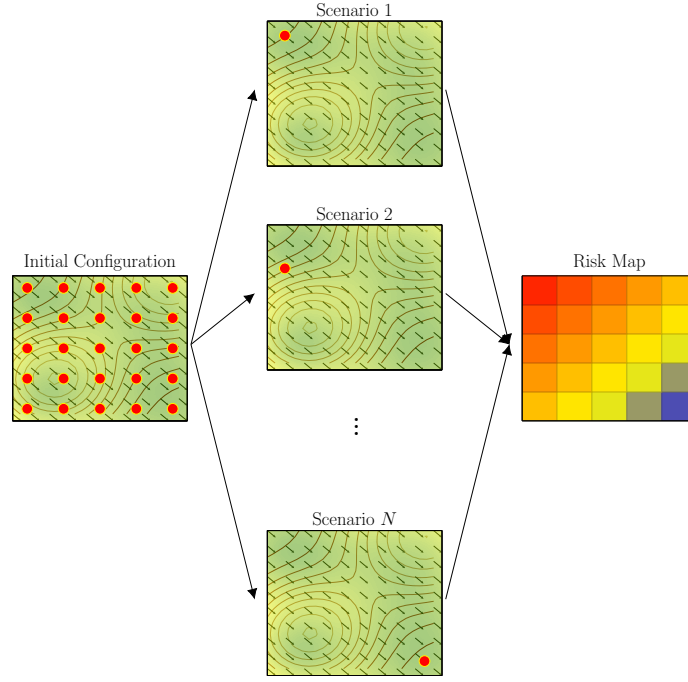


Figure 4.1: Diagram of risk map generation.

4.1 Working Hypothesis

The hypothesis developed in this work is the following:

Based on a mathematical model of wildfire spreading, a parallel algorithm is proposed on a GPU approach that will reduce the computational execution time of multiple simulations compared to a CPU implementation.

4.2 Simplified Two-dimensional Model

The mathematical model used for this thesis is based on the proposed by Asensio and Ferragut (2002) with the simplification introduced by Eberle et al. (2015) and which derivation was presented in Chapter 3. Next, a summary of the model.

Assuming a spatial domain $\mathbf{x} = (x, y) \in \Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \subset \mathbb{R}^2$ with boundary $\Gamma = \partial\Omega$, and a temporal domain $t \in [0, t_{\max}] \subseteq \mathbb{R}_0^+$. Let $u(\mathbf{x}, t)$ be the numerical value of the temperature and $\beta(\mathbf{x}, t)$ the amount of fuel, both at spatial coordinate \mathbf{x} and at time t . The vector field that models the effect of wind $\mathbf{w}(t)$ and topography $Z(\mathbf{x})$ is $\mathbf{v}(\mathbf{x}, t) = \mathbf{w}(t) + \nabla Z(\mathbf{x})$, and $f(u, \beta)$ is a nonlinear heat source that describes the interaction between fuel and temperature. The simplified mathematical model of wildfire spreading, based on the equations 3.31 and 3.32

derived in Chapter 3, is defined as

$$\begin{aligned}
 u_t &= \kappa \Delta u - \mathbf{v} \cdot \nabla u + f(u, \beta), & \text{in } \Omega \times (0, t_{\max}], \\
 \beta_t &= g(u, \beta), & \text{in } \Omega \times (0, t_{\max}], \\
 u(\mathbf{x}, t) &= u_\Gamma(\mathbf{x}, t), & \text{on } \Gamma \times (0, t_{\max}], \\
 \beta(\mathbf{x}, t) &= \beta_\Gamma(\mathbf{x}, t), & \text{on } \Gamma \times (0, t_{\max}], \\
 u(\mathbf{x}, 0) &= u_0(\mathbf{x}), & \text{in } \Omega, \\
 \beta(\mathbf{x}, 0) &= \beta_0(\mathbf{x}), & \text{in } \Omega,
 \end{aligned} \tag{4.1}$$

where,

$$\begin{aligned}
 f(u, \beta) &= H_{pc}(u) \beta \exp\left(\frac{u}{1 + \varepsilon u}\right) - \alpha u, \\
 g(u, \beta) &= -H_{pc}(u) \frac{\varepsilon}{q} \beta \exp\left(\frac{u}{1 + \varepsilon u}\right), \\
 H_{pc}(u) &= \begin{cases} 1, & \text{if } u \geq u_{pc}, \\ 0, & \text{otherwise.} \end{cases}
 \end{aligned}$$

Since the model is defined in a 2-dimensional domain the gradient is defined as $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$ and the Laplace operator $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. The nondimensional parameters of the model are diffusion coefficient κ , the inverse of activation energy of fuel ε , natural convection α , reaction heat q , and the phase change threshold u_{pc} . The parameters used by this work are based on the range presented in Table 3.2, and the values reported in Asensio and Ferragut (2002). Some values were modified for the analysis of the algorithm and according to the requirements of each numerical simulation. Values are detailed in Table 4.1.

Table 4.1: Physical parameters used by this thesis.

Parameter	Used Value
Thermal conductivity (κ)	1 [W · m ⁻¹ · K ⁻¹]
Density (ρ)	100 [kg · m ⁻³]
Specific heat (c)	1000 [m ² · K ⁻¹ · s ⁻²]
Activation energy (E_A)	8100 [J · mol ⁻¹]
Heat of combustion (H)	66525.6 [kJ · kg ⁻¹]
Pre-exponential factor (A)	0.001 [s ⁻¹]
Reference temperature (T_∞)	300 [K]
Phase change temperature (T_{pc})	550 [K]
Characteristic length (l_0)	0.29 [m]
Characteristic time (t_0)	8409.49 [s]

An assumption used in this work is that the spatial domain is large enough to avoid fire spreading at the boundary of Ω , then the following Dirichlet boundary conditions are used: $u_\Gamma(\mathbf{x}, t) = \beta_\Gamma(\mathbf{x}, t) = 0$ on $\Gamma \times (0, t_{\max}]$. Nevertheless, the model can be used with Neumann boundary conditions. The components used by the model are presented in Figure 4.2.

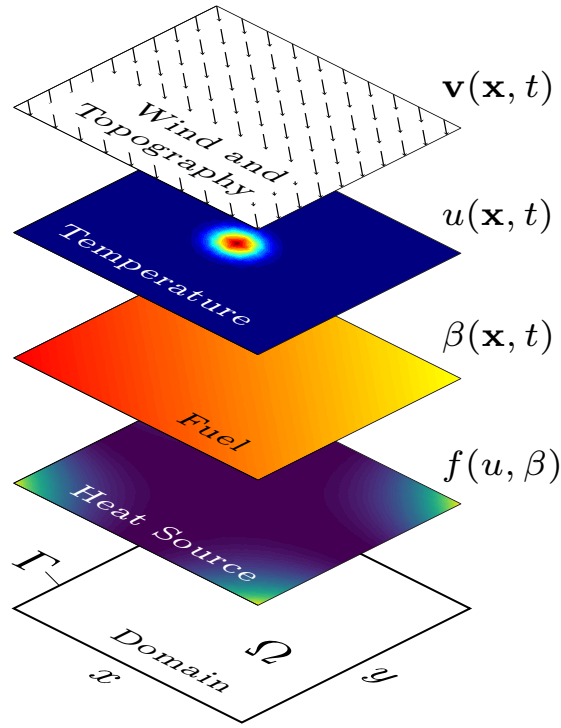


Figure 4.2: Sketch of model components (San Martín & Torres, 2018d).

Chapter 5

Numerical Algorithm

Numerical implementation of (4.1) requires the approximation of spatial and time partial derivatives involved in the problem formulation. To address these approximations, this work uses the *Method of Lines*, technique for numerical solving of *PDEs* in which all but one dimension is discretized. The following sections will describe the methods used to approximate the partial derivatives indicated previously.

5.1 Method of Lines

Method of Lines (MOL) is an important approach for the numerical solving of *PDEs*. This method consists of two parts, space discretization and time integration (Verwer & Sanz-Serna, 1984). The space discretization approximates the *PDE* by a system of *Ordinary Differential Equations (ODEs)* by discretizing the space domain using finite difference, finite elements, spectral methods, among others. Here, time is the independent variable of the *ODE* system. In the second step, the system is solved using some numerical method for time integration, discretizing the time domain to get a fully discrete scheme.

There are several works, such as Sarmin and Chudov (1963); Zafarullah (1970); Verwer and Sanz-Serna (1984); Reddy and Trefethen (1992), discussing the stability and convergence of the method for some particular problems and this thesis uses *MOL* as a mathematical framework to solve (4.1) numerically.

5.2 Spatial Approximation

In the mathematical model used for this work, temperature dynamics need the computation of Δu and ∇u . To deal with these space partial derivatives, a discretization on a rectangular grid of $N_x + 1$ by $N_y + 1$ nodes is established. Let x_i and y_j the discrete version of the spatial domain, defined as

$$\begin{aligned} x_i &= x_{\min} + i\Delta x, \quad i = 0, 1, \dots, N_x, \\ y_j &= y_{\min} + j\Delta y, \quad j = 0, 1, \dots, N_y, \end{aligned} \tag{5.1}$$

where $\Delta x = (x_{\max} - x_{\min})/N_x$ and $\Delta y = (y_{\max} - y_{\min})/N_y$. Now, temperature component is defined as $u_{i,j}(t) = u(x_i, y_j, t)$ and stored in the matrix $U(t)$ described by

$$U(t) = \begin{pmatrix} u_{0,0}(t) & \dots & u_{N_x,0}(t) \\ \vdots & \ddots & \vdots \\ u_{0,N_y}(t) & \dots & u_{N_x,N_y}(t) \end{pmatrix}. \quad (5.2)$$

Following this ordering, fuel component $\beta_{i,j}(t) = \beta(x_i, y_j, t)$ is

$$B(t) = \begin{pmatrix} \beta_{0,0}(t) & \dots & \beta_{N_x,0}(t) \\ \vdots & \ddots & \vdots \\ \beta_{0,N_y}(t) & \dots & \beta_{N_x,N_y}(t) \end{pmatrix}, \quad (5.3)$$

and the vector field $\mathbf{v}(x_i, y_j, t) = (v_1(x_i, y_j, t), v_2(x_i, y_j, t)) = (v_{1,i,j}(t), v_{2,i,j}(t))$ is stored on the grid by the matrices $V_1(t), V_2(t)$ defined as follows

$$V_1(t) = \begin{pmatrix} v_{10,0}(t) & \dots & v_{1N_x,0}(t) \\ \vdots & \ddots & \vdots \\ v_{1N_y,0}(t) & \dots & v_{1N_x,N_y}(t) \end{pmatrix}, \quad (5.4)$$

$$V_2(t) = \begin{pmatrix} v_{20,0}(t) & \dots & v_{2N_x,0}(t) \\ \vdots & \ddots & \vdots \\ v_{20,N_y}(t) & \dots & v_{2N_x,N_y}(t) \end{pmatrix}. \quad (5.5)$$

Analogous for functions $f(x_i, y_j, t) = f_{i,j}(t)$ and $g(x_i, y_j, t) = g_{i,j}(t)$,

$$F(t) = \begin{pmatrix} f_{0,0}(t) & \dots & f_{N_x,0}(t) \\ \vdots & \ddots & \vdots \\ f_{0,N_y}(t) & \dots & f_{N_x,N_y}(t) \end{pmatrix}, \quad (5.6)$$

$$G(t) = \begin{pmatrix} g_{0,0}(t) & \dots & g_{N_x,0}(t) \\ \vdots & \ddots & \vdots \\ g_{0,N_y}(t) & \dots & g_{N_x,N_y}(t) \end{pmatrix}. \quad (5.7)$$

Now, partial derivatives will be computed using the *Finite Differences Method* by *Differentiation Matrices*.

5.2.1 Differentiation Matrices

Following the approach of Trefethen (2000), a *Central Finite Difference Method (CD FDM)* approximation for the first and second spatial partial derivative (Sauer, 2011) is used. An extension of *Forward* and *Backward Difference* is used to keep the order

of accuracy at the boundary of the domain. For the first partial derivative with respect to x ,

$$\left. \frac{\partial u}{\partial x} \right|_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + O(\Delta x^2), \quad i = 1, \dots, N_x - 1, j = 0, \dots, N_y, \quad (5.8)$$

$$\left. \frac{\partial u}{\partial x} \right|_{i,j} = \frac{-3u_{i,j} + 4u_{i+1,j} - u_{i+2,j}}{2\Delta x} + O(\Delta x^2), \quad i = 0, j = 0, \dots, N_y, \quad (5.9)$$

$$\left. \frac{\partial u}{\partial x} \right|_{i,j} = \frac{u_{i-2,j} - 4u_{i-1,j} + 3u_{i,j}}{2\Delta x} + O(\Delta x^2), \quad i = N_x, j = 0, \dots, N_y. \quad (5.10)$$

For the second partial derivative with respect to x ,

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + O(\Delta x^2), \quad i = 1, \dots, N_x - 1, j = 0, \dots, N_y, \quad (5.11)$$

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{i,j} = \frac{2u_{i,j} - 5u_{i+1,j} + 4u_{i+2,j} - u_{i+3,j}}{\Delta x^2} + O(\Delta x^2), \quad i = 0, j = 0, \dots, N_y, \quad (5.12)$$

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{i,j} = \frac{-u_{i-3,j} + 4u_{i-2,j} - 5u_{i-1,j} + 2u_{i,j}}{\Delta x^2} + O(\Delta x^2), \quad i = N_x, j = 0, \dots, N_y. \quad (5.13)$$

For the first partial derivative with respect to y :

$$\left. \frac{\partial u}{\partial y} \right|_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} + O(\Delta y^2), \quad i = 0, \dots, N_x, j = 1, \dots, N_y - 1, \quad (5.14)$$

$$\left. \frac{\partial u}{\partial y} \right|_{i,j} = \frac{-3u_{i,j} + 4u_{i,j+1} - u_{i,j+2}}{2\Delta y} + O(\Delta y^2), \quad i = 0, \dots, N_x, j = 0, \quad (5.15)$$

$$\left. \frac{\partial u}{\partial y} \right|_{i,j} = \frac{u_{i,j-2} - 4u_{i,j-1} + 3u_{i,j}}{2\Delta y} + O(\Delta y^2), \quad i = 0, \dots, N_x, j = N_y. \quad (5.16)$$

For the second partial derivative with respect to y :

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} + O(\Delta y^2), \quad i = 0, \dots, N_x, j = 1, \dots, N_y - 1, \quad (5.17)$$

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_{i,j} = \frac{2u_{i,j} - 5u_{i,j+1} + 4u_{i,j+2} - u_{i,j+3}}{\Delta y^2} + O(\Delta y^2), \quad i = 0, \dots, N_x, j = 0, \quad (5.18)$$

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_{i,j} = \frac{-u_{i,j-3} + 4u_{i,j-2} - 5u_{i,j-1} + 2u_{i,j}}{\Delta y^2} + O(\Delta y^2), \quad i = 0, \dots, N_x, j = N_y. \quad (5.19)$$

For simplicity, time dependency was omitted in $u_{i,j}(t)$ and will be omitted for $U(t)$. Taking into account the storage order of matrices, the boundary conditions described before, and dropping the error terms $O(\Delta x^2)$ and $O(\Delta y^2)$ spatial partial can be approximated as presented below.

- The first partial derivative with respect to x using (5.8), (5.9), and (5.10) is:

$$\frac{\partial u}{\partial x} \approx U D_{N_x}, \quad (5.20)$$

with,

$$D_{N_x} = \frac{1}{2\Delta x} \begin{pmatrix} -3 & -1 & 0 & \cdots & 0 & 0 \\ 4 & 0 & \ddots & \ddots & \vdots & \vdots \\ -1 & 1 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & -1 & 1 \\ \vdots & \vdots & \ddots & \ddots & 0 & -4 \\ 0 & 0 & \cdots & 0 & 1 & 3 \end{pmatrix}, \quad (5.21)$$

- The second partial derivative with respect to x using (5.11), (5.12), and (5.13) is:

$$\frac{\partial^2 u}{\partial x^2} \approx U D_{N_x}^{(2)}, \quad (5.22)$$

with,

$$D_{N_x}^{(2)} = \frac{1}{\Delta x^2} \begin{pmatrix} 2 & 1 & 0 & \cdots & \cdots & 0 & 0 \\ -5 & -2 & \ddots & \ddots & \ddots & \vdots & \vdots \\ 4 & 1 & \ddots & \ddots & \ddots & 0 & 0 \\ -1 & 0 & \ddots & \ddots & \ddots & 0 & -1 \\ 0 & 0 & \ddots & \ddots & \ddots & 1 & 4 \\ \vdots & \vdots & \ddots & \ddots & \ddots & -2 & -5 \\ 0 & 0 & \cdots & \cdots & 0 & 1 & 2 \end{pmatrix}. \quad (5.23)$$

- The first partial derivative with respect to y using (5.14), (5.15), and (5.16) is:

$$\frac{\partial u}{\partial y} \approx D_{N_y} U, \quad (5.24)$$

with,

$$D_{N_y} = \frac{1}{2\Delta y} \begin{pmatrix} -3 & 4 & -1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 0 & 1 \\ 0 & \cdots & 0 & 1 & -4 & 3 \end{pmatrix}. \quad (5.25)$$

- The second partial derivative with respect to y using (5.17), (5.18), and (5.19) is:

$$\frac{\partial^2 u}{\partial y^2} \approx D_{N_y}^{(2)} U, \quad (5.26)$$

with,

$$D_{N_y}^{(2)} = \frac{1}{\Delta y^2} \begin{pmatrix} 2 & -5 & 4 & -1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & -1 & 4 & -5 & 2 \end{pmatrix}. \quad (5.27)$$

Defined these expressions, we can approximate Laplacian and gradient of u as follows,

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \approx U D_{N_x}^{(2)} + D_{N_y}^{(2)} U, \quad (5.28)$$

and,

$$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) \approx (U D_{N_x}, D_{N_y} U). \quad (5.29)$$

Using (5.28) and (5.29), the system of *PDEs* defined in (4.1) becomes to the following dynamical system

$$\begin{aligned} \frac{dU(t)}{dt} &= \kappa(U(t) D_{N_x}^{(2)} + D_{N_y}^{(2)} U(t)) - (V_1(t) \odot (U(t) D_{N_x}) + V_2(t) \odot (D_{N_y} U(t))) + F(t), \\ \frac{dB(t)}{dt} &= G(t), \end{aligned} \quad (5.30)$$

where $F(t) = f(U(t), B(t))$, $G(t) = g(U(t), B(t))$ and \odot denotes the *Hadamard product* or element-wise multiplication. Notice that the values of U on Γ ensures that the boundary conditions are satisfied for each t . In particular, the Dirichlet boundary conditions $u(\mathbf{x}, t) = \beta(\mathbf{x}, t) = 0$, $\mathbf{x} \in \Gamma$, are imposed setting the values of $u_{i,j}(t) = \beta_{i,j}(t) = 0$ when $i = 0, N_x, j = 0, \dots, N_y$ and when $i = 0, \dots, N_x, j = 0, N_y$.

The next step is to solve (5.30) by some numerical integration method.

5.3 Time Approximation

This section aims to represent (5.30) as an *Initial Value Problem (IVP)* and solve it using some numerical method. Rewriting (5.30) as

$$\begin{pmatrix} \dot{U}(t) \\ \dot{B}(t) \end{pmatrix} = \begin{pmatrix} \kappa(U(t) D_{N_x}^{(2)} + D_{N_y}^{(2)} U(t)) - (V_1(t) \odot (U(t) D_{N_x}) + V_2(t) \odot (D_{N_y} U(t))) + F(t) \\ G(U(t), B(t)) \end{pmatrix}, \quad (5.31)$$

with $\dot{U}(t) = \frac{dU(t)}{dt}$ and $\dot{B}(t) = \frac{dB(t)}{dt}$, a vector version of (5.31) is presented due to the existence of several libraries that solve *IVPs* but require a vector representation of the

problem. The vectorization operator used by this work is defined in column-major order as follows. Let $A \in \mathbb{R}^{m \times n}$,

$$A = \begin{pmatrix} a_{1,1} & a_{1,1} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ddots & a_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n-1} & a_{m,n} \end{pmatrix}, \quad (5.32)$$

then, the vectorization operator is defined by

$$\text{vec}(A) = (a_{1,1}, a_{2,1}, \dots, a_{m,1}, a_{1,2}, a_{2,2}, \dots, a_{m-1,n}, a_{m,n})^T. \quad (5.33)$$

To restore the original shape of a vectorized matrix $a = \text{vec}(A) \in \mathbb{R}^{mn}$,

$$\text{vec}^{-1}(a) = \begin{pmatrix} a_{1,1} & a_{1,1} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ddots & a_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n-1} & a_{m,n} \end{pmatrix}. \quad (5.34)$$

Let us define $\mathbf{y}(t)$ as,

$$\mathbf{y}(t) = (\text{vec}(U(t)), \text{vec}(B(t)))^T, \quad (5.35)$$

now, (5.31) is redefined by the following *IVP*

$$\dot{\mathbf{y}}(t) = \Phi(t, \mathbf{y}(t)), \quad (5.36)$$

where,

$$\dot{\mathbf{y}}(t) = \left(\text{vec}(\dot{U}(t)), \text{vec}(\dot{B}(t)) \right)^T, \quad (5.37)$$

$$\Phi(t, \mathbf{y}(t)) = \begin{pmatrix} \text{vec} \left(\kappa(U(t)) D_{N_x}^{(2)} + D_{N_y}^{(2)} U(t) \right) - (V_1(t) \odot (U(t) D_{N_x}) + V_2(t) \odot (D_{N_y} U(t))) + F(t) \\ \text{vec}(G(t)) \end{pmatrix}, \quad (5.38)$$

and $\mathbf{y}(0) = (\text{vec}(U(0)), \text{vec}(B(0)))^T$.

There is a wide range of numerical methods to solve (5.36) and it is necessary to specify a discrete version of time. This domain will be defined as,

$$t_n = t_{\min} + n\Delta t, \quad n = 0, 1, \dots, N_t, \quad (5.39)$$

where $\Delta t = (t_{\max} - t_{\min})/N_t$. The numerical methods implemented by this work will be detailed below.

5.3.1 Euler Method

The first method used to solve (5.36) is *Euler Method*, mainly for first experiments and testing the convergence of spatial approximation. Defining $\mathbf{y}_n = \mathbf{y}(t_n)$, the approximation of $\mathbf{y}(t)$ at time t_n , the computation of the *IVP* using this method is

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \Phi(t_n, \mathbf{y}_n). \quad (5.40)$$

This method has a temporal truncation error of $O(\Delta t)$ (Sauer, 2011).

5.3.2 Fourth Order-Runge Kutta Method

This thesis also uses *Fourth Order Runge-Kutta Method (RK4)* to solve (5.36), since its stability region give the flexibility of a Δt choice (Trefethen, 2000). Defining $\mathbf{y}_n = \mathbf{y}(t_n)$, the approximation of $\mathbf{y}(t)$ at time t_n , the computation of the *IVP* using this method is

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (5.41)$$

where,

$$\begin{aligned} \mathbf{k}_1 &= \Phi(t_n, \mathbf{y}_n), \\ \mathbf{k}_2 &= \Phi\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2}\mathbf{k}_1\right), \\ \mathbf{k}_3 &= \Phi\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2}\mathbf{k}_2\right), \\ \mathbf{k}_4 &= \Phi(t_n + \Delta t, \mathbf{y}_n + \Delta t \mathbf{k}_3). \end{aligned} \quad (5.42)$$

This method has a temporal truncation error of $O(\Delta t^4)$ (Sauer, 2011).

5.4 Algorithm Description

Using the approach of *MOL* described in Section 5.1, the algorithm proposed integrates spatial and time approximation presented in Section 5.2 and 5.3. Algorithm 1 details the application of *MOL* to approximate the simplified wildfire mathematical model described in Section 3. The approximation of Φ using expression 5.38 is presented in Algorithm 2. Finally, the application of *RK4* with expression 5.41 and 5.38 is detailed in Algorithm 3.

Algorithm 1 Main algorithm.

Require: $u_0, \beta_0, u_\Gamma, \beta_\Gamma, \mathbf{v}, \kappa, \varepsilon, \alpha, q, u_{pc}, x_{\min}, x_{\max}, y_{\min}, y_{\max}, t_{\min}, t_{\max}, N_x, N_y, N_t$

- 1: $\Delta x \leftarrow (x_{\max} - x_{\min})/N_x$
 - 2: $\Delta y \leftarrow (y_{\max} - y_{\min})/N_y$
 - 3: $\Delta t \leftarrow (t_{\max} - t_{\min})/N_t$
 - 4: $x_i \leftarrow x_{\min} + i \Delta x, \quad i = 0, \dots, N_x$
 - 5: $y_j \leftarrow y_{\min} + j \Delta y, \quad j = 0, \dots, N_y$
 - 6: $t_n \leftarrow t_{\min} + n \Delta t, \quad n = 0, \dots, N_t$
 - 7: $U^0 \leftarrow u_0(x_i, y_j)$
 - 8: $B^0 \leftarrow \beta_0(x_i, y_j)$
 - 9: $\mathbf{y}_0 \leftarrow (\text{vec}(U^0), (B^0))^T$
 - 10: $\mathbf{y}_n \leftarrow \text{RK4}(t_n, \mathbf{y}_0, \Phi)$
 - 11: $U^n, B^n \leftarrow \text{reshape } \mathbf{y}_n$
-

Algorithm 2 Right hand side Φ approximation.

Require: $u_\Gamma, \beta_\Gamma, \mathbf{v}, \kappa, \varepsilon, \alpha, q, u_{pc}, f, g, x_i, y_j, D_{N_x}, D_{N_y}, D_{N_x}^{(2)}, D_{N_y}^{(2)}$

- 1: **procedure** $\Phi(t, \mathbf{y})$
- 2: $\mathbf{u}, \beta \leftarrow \mathbf{y}$
- 3: $V_1, V_2 \leftarrow \mathbf{v}(x_i, y_j, t)$
- 4: $U \leftarrow \text{vec}^{-1}(\mathbf{u})$
- 5: $B \leftarrow \text{vec}^{-1}(\beta)$
- 6: $\dot{U} \leftarrow \kappa \left(U D_{N_x}^{(2)} + D_{N_y}^{(2)} U \right) - (V_1 \odot (U D_{N_x}) + V_2 \odot (D_{N_y} U)) + f(U, B)$
- 7: $\dot{B} \leftarrow g(U, B)$
- 8: Set BC with $u_\Gamma(x_i, y_j)$ and $\beta_\Gamma(x_i, y_j)$.
- 9: $\mathbf{y} \leftarrow \left(\text{vec}(\dot{U}), \text{vec}(\dot{B}) \right)^T$
- 10: **return** \mathbf{y}
- 11: **end procedure**

Algorithm 3 4th Order Runge-Kutta.

- 1: **procedure** RK4(t_n, \mathbf{y}_0, Φ)
- 2: **for** $n = 0$ to $N_t - 1$ **do**
- 3: $\mathbf{k}_1 \leftarrow \Phi(t_n, \mathbf{y}_n)$
- 4: $\mathbf{k}_2 \leftarrow \Phi\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2} \mathbf{k}_1\right)$
- 5: $\mathbf{k}_3 \leftarrow \Phi\left(t_n + \frac{\Delta t}{2}, \mathbf{y}_n + \frac{\Delta t}{2} \mathbf{k}_2\right)$
- 6: $\mathbf{k}_4 \leftarrow \Phi(t_n + \Delta t, \mathbf{y}_n + \Delta t \mathbf{k}_3)$
- 7: $\mathbf{y}_{n+1} \leftarrow \mathbf{y}_n + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$
- 8: **end for**
- 9: **return** \mathbf{y}_n
- 10: **end procedure**

The **reshape** procedure in Algorithm 1 is used to reconstruct the matrices according to the number of nodes N_x , N_y , and N_t . Essentially, *MOL* allows solving the *PDE* model transforming the problem into an *IVP*. This approach gives the flexibility to choose any numerical scheme for time integration and another numerical method for space derivatives. In this particular case, the proposal uses *RK4* for time and *CD FDM* for space. *Ngen-Kütral* framework also includes the *Euler Method* for time and *Fast Fourier Transform* for space (San Martín & Torres, 2019).

The computational complexity of the algorithm can be divided into two parts. First, the number of elemental operations for the approximation of Φ using Algorithm 2. Assuming $N = N_x = N_y$, the number of operations is:

- 4 matrix-matrix product, $\sim 8 N \times N^2$ operations,
- 2 element-wise multiplication of N^2 elements,
- 4 sums of N^2 elements,
- 2 functions evaluations for f and g of N^2 elements,

therefore, the total number of elemental operations of Φ approximation is $\sim 8 N^3 + 6 N^2 + C_f N^2 + C_g N^2 = O(N^3)$, where C_f and C_g are the number of elemental operations to compute f and g , respectively. This is an upper bound estimation and it does not take advantage of the sparsity pattern of the differentiation matrices involved. Subsection 7.1.3 will show the experimental complexity achieved by the algorithm. In the second part, the number of elemental operations of Algorithm 3, depends on time steps N_t and the computational cost of evaluating Φ , C_Φ , giving a total of $\sim 4 N_t C_\Phi$. Finally, the computational complexity of the entire Algorithm is $\sim 4 N_t (8 N^3 + 6 N^2 + C_f N^2 + C_g N^2)$. Depending on N_t , the complexity could be:

- $O(N^4)$ if $N_t \sim N_x = N_y$,
- $O(N^3)$ if $N_t \ll N_x = N_y$,
- $O(N)$ if $N_t \gg N_x = N_y$.

5.5 Stability Analysis

To ensure the operation of the algorithm, a stability analysis to solve the problem $\dot{\mathbf{y}}(t) = \Phi(t, \mathbf{y}(t))$ is performed. Expanding the expression 5.31,

$$\Phi(t, \mathbf{y}) = \left(\dot{U}_{0,0}, \dots, \dot{U}_{i,j}, \dots, \dot{U}_{N_x, N_y}, \dots, \dot{B}_{0,0}, \dots, \dot{B}_{i,j}, \dots, \dot{B}_{N_x, N_y} \right)^T, \quad (5.43)$$

where,

$$\begin{aligned}\dot{U}_{i,j} = & \kappa \left(\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta x^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta y^2} \right) \\ & - \left(V_{1i,j} \left(\frac{U_{i+1,j} - U_{i-1,j}}{2\Delta x} \right) + V_{2i,j} \left(\frac{U_{i,j+1} - U_{i,j-1}}{2\Delta y} \right) \right) \\ & + H_{pc}(U_{i,j})B_{i,j} \exp \left(\frac{U_{i,j}}{1 + \varepsilon U_{i,j}} \right) - \alpha U_{i,j},\end{aligned}$$

and,

$$\dot{B}_{i,j} = -H(U_{i,j}) \frac{\varepsilon}{q} B_{i,j} \exp \left(\frac{U_{i,j}}{1 + \varepsilon U_{i,j}} \right).$$

The stability analysis requires the Jacobian Matrix of (5.43), that is, the computation of the derivative of each vector element with respect to $U_{i,j}$ and $B_{i,j}$, $i = 0, \dots, N_x, j = 0, \dots, N_y$ as follows,

$$\mathbf{J}_\Phi = \begin{pmatrix} \frac{\partial \dot{U}_{0,0}}{\partial U_{0,0}} & \cdots & \frac{\partial \dot{U}_{0,0}}{\partial U_{N_x, N_y}} & \frac{\partial \dot{U}_{0,0}}{\partial B_{0,0}} & \cdots & \frac{\partial \dot{U}_{0,0}}{\partial B_{N_x, N_y}} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial \dot{U}_{i,j}}{\partial U_{0,0}} & \cdots & \frac{\partial \dot{U}_{i,j}}{\partial U_{N_x, N_y}} & \frac{\partial \dot{U}_{i,j}}{\partial B_{0,0}} & \cdots & \frac{\partial \dot{U}_{i,j}}{\partial B_{N_x, N_y}} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial \dot{U}_{N_x, N_y}}{\partial U_{0,0}} & \cdots & \frac{\partial \dot{U}_{N_x, N_y}}{\partial U_{N_x, N_y}} & \frac{\partial \dot{U}_{N_x, N_y}}{\partial B_{0,0}} & \cdots & \frac{\partial \dot{U}_{N_x, N_y}}{\partial B_{N_x, N_y}} \\ \frac{\partial \dot{B}_{0,0}}{\partial U_{0,0}} & \cdots & \frac{\partial \dot{B}_{0,0}}{\partial U_{N_x, N_y}} & \frac{\partial \dot{B}_{0,0}}{\partial B_{0,0}} & \cdots & \frac{\partial \dot{B}_{0,0}}{\partial B_{N_x, N_y}} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial \dot{B}_{i,j}}{\partial U_{0,0}} & \cdots & \frac{\partial \dot{B}_{i,j}}{\partial U_{N_x, N_y}} & \frac{\partial \dot{B}_{i,j}}{\partial B_{0,0}} & \cdots & \frac{\partial \dot{B}_{i,j}}{\partial B_{N_x, N_y}} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial \dot{B}_{N_x, N_y}}{\partial U_{0,0}} & \cdots & \frac{\partial \dot{B}_{N_x, N_y}}{\partial U_{N_x, N_y}} & \frac{\partial \dot{B}_{N_x, N_y}}{\partial B_{0,0}} & \cdots & \frac{\partial \dot{B}_{N_x, N_y}}{\partial B_{N_x, N_y}} \end{pmatrix}. \quad (5.44)$$

Then, the elements of (5.44) are:

$$\frac{\partial \dot{U}_{i,j}}{\partial U_{i,j}} = -2\kappa \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) + B_{i,j} \exp \left(\frac{U_{i,j}}{1 + \varepsilon U_{i,j}} \right) \left(\frac{\partial H_{pc}(U_{i,j})}{\partial U_{i,j}} + \frac{H_{pc}(U_{i,j})}{(1 + \varepsilon U_{i,j})^2} \right) - \alpha, \quad (5.45)$$

$$\frac{\partial \dot{U}_{i,j}}{\partial U_{i+1,j}} = \frac{\kappa}{\Delta x^2} - \frac{V_{1i,j}}{2\Delta x} \quad (5.46)$$

$$\frac{\partial \dot{U}_{i,j}}{\partial U_{i-1,j}} = \frac{\kappa}{\Delta x^2} + \frac{V_{1i,j}}{2\Delta x} \quad (5.47)$$

$$\frac{\partial \dot{U}_{i,j}}{\partial U_{i,j+1}} = \frac{\kappa}{\Delta y^2} - \frac{V_{2i,j}}{2\Delta y} \quad (5.48)$$

$$\frac{\partial \dot{U}_{i,j}}{\partial U_{i,j-1}} = \frac{\kappa}{\Delta y^2} + \frac{V_{2i,j}}{2\Delta y} \quad (5.49)$$

$$\frac{\partial \dot{U}_{i,j}}{\partial B_{i,j}} = H_{pc}(U_{i,j}) \exp\left(\frac{U_{i,j}}{1 + \varepsilon U_{i,j}}\right) \quad (5.50)$$

$$\frac{\partial \dot{B}_{i,j}}{\partial U_{i,j}} = -\frac{\varepsilon}{q} B_{i,j} \exp\left(\frac{U_{i,j}}{1 + \varepsilon U_{i,j}}\right) \left(\frac{\partial H_{pc}(U_{i,j})}{\partial U_{i,j}} + \frac{H_{pc}(U_{i,j})}{(1 + \varepsilon U_{i,j})^2}\right) \quad (5.51)$$

$$\frac{\partial \dot{B}_{i,j}}{\partial B_{i,j}} = -H_{pc}(U_{i,j}) \frac{\varepsilon}{q} \exp\left(\frac{U_{i,j}}{1 + \varepsilon U_{i,j}}\right) \quad (5.52)$$

Notice that $H_{pc}(U_{i,j})$ is not differentiable at u_{pc} but $\frac{\partial H_{pc}(U_{i,j})}{\partial U_{i,j}} = 0$ for $U_{i,j} \neq u_{pc}$. Finally, \mathbf{J}_Φ is defined by

$$\mathbf{J}_\Phi = \begin{bmatrix} \mathbf{J}_A & \mathbf{J}_B \\ \mathbf{J}_C & \mathbf{J}_D \end{bmatrix} \in \mathbb{R}^{2(N_x+1)(N_y+1) \times 2(N_x+1)(N_y+1)}, \quad (5.53)$$

where,

$$\mathbf{J}_A = \begin{bmatrix} \begin{matrix} \ddots & \ddots & \ddots \\ \cdots & 0 & \end{matrix} \quad (5.47) & \underbrace{\begin{matrix} \ddots & \ddots & \ddots \\ 0 & \cdots & 0 \end{matrix}}_{N_y \text{ times}} \quad (5.49) & \begin{matrix} \ddots & \ddots & \ddots \\ (5.45) & (5.48) & \end{matrix} & \underbrace{\begin{matrix} \ddots & \ddots & \ddots \\ 0 & \cdots & 0 \end{matrix}}_{N_y \text{ times}} \quad (5.46) & \begin{matrix} \ddots & \ddots & \ddots \\ 0 & \cdots & \end{matrix} \end{bmatrix}, \quad (5.54)$$

$$\mathbf{J}_B = \begin{bmatrix} (5.50) & 0 \\ & \ddots \\ 0 & (5.50) \end{bmatrix}, \quad (5.55)$$

$$\mathbf{J}_C = \begin{bmatrix} (5.51) & 0 \\ & \ddots \\ 0 & (5.51) \end{bmatrix}, \quad (5.56)$$

$$\mathbf{J}_D = \begin{bmatrix} (5.52) & 0 \\ & \ddots \\ 0 & (5.52) \end{bmatrix}. \quad (5.57)$$

Now, the stability analysis requires the computation of the eigenvalues λ of $\mathbf{J}_\Phi(t_n, \mathbf{y}_n)$ for $n = 0, \dots, N_t$, and they must be inside the stability region of the *IVP* solver. Let $\bar{z} = \lambda \Delta t \in \mathbb{C}$, the *Euler method* is stable if $|1 + \bar{z}| \leq 1$ and *RK4* is stable if $|1 + \bar{z} + \frac{1}{2}\bar{z}^2 + \frac{1}{6}\bar{z}^3 + \frac{1}{24}\bar{z}^4| \leq 1$. This implies that the algorithm stability depends on Δt selection and the Subsection 7.1.1 presents results about experiments of this analysis. Notice that the stability region of *RK4* is larger than the *Euler Method*, allowing the use of larger values of Δt .

5.6 Risk Maps Generation Algorithm

The *risk map* generation requires the computation of independent scenarios or numerical simulations in the context of *IVPs*. Let N_s the number of numerical simulations needed to generate a *risk map*, and let $\mathbf{y}(t)^{[m]} = (\text{vec}(U(t)^{[m]}), \text{vec}(B(t)^{[m]}))^T$, the vector associated with an independent scenario $m \in [0, N_s - 1]$. The algorithm strategy is to define a new *IVP* containing the N_s independent scenarios. To address this new problem, let $\mathbf{Y}(t) \in \mathbb{R}^{2N_s(N_x+1)(N_y+1)}$ the vector containing all approximations at time t , defined by

$$\begin{aligned} \mathbf{Y}(t) &= ((\text{vec}(U(t)^{[0]}), \text{vec}(B(t)^{[0]})), \dots, (\text{vec}(U(t)^{[N_s-1]}), \text{vec}(B(t)^{[N_s-1]})))^T \\ &= (\mathbf{y}(t)^{[0]}, \dots, \mathbf{y}(t)^{[N_s-1]})^T. \end{aligned} \quad (5.58)$$

The new *IVP* using (5.58) is

$$\frac{d\mathbf{Y}(t)}{dt} = \dot{\mathbf{Y}}(t) = (\Phi(t, \mathbf{y}(t)^{[0]}), \dots, \Phi(t, \mathbf{y}(t)^{[N_s-1]}))^T = \Phi(t, \mathbf{Y}(t)), \quad (5.59)$$

with $\mathbf{Y}(0) = (\mathbf{y}(0)^{[0]}, \mathbf{y}(0)^{[1]}, \dots, \mathbf{y}(0)^{[N_s-1]})^T$. To solve (5.59) numerically, the same time integration methods presented in section 5.3 can be used, defining a new approximation $\mathbf{Y}_n = \mathbf{Y}(t_n)$. Now, the number of elemental operations of (5.59) using *RK4* is $\sim 4N_s N_t (8N^3 + 6N^2 + C_f N^2 + C_g N^2)$.

Chapter 6

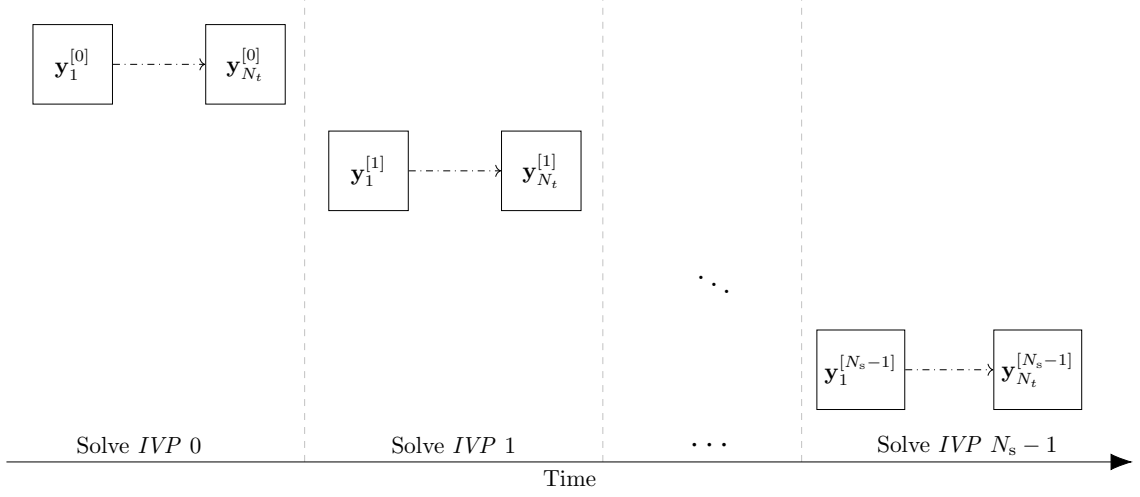
Numerical Implementation

The following chapter describes the *CPU* and *GPU* implementations of the algorithm derived in Chapter 5.

6.1 *CPU* Version

A prototype implementation of the algorithm was developed in *Python*, using *NumPy* for numerical calculations and vector data structures manipulation, and *SciPy* for sparse matrices. It is important to note that these libraries include an optimized implementation of linear algebra algorithms and are mostly developed in *C/C++* (Oliphant, 2007). This version directly implements Algorithms 1, 2, and 3. The source code of this version is publicly available in GitHub (San Martín & Torres, 2018a). For spatial derivatives, the matrix multiplication was performed using `numpy.dot` with the matrices detailed in Algorithm 2. Exploiting the sparse structure of the differentiation matrices, matrix multiplication using the `scipy.sparse.csr_matrix.dot` method is also included. To solve the *IVP* problem, the *Euler Method* and *RK4* method were implemented from scratch. Implementations of other integration methods that are part of the `scipy.integrate.solve_ivp` library are also included. The algorithm to generate a *risk map* using *Python* implementation is summarized in Figure 6.1.

A sequential version in *C* and another with multi-thread handling using *OpenMP* is also developed for performance comparison. Both implementations solve the problem (5.59) detailed in section 5.6. To avoid the construction of differentiation matrices with the sparse pattern, differences schemes of (5.8), (5.14), (5.11), and (5.17) were applied. Algorithm 4 details the approximation of $\Phi(t, \mathbf{Y})$ using *C*. This implementation only uses a nested loop to process each scenario m . On the other hand, *OpenMP* uses the `#pragma omp for` directive in line 4 of Algorithm 4 to handle the N_s independent numerical simulations. The multi-threading implementation is just a naive version made for parallel approach reference. Defining N_T as the number of *CPU threads*, usually $N_T \ll N_s$ when $N_s \gg 1$, implies that *OpenMP* will process groups of N_T scenarios in parallel until N_s are reached. A sketch of the *C* implemen-


 Figure 6.1: *Python* implementation of risk maps generation algorithm.

Algorithm 4 Φ approximation using C implementation.

Require: $u_\Gamma, \beta_\Gamma, \mathbf{v}, \kappa, \varepsilon, \alpha, q, u_{pc}, f, g, x_i, y_j, D_{N_x}, D_{N_y}, D_{N_x}^{(2)}, D_{N_y}^{(2)}, N_s$

```

1: procedure  $\Phi(t, \mathbf{Y})$ 
2:   for  $m = 0$  to  $N_s - 1$  do
3:      $\mathbf{u}, \beta \leftarrow \mathbf{y}^{[m]}$ 
4:      $V_1, V_2 \leftarrow \mathbf{v}(x_i, y_j, t)$ 
5:      $U \leftarrow \text{vec}^{-1}(\mathbf{u})$ 
6:      $B \leftarrow \text{vec}^{-1}(\beta)$ 
7:      $\dot{U} \leftarrow \kappa \left( U D_{N_x}^{(2)} + D_{N_y}^{(2)} U \right) - (V_1 \odot (U D_{N_x}) + V_2 \odot (D_{N_y} U)) + f(U, B)$ 
8:      $\dot{B} \leftarrow g(U, B)$ 
9:     Set BC with  $u_\Gamma(x_i, y_j)$  and  $\beta_\Gamma(x_i, y_j)$ .
10:     $\mathbf{y}^{[m]} \leftarrow \left( \text{vec}(\dot{U}), \text{vec}(\dot{B}) \right)^T$ 
11:  end for
12:  return  $\mathbf{Y}$ 
13: end procedure
    
```

tation is shown in Figure 6.2 and Figure 6.3 presents the description for *OpenMP* implementation.

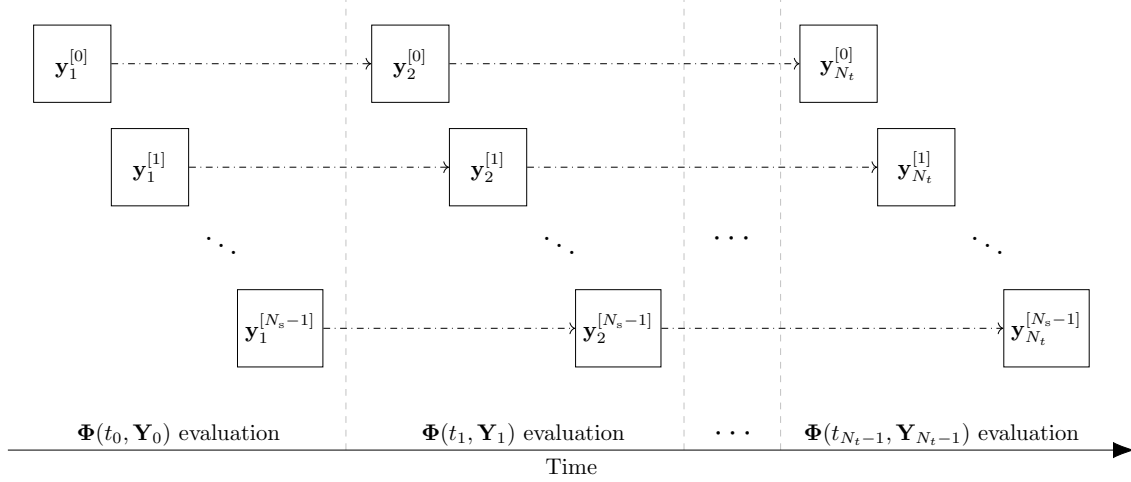


Figure 6.2: *C* implementation of risk maps generation algorithm.

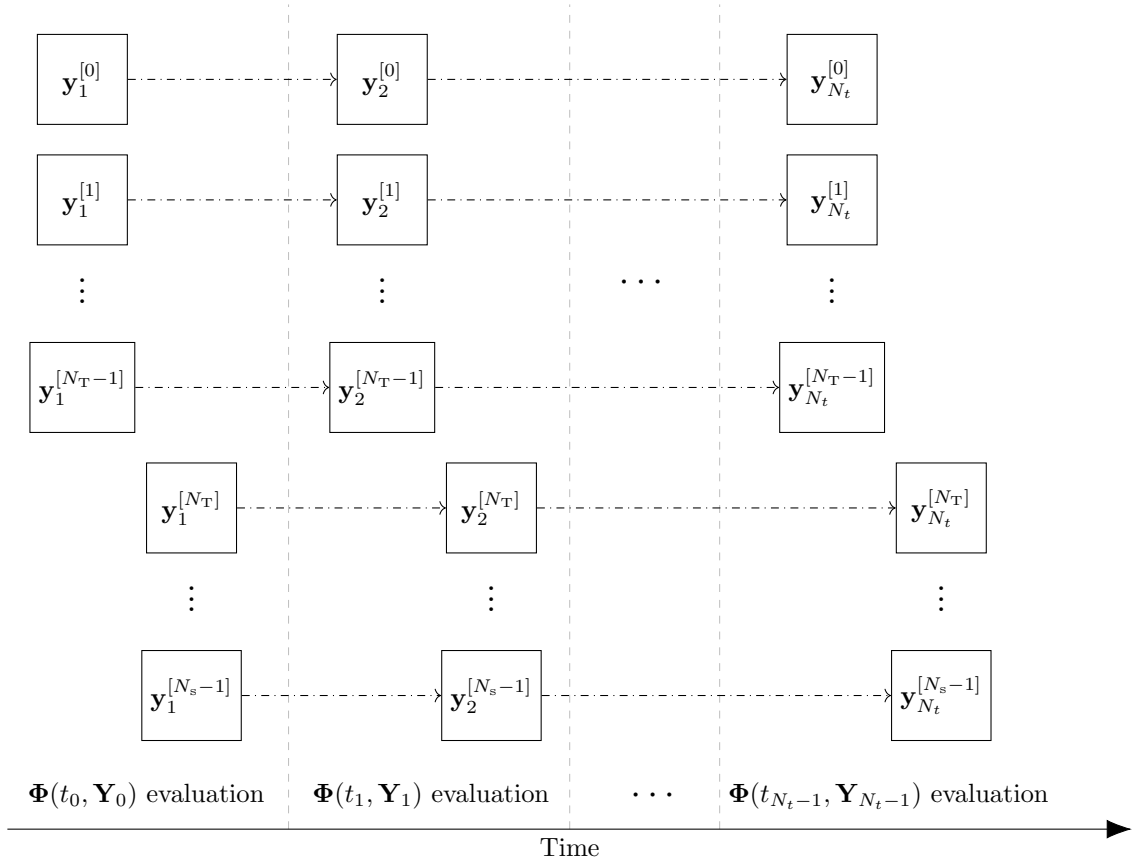
6.2 GPU Version

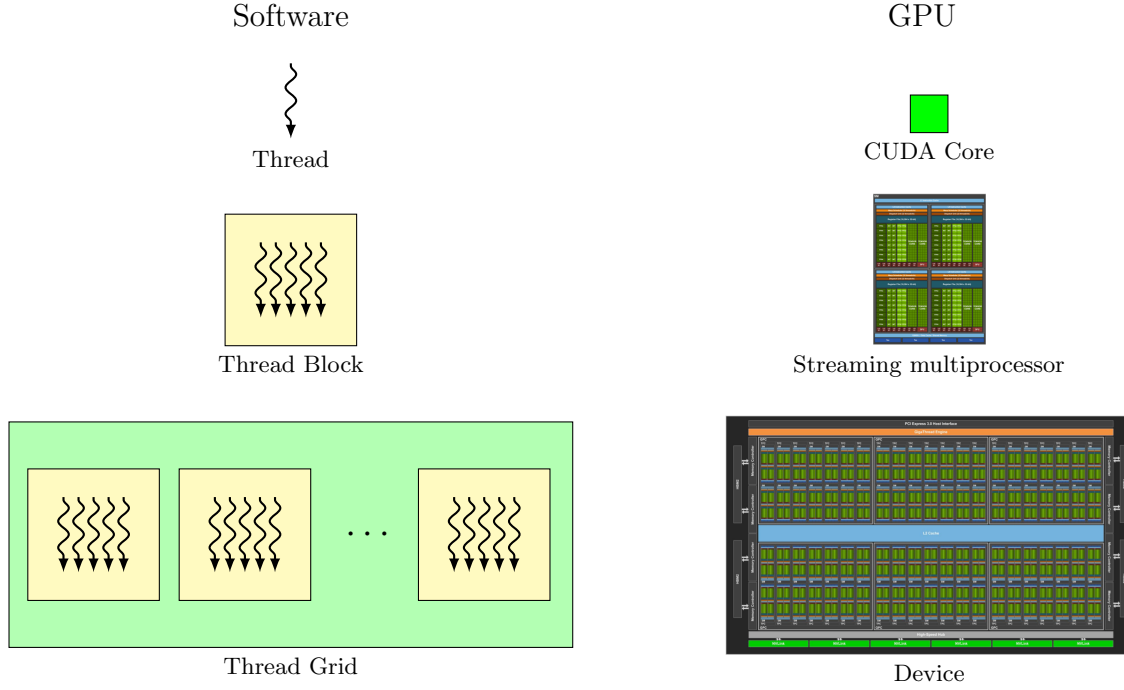
6.2.1 CUDA description

Graphics Processing Unit (GPU), included in graphic cards, has been widely using over the last few years due to the large amount of computation that can be processed, including numerical simulations of *PDEs* (Mattes & Kofuji, 2014). Although the graphic cards were built primarily for work on graphics, *General Purpose GPU (GPGPU)* concept has been increasingly used in scientific applications because of the computational advantages over *CPUs* (NVIDIA, 2018).

The *GPU* parallel programming model allows to make a simple instruction into multiple threads and it is known as *Single Instruction Multiple Threads (SIMT)*. *CUDA* is one of the best-known platforms for *GPU* software development, providing several directives to work with *C/C++* programming language, but extends its use to other languages. *GPU* code execution is performed on a function called *Kernels*, which are executed in parallel on the available threads according to the features of the graphics card. *CUDA* defines *Host* the resources of *CPU* and its memory, and *Device* refers to the *GPU* and its memory. Figure 6.4 presents a sketch of the software-hardware mapping. General parameters in *kernel* execution are the number of blocks per grid and the number of threads per block. The *GPU* code performance depends, in many cases, on the right selection of these parameters. *CUDA* built-in variables used by *Kernels* to access information about threads, blocks, and grids

- `blockIdx.x`, `blockIdx.y`, `blockIdx.z` return the block ID in the x-axis, y-axis, and z-axis of the block that is executing the given block of code.


 Figure 6.3: *OpenMP* implementation of risk maps generation algorithm.


 Figure 6.4: *CUDA* software mapping.

- `threadIdx.x`, `threadIdx.y`, `threadIdx.z` return the thread ID in the x-axis, y-axis, and z-axis of the thread that is being executed by the *GPU* in this particular block.
- `blockDim.x`, `blockDim.y`, `blockDim.z` return the number of blocks per grid.

CUDA threads access data from different memory areas while their execution. Each thread has private local memory. Each thread block has a shared memory space visible to all threads of the block and with the same lifetime as the block. All threads may access the global memory (NVIDIA, 2021). *CUDA* also provides two additional read-only memory spaces accessible by all threads: the constant and texture memory spaces. The global, constant, and texture memory spaces are optimized for diverse memory usages. Additionally, texture memory offers different addressing modes, as well as data filtering, for some specific data formats. The global, constant, and texture memory spaces are persistent across kernel launches by the same application (NVIDIA, 2021).

6.2.2 *GPU* algorithm

Since the main application of this work is to perform many numerical simulations as shown in Figure 4.1 and detailed in section 5.6, the *GPU* approach is suitable in this context. Each scenario $m \in [0, N_s - 1]$, N_s the number of scenarios, is associated with an independent numerical simulation. The idea is to process $2 N_s (N_x + 1)(N_y + 1)$

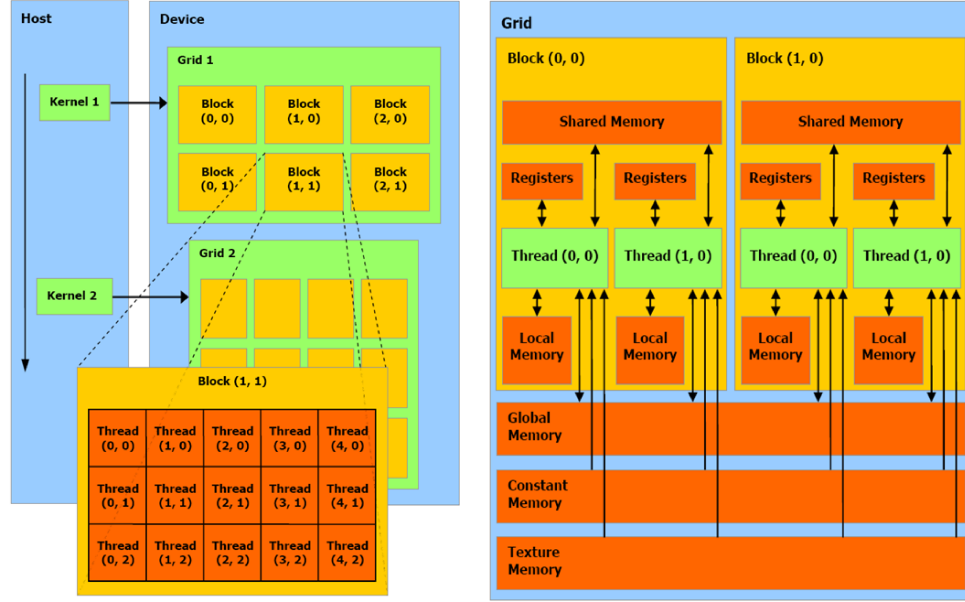


Figure 6.5: Programming tools provided by *CUDA* platform (Nobile et al., 2014).

elements in parallel for each time step of the numerical integration method. To summarize, the algorithm solves N_s IVPs, equation 5.59, in parallel using the *GPU*. Figure 6.6 shows a sketch of the procedure. The *GPU* approach proposed uses each thread to process the element $\mathbf{Y}_{n,l}$, $l \in [0, 2N_s(N_x+1)(N_y+1)-1]$, at discrete time t_n of the scheme described in Figure 6.7. This implementation uses a one-dimensional approach to manipulate threads and for memory storage. Figure 6.8 shows the single dimensional array of *CUDA* threads.

With this approach, the *CUDA* built-in variables used are `blockIdx.x`, `threadIdx.x`, and `blockDim.x`. Each *thread id*, defined by `tid = threadIdx.x + blockIdx.x * blockDim.x`, maps to element $\mathbf{Y}_{n,l}$ in the approximation of $\Phi(t, \mathbf{Y})$ evaluation. The vector \mathbf{Y}_n is stored in *global memory* due to the size that it can reach. As indicated in 6.1, the difference schemes were used directly to avoid the storage of differentiation matrices. Since the spatial domain of the problem is fixed, the implementation uses *constant memory* to store the information of x and y . The principal code section of the implementation is presented in Algorithm 5 and it corresponds to the loop executed in the *RK4* method by the host. Routines used by the method are included in Algorithms 6 and 7. Note that the computational effort is the evaluation of $\Phi(t, \mathbf{Y})$ detailed in Algorithm 8. Functions parameter correspond to those described in equation (4.1) and *N_BLOCKS*, *N_THREADS* are the *CUDA Kernel* parameters. In particular, *RHSVec Kernel* consumes approximately 90% of the implementation execution time.

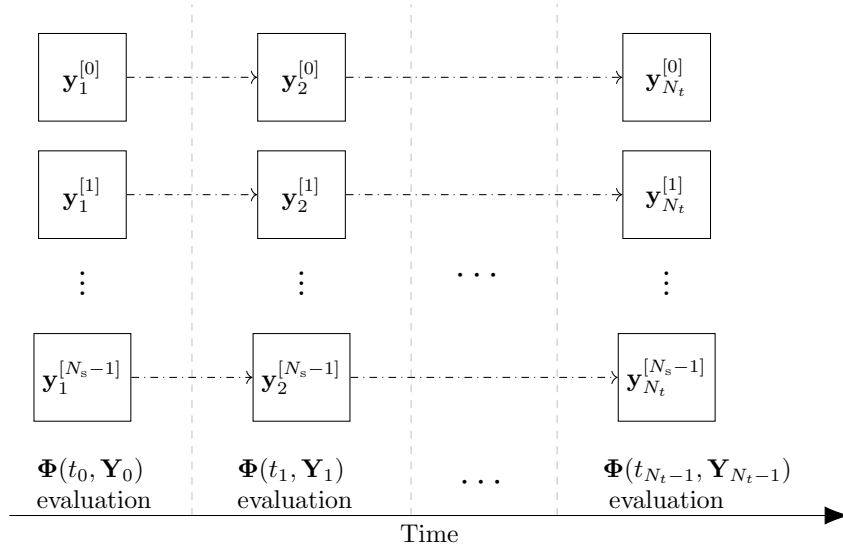
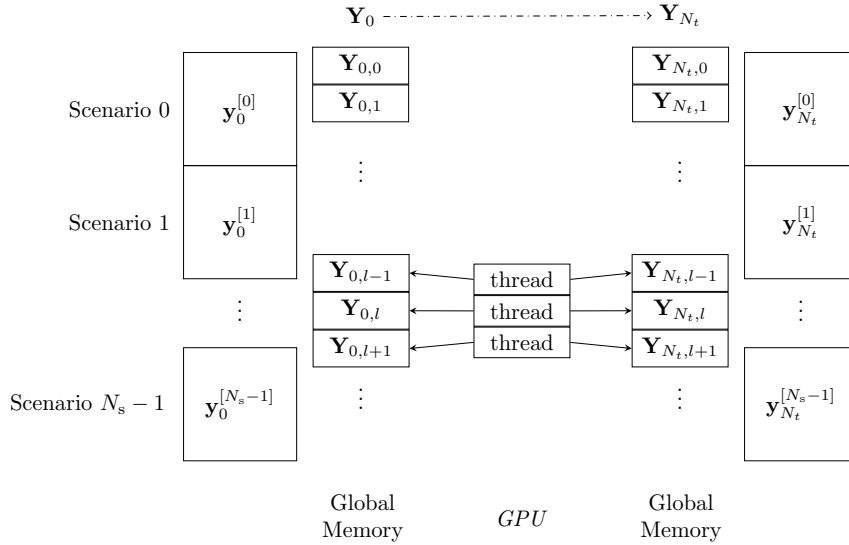
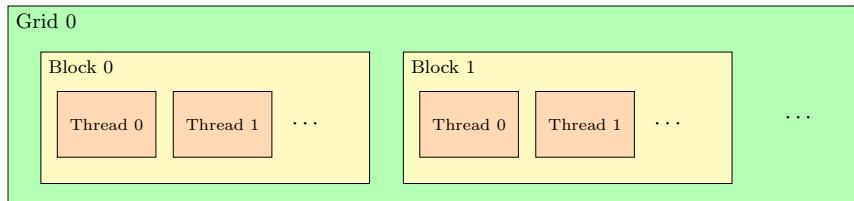

 Figure 6.6: *CUDA* implementation of risk maps generation algorithm.


Figure 6.7: Description of the GPU implementation.


 Figure 6.8: Single dimensional array of *CUDA* threads.

Algorithm 5 *RK4* host implementation.

```

1: ...
2: for  $n = 1$  to  $N_t$  do
3:   Copy  $\mathbf{Y}$  into  $\mathbf{Y}_{tmp}$ 
4:   RHSVec<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{k}_1$ ,  $\mathbf{Y}_{tmp}$ )
5:   sumVector<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{k}_{tmp}$ ,  $\mathbf{Y}_{tmp}$ ,  $\mathbf{k}_1$ ,  $0.5\Delta t$ , size)
6:   RHSVec<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{k}_2$ ,  $\mathbf{Y}_{tmp}$ )
7:   sumVector<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{k}_{tmp}$ ,  $\mathbf{Y}_{tmp}$ ,  $\mathbf{k}_2$ ,  $0.5\Delta t$ , size)
8:   RHSVec<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{k}_3$ ,  $\mathbf{Y}_{tmp}$ )
9:   sumVector<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{k}_{tmp}$ ,  $\mathbf{Y}_{tmp}$ ,  $\mathbf{k}_3$ ,  $\Delta t$ , size)
10:  RHSVec<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{k}_4$ ,  $\mathbf{Y}_{tmp}$ )
11:  RK4Scheme<<<N_BLOCKS, N_THREADS>>>(Parameters,  $\mathbf{Y}$ ,  $\mathbf{Y}_{tmp}$ ,  $\mathbf{k}_1$ ,  $\mathbf{k}_2$ ,  $\mathbf{k}_3$ ,  $\mathbf{k}_4$ ,  $\Delta t$ , size)
12: end for
13: ...

```

Algorithm 6 *CUDA* implementation of vector a plus scalar times vector c .

```

1: procedure SUMVECTOR(Parameters,  $c$ ,  $a$ ,  $b$ , size, scalar)
2:    $t_{id} \leftarrow \text{threadIdx.x} + \text{blockIdx.x} \cdot \text{blockDim.x}$ 
3:   while  $t_{id} < \text{size}$  do
4:      $c_{t_{id}} \leftarrow a_{t_{id}} + \text{scalar} \cdot b_{t_{id}}$ 
5:      $t_{id} \leftarrow t_{id} + \text{gridDim.x} \cdot \text{blockDim.x}$ 
6:   end while
7: end procedure

```

Algorithm 7 *RK4* scheme implementation in *CUDA*.

```

1: procedure RK4SCHEME(Parameters,  $\mathbf{Y}_{new}$ ,  $\mathbf{Y}_{old}$ ,  $\mathbf{k}_1$ ,  $\mathbf{k}_2$ ,  $\mathbf{k}_3$ ,  $\mathbf{k}_4$ ,  $\Delta t$ , size)
2:    $t_{id} \leftarrow \text{threadIdx.x} + \text{blockIdx.x} \cdot \text{blockDim.x}$ 
3:   while  $t_{id} < \text{size}$  do
4:      $\mathbf{Y}_{new t_{id}} \leftarrow \mathbf{Y}_{old t_{id}} + \frac{\Delta t}{6} (\mathbf{k}_{1 t_{id}} + \mathbf{k}_{2 t_{id}} + \mathbf{k}_{3 t_{id}} + \mathbf{k}_{4 t_{id}})$ 
5:      $t_{id} \leftarrow t_{id} + \text{gridDim.x} \cdot \text{blockDim.x}$ 
6:   end while
7: end procedure

```

Algorithm 8 *CUDA* implementation of $\Phi(\mathbf{Y})$.

```

1: procedure RHSVEC(Parameters,  $\mathbf{Y}$ ,  $\mathbf{Y}_{\text{tmp}}$ )
2:    $t_{\text{id}} \leftarrow \text{threadIdx.x} + \text{blockIdx.x} \cdot \text{blockDim.x}$ 
3:   while  $t_{\text{id}} < N_s (N_x + 1)(N_y + 1)$  do
4:      $m \leftarrow t_{\text{id}} / ((N_x + 1)(N_y + 1))$ 
5:      $i \leftarrow (t_{\text{id}} - m (N_x + 1)(N_y + 1)) \% (N_y + 1)$ 
6:      $j \leftarrow (t_{\text{id}} - m (N_x + 1)(N_y + 1)) / (N_y + 1)$ 
7:      $\text{offset} \leftarrow 2m (N_x + 1)(N_y + 1)$ 
8:      $l \leftarrow \text{offset} + j (N_y + 1) + i$ 
9:      $u_{\text{idx}} \leftarrow l$ 
10:     $b_{\text{idx}} \leftarrow l + (N_x + 1)(N_y + 1)$ 
11:     $u_{\text{k}} \leftarrow 0$ 
12:     $b_{\text{k}} \leftarrow 0$ 
13:    if  $\neg(i = 0 \vee i = N_y \vee j = 0 \vee j = N_x)$  then
14:       $u \leftarrow \mathbf{Y}_{\text{tmp}}[u_{\text{idx}}]$ 
15:       $b \leftarrow \mathbf{Y}_{\text{tmp}}[b_{\text{idx}}]$ 
16:       $v_1, v_2 \leftarrow \mathbf{v}(x, y, t)$ 
17:       $u_{\text{r}} \leftarrow \mathbf{Y}_{\text{tmp}}[\text{offset} + (j+1) \cdot (N_y+1) + i]$ 
18:       $u_{\text{l}} \leftarrow \mathbf{Y}_{\text{tmp}}[\text{offset} + (j-1) \cdot (N_y+1) + i]$ 
19:       $u_{\text{u}} \leftarrow \mathbf{Y}_{\text{tmp}}[\text{offset} + j \cdot (N_y+1) + i + 1]$ 
20:       $u_{\text{d}} \leftarrow \mathbf{Y}_{\text{tmp}}[\text{offset} + j \cdot (N_y+1) + i - 1]$ 
21:       $u_{\text{x}} \leftarrow (u_{\text{r}} - u_{\text{l}}) / (2\Delta x)$ 
22:       $u_{\text{y}} \leftarrow (u_{\text{u}} - u_{\text{d}}) / (2\Delta y)$ 
23:       $u_{\text{xx}} \leftarrow (u_{\text{r}} - 2u + u_{\text{l}}) / \Delta x^2$ 
24:       $u_{\text{yy}} \leftarrow (u_{\text{u}} - 2u + u_{\text{d}}) / \Delta y^2$ 
25:       $u_{\text{k}} \leftarrow \kappa (u_{\text{xx}} + u_{\text{yy}}) - (v_1 u_{\text{x}} + v_2 u_{\text{y}}) + H_{\text{pc}}(u)b \exp(u/(1 + \varepsilon u)) - \alpha u$ 
26:       $b_{\text{k}} \leftarrow -H_{\text{pc}}(u)\varepsilon b \exp(u/(1 + \varepsilon u))/q$ 
27:    end if
28:     $\mathbf{Y}_{u_{\text{idx}}} \leftarrow u_{\text{k}}$ 
29:     $\mathbf{Y}_{b_{\text{idx}}} \leftarrow b_{\text{k}}$ 
30:     $t_{\text{id}} \leftarrow t_{\text{id}} + \text{gridDim.x} \cdot \text{blockDim.x}$ 
31:  end while
32: end procedure

```

Chapter 7

Numerical Results

The following chapter presents the capabilities of the proposed algorithm. Section 7.1 shows an analysis of the algorithm in terms of stability, convergence, and computational complexity. Section 7.2 presents examples of numerical simulations performed by the proposal. Section 7.3 shows the main application of this thesis, the generation of risk maps. Finally, section 7.4 presents the comparison of *CPU* and *GPU* implementations.

The values of model parameters reported in this section were computed according to nondimensional formulas detailed in subsection 3.2.2 using the physical parameters presented in Table 4.1.

7.1 Algorithm Analysis

To evaluate the properties of the algorithm as a valid tool for approximation, this section presents a stability analysis, the convergence of numerical methods used for space and time approximations. Besides, it is included a computational complexity analysis of the algorithm to ensure that the numerical simulations can be obtained in reasonable computation times. All the analyses elaborated in this section were developed with *Python CPU* implementation.

The following parameters were used for algorithm analysis:

- Physical parameters:
 - A large enough domain to avoid boundary conditions effect, that is, $\Omega \times [t_{\min}, t_{\max}] = [0, 90]^2 \times [0, 0.1]$. In physical domain it is equivalent to a square of 26 [m] \times 26 [m] and 14 [min] of simulation.
 - Temperature and fuel were set to zero at boundary, $u_{\Gamma}(\mathbf{x}, t) = \beta_{\Gamma}(\mathbf{x}, t) = 0$, on $\Gamma \times [0, 0.1]$.
 - Initial condition for temperature is $u_0(\mathbf{x}) = 6 \exp(-0.05((x - 20)^2 + (y - 20)^2))$ and for fuel $\beta_0(\mathbf{x})$ was initialized with an uniformly distributed random variable on $[0, 1]^2$.

- The vector field used was $\mathbf{v}(\mathbf{x}, t) = (\cos(\pi/4), \sin(\pi/4))$.
- Model parameters $\kappa = 0.1$, $\varepsilon = 0.3$, $u_{pc} = 3$, $q = 1$, $\alpha = 0.001$.
- Numerical parameters:
 - Number of space nodes for *FDM*, $N_x = N_y = \{255, 511, 1023, 2047\}$.
 - Number of time nodes for *RK4*, $N_t = \{64, 128, 256, 512\}$.

7.1.1 Stability

Stability analysis is performed using the same parameters as before, but $N_x = N_y = 63$, $t_{\max} = 25$, $N_t = 500$ for *Euler Method*, and $N_t = 100$ for *RK4*. Figure 7.1 presents the stability region of the *Euler Method* and eigenvalues of \mathbf{J}_Φ times Δt for the experiment performed.

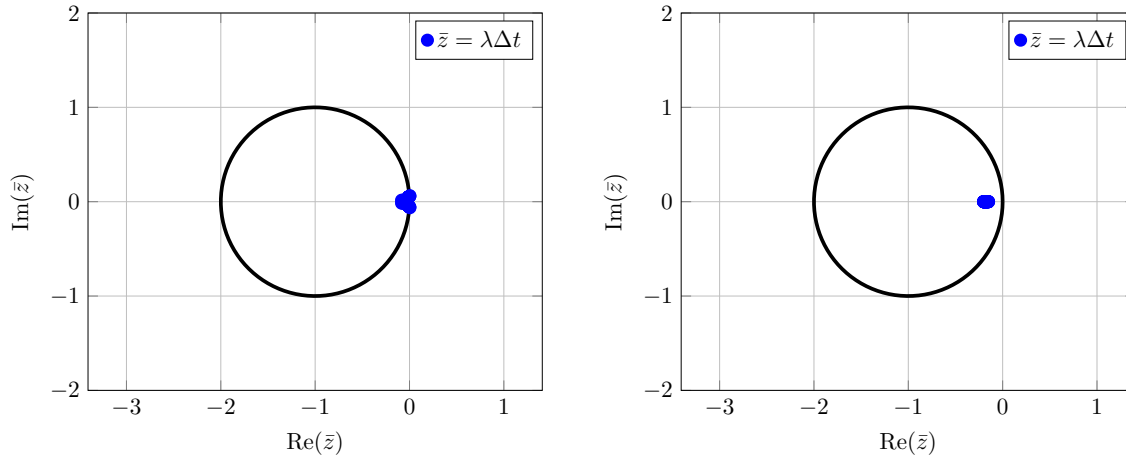


Figure 7.1: Eigenvalues of \mathbf{J}_Φ for first (left plot) and last (right plot) time step using the *Euler Method*.

Analysis of experiment for *RK4* is presented in Figure 7.2. It contains the stability region of the method and eigenvalues of \mathbf{J}_Φ times Δt .

According to subsection 7.1.1, eigenvalues computed for the experiment with *Euler Method* and *RK4* times Δt are inside the stability regions for each method. Although this shows experimentally that the algorithm is stable for a certain set of model parameters and methods, it is important to perform it because, as stated in chapter 5, it validates experimentally that both the selection of the numerical integration method and the spatial derivatives approximation method allow that the developed algorithm provides consistent results and does not produce numerical artifacts attributable to stability problems. Note that the eigenvalues of \mathbf{J}_Φ may vary according to the spatial resolution associated with N_x and N_y , but it is emphasized that these experimental results enable the proposed algorithm as a valid tool for numerically solving the model 4.1. To complement this subsection, the authors in

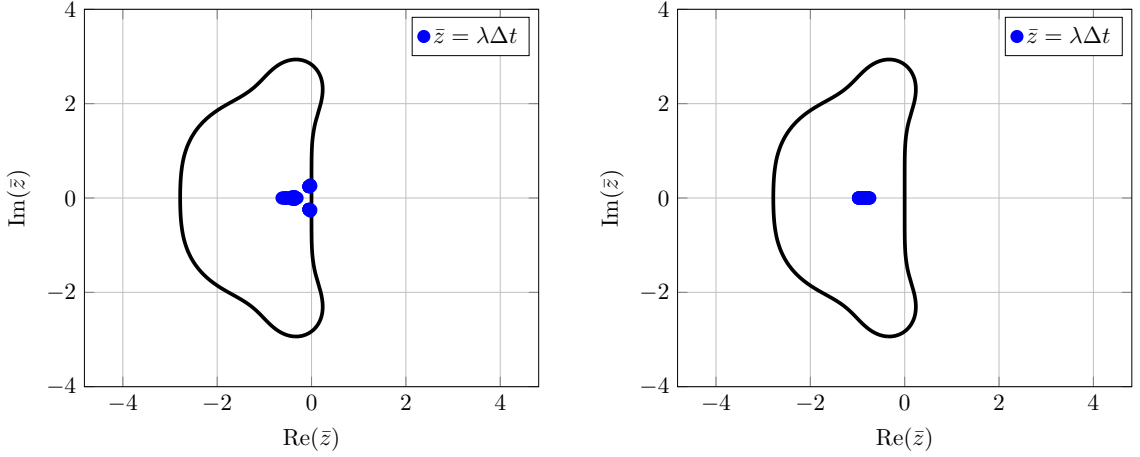


Figure 7.2: Eigenvalues of \mathbf{J}_Φ for first (left plot) and last (right plot) time step using the $RK4$ method.

Chan (1984); Reddy and Trefethen (1992); Levy and Tadmor (1998) provide a deeper stability analysis for different schemes used by *MOL* in the approximation of *PDEs*.

7.1.2 Convergence

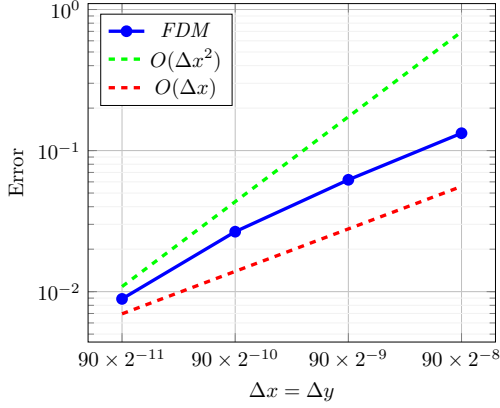
The error is computed over a finer mesh with $N_x = N_y = 4095$ and a fixed value of $N_t = 128$ using the infinity norm, that is, $\|U_{\text{finer}} - U_{\text{coarse}}\|_\infty$. Figure 7.3a shows that as $\Delta x \rightarrow 0$, the spatial convergence is $O(\Delta x^2)$. This behavior is expected due to the order accuracy of *CD FDM* used for the approximation of the spatial derivatives as described in Section 5.2.

For time convergence, the error is computed comparing the different number of time steps with $N_t = 1024$, and for a fixed $N_x = N_y = 255$ in space. The error is computed in the same way that spatial convergence using the infinity norm. The function $f(u, \beta)$ is removed from Φ for this analysis since $H_{pc}(u)$ is not a smooth function and it is not differentiable u_{pc} . Figure 7.3b shows the results of the experiment. In this case, the convergence of time scheme is $O(\Delta t^4)$ which is consistent with the theoretical convergence of the $RK4$ method detailed in Section 5.3.

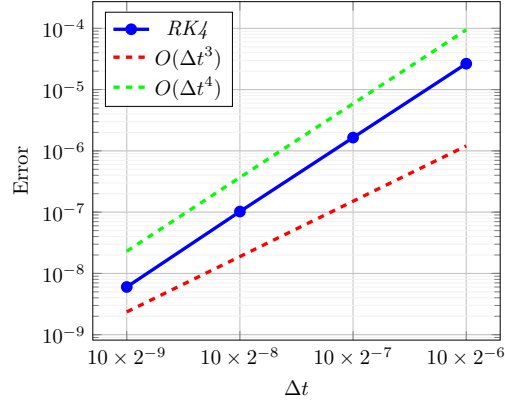
As the conclusion of this subsection can be affirmed that the proposed algorithm is consistent because the error for both approximations goes to 0 when $\Delta x = \Delta y \rightarrow 0$ and $\Delta t \rightarrow 0$

7.1.3 Complexity

The computational complexity analysis was performed for both, space and time approximation. Parameters used for the numerical experiment were the indicated in the begin of the chapter. The first analysis evaluates the complexity of the right-hand side of the equation, which theoretically is $O(N^3)$ according to Section 5.4.



(a) Space convergence.



(b) Time convergence.

Figure 7.3: Analysis of the algorithm convergence.

According to Figure 7.4a, the complexity achieved by the spatial approximation is better than the theoretically obtained, since the curve is between $O(N^2)$ and $O(N^3)$. This can be explained by the optimized implementation of matrix multiplication in *NumPy*. On the other hand, time approximation presents a linear complexity as shown in Figure 7.4b. This is because the $RK4$ method evaluates Φ 4 times, but the complexity is linear to the number of time steps N_t .

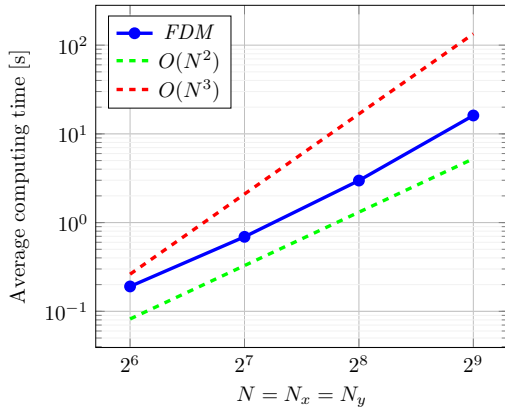
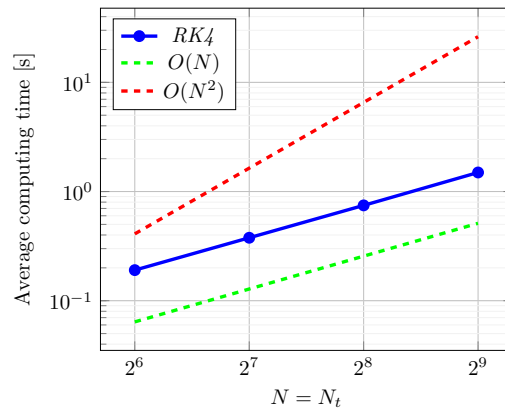

 (a) Complexity of the Φ approximation.

 (b) Complexity of the $RK4$ method.

Figure 7.4: Analysis of the algorithm complexity.

This subsection allows to guarantees that the numerical algorithm can compute numerical simulations in affordable times because the computational complexity is polynomial with respect to the problem size.

7.2 Numerical Simulation

For a qualitative fire spreading analysis, the following subsections present several numerical experiments using the algorithm.

7.2.1 Asensio & Ferragut output

The first numerical simulation tries to reproduce the output shown in Asensio and Ferragut (2002). To achieve a similar wildfire shape, the following parameters were used:

- Physical parameters:
 - Domain $\Omega \times [t_{\min}, t_{\max}] = [0, 90]^2 \times [0, 30]$. In physical domain it is equivalent to a square of 26 [m] \times 26 [m] and 70 [h] of simulation.
 - Temperature and fuel were set to zero at boundary, $u_{\Gamma}(\mathbf{x}, t) = \beta_{\Gamma}(\mathbf{x}, t) = 0$, on $\Gamma \times [0, 30]$.
 - Initial condition for temperature is $u_0(\mathbf{x}) = 6 \exp(-0.05((x - 20)^2 + (y - 20)^2))$ and for fuel $\beta_0(\mathbf{x})$ was initialized with a uniformly distributed random variable between $[0, 1]$.
 - The vector field used was $\mathbf{v}(\mathbf{x}, t) = (\cos(\pi/4), \sin(\pi/4))$.
 - Model parameters $\kappa = 0.1$, $\varepsilon = 0.3$, $u_{pc} = 3$, $q = 1$, $\alpha = 0.001$.
- Numerical parameters:
 - Number of space nodes for *FDM*, $N_x = N_y = 127$
 - Number of time nodes for *RK4*, $N_t = 500$.

The result of this experiment is shown in Figure 7.5. The high-temperature zone, or fire front, is formed following the wind direction. Also, the fuel is consumed in the same shape as the fire front. The maximum temperature value reached is around $u = 9$, which in physical variables corresponds to $T = 1110$ [K]. This result is within the values measured in a forest fire.

Based on these results, the output of the numerical simulation is in agreement with Figure 7.6.

7.2.2 Controlled fire

The second numerical experiment attempts to simulate a controlled wildfire. The parameters for this simulation were:

- Physical parameters:
 - $\Omega \times [t_{\min}, t_{\max}] = [-100, 100]^2 \times [0, 30]$ as domain. In physical domain it is equivalent to a square of 58 [m] \times 58 [m] and 70 [h] of simulation.

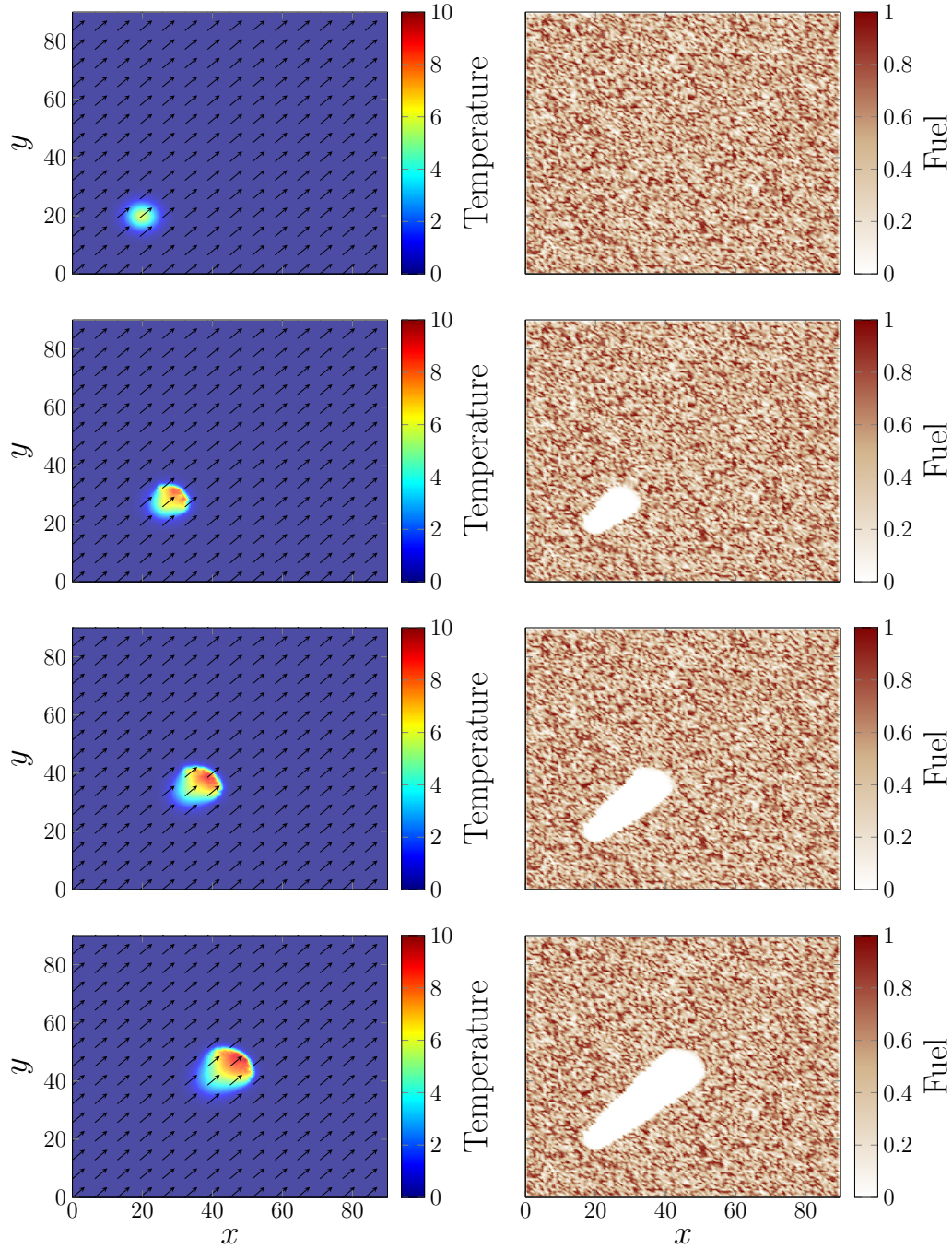


Figure 7.5: Numerical simulation to achieve Asensio and Ferragut (2002) output. Left column presents the nondimensional temperature and right column presents the fuel fraction. Times for each row are: $t = \{0, 10, 15, 20\}$.

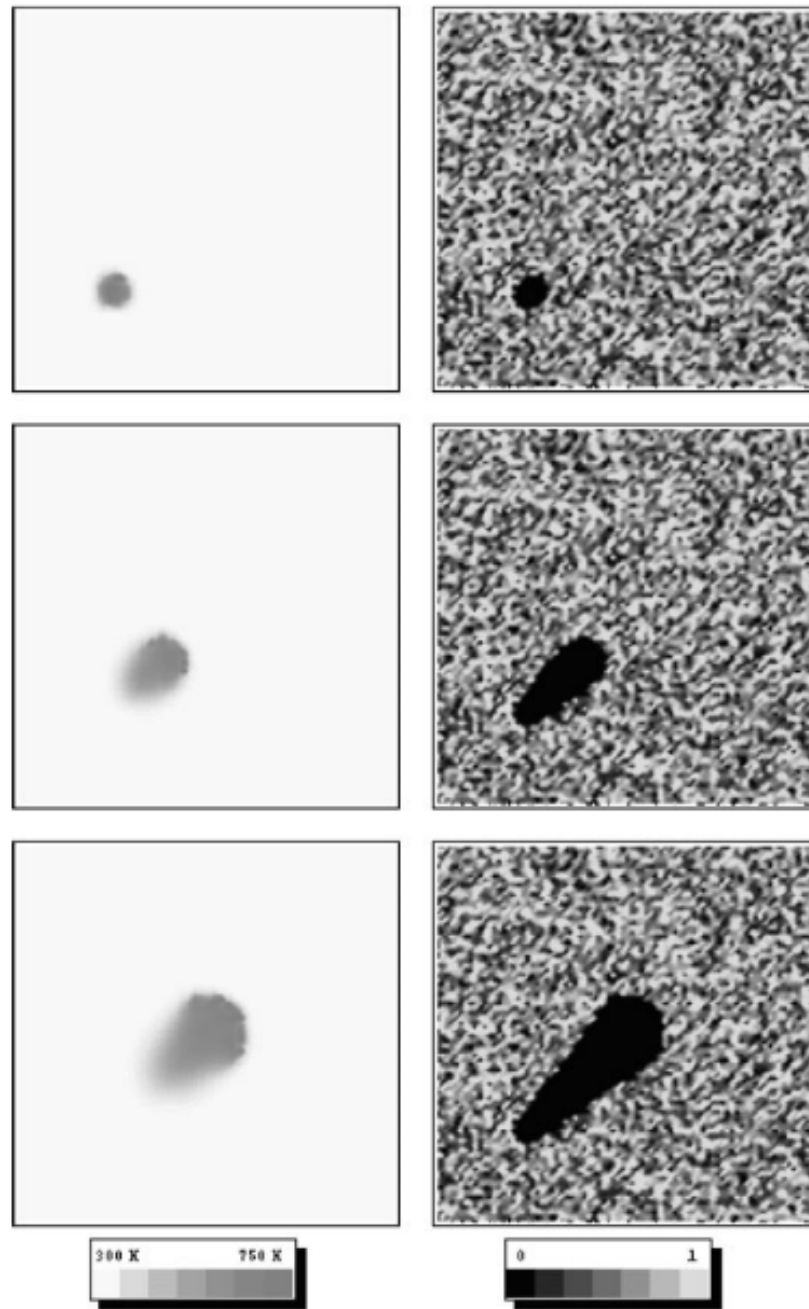


Figure 7.6: Numerical simulation output adopted from Asensio and Ferragut (2002, p. 155). Left column presents the temperature and right column presents the fuel fraction. Republished with permission of John Wiley & Sons - Books, from *On a wildland fire model with radiation*, Asensio, M. I.; Ferragut, L., 54, 1, 2002; permission conveyed through Copyright Clearance Center, Inc.

- Temperature and fuel were set to zero at boundary, $u_\Gamma(\mathbf{x}, t) = \beta_\Gamma(\mathbf{x}, t) = 0$, on $\Gamma \times [0, 30]$.
- Initial condition for fuel $\beta_0(\mathbf{x})$ was initialized with an uniformly distributed random variable between $[0, 1]$ and temperature used was

$$u_0(\mathbf{x}) = \begin{cases} 6 & \text{if } -100 \leq x \leq -97 \text{ and } -35 \leq y \leq 34 \\ 0 & \text{otherwise.} \end{cases}$$

- The vector field used was $\mathbf{v}(\mathbf{x}, t) = (1, 0)$.
- Model parameters $\kappa = 10$, $\varepsilon = 0.3$, $u_{pc} = 1$, $q = 3$, $\alpha = 0.01$.
- Numerical parameters:
 - Number of space nodes for *FDM*, $N_x = N_y = 127$
 - Number of time nodes for *RK4*, $N_t = 3000$.

Figure 7.7 shows the result of the second experiment. The output of this experiment is qualitatively similar to figures presented in Mell et al. (2013), presented in Figure 7.8, where authors study grassland fires. These simulations are compared to a real experiment performed in controlled conditions.

7.2.3 Wind effect

The last experiment shows the effect of a time-depending wind effect. The parameters used for this experiment were the same of the first experiment, but $t \in [0, 50]$, temperature initial condition was defined as $u_0(\mathbf{x}) = 6 \exp(-0.05((x-20)^2 + (y-70)^2))$ and vector field was defined as $\mathbf{v}(\mathbf{x}, t) = (\cos(-\pi/4 + 0.01t), \sin(\pi/4 + 0.01t))$. Figure 7.9 shows the last experiment result, where the fire front spreads following the wind direction.

This experiment shows that the algorithm allows the use of dynamic vector fields in the simulations, for instance, wind depending on time.

7.3 Risk Maps

The generation of risk maps or vulnerability maps is one of the specific objectives of this thesis work. Three examples for this kind of application are presented in the next experiments.

The first experiment implements the scheme described in Figure 4.1 to build two risk maps. This experiment considered the physical parameter $A = 1 [s^{-1}]$ and characteristic magnitudes $l_0 = 2.1 [m]$ and $t_0 = 440529.3 [s]$. Then, the parameters used for this numerical simulation were:

- Physical parameters:

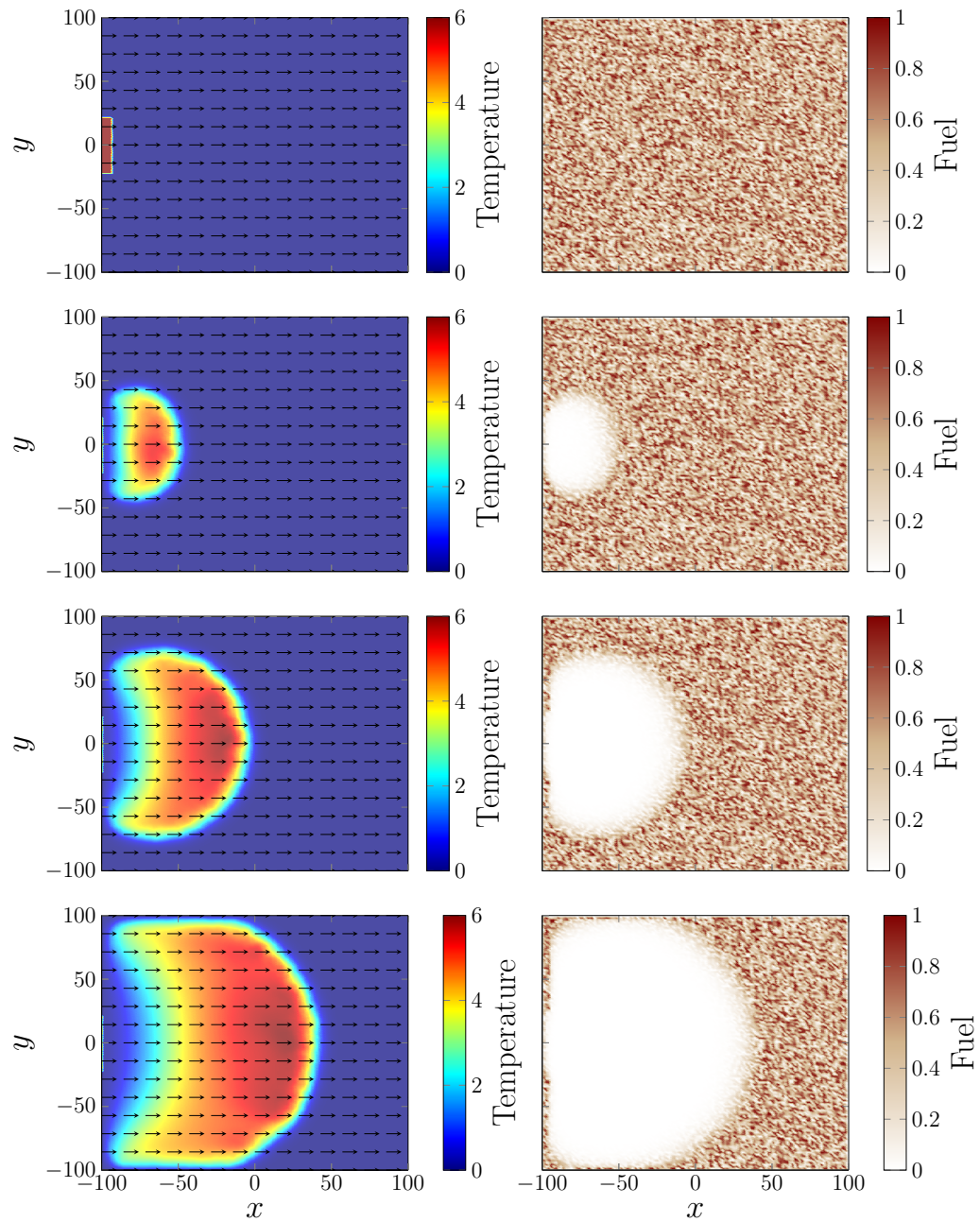


Figure 7.7: Experiment of controlled fire at $t = \{0, 10, 15, 20\}$. Left column presents the nondimensional temperature and right column presents the fuel fraction.

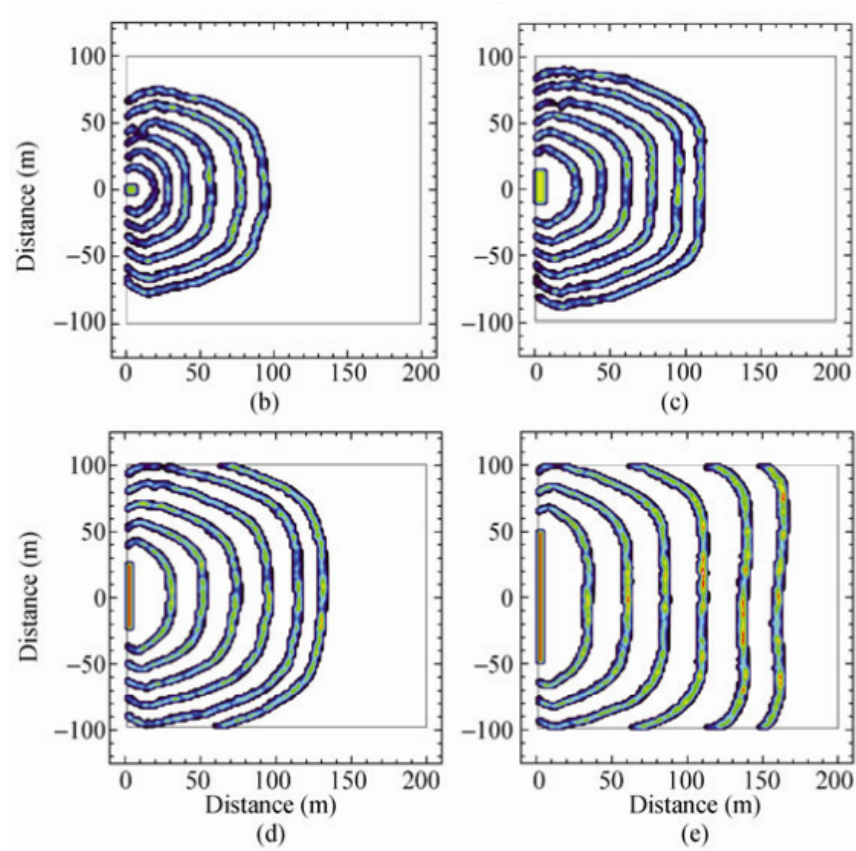


Figure 7.8: Numerical simulation output adopted from Mell et al. (2013, p.11). Republished with permission of Springer, from Numerical Simulations of Grassland Fire Behavior from the LANL-FIRETEC and NIST-WFDS Models, Mell W., Charney J., Jenkins M.A., Cheney P., Gould J, 2013; permission conveyed through Copyright Clearance Center, Inc.

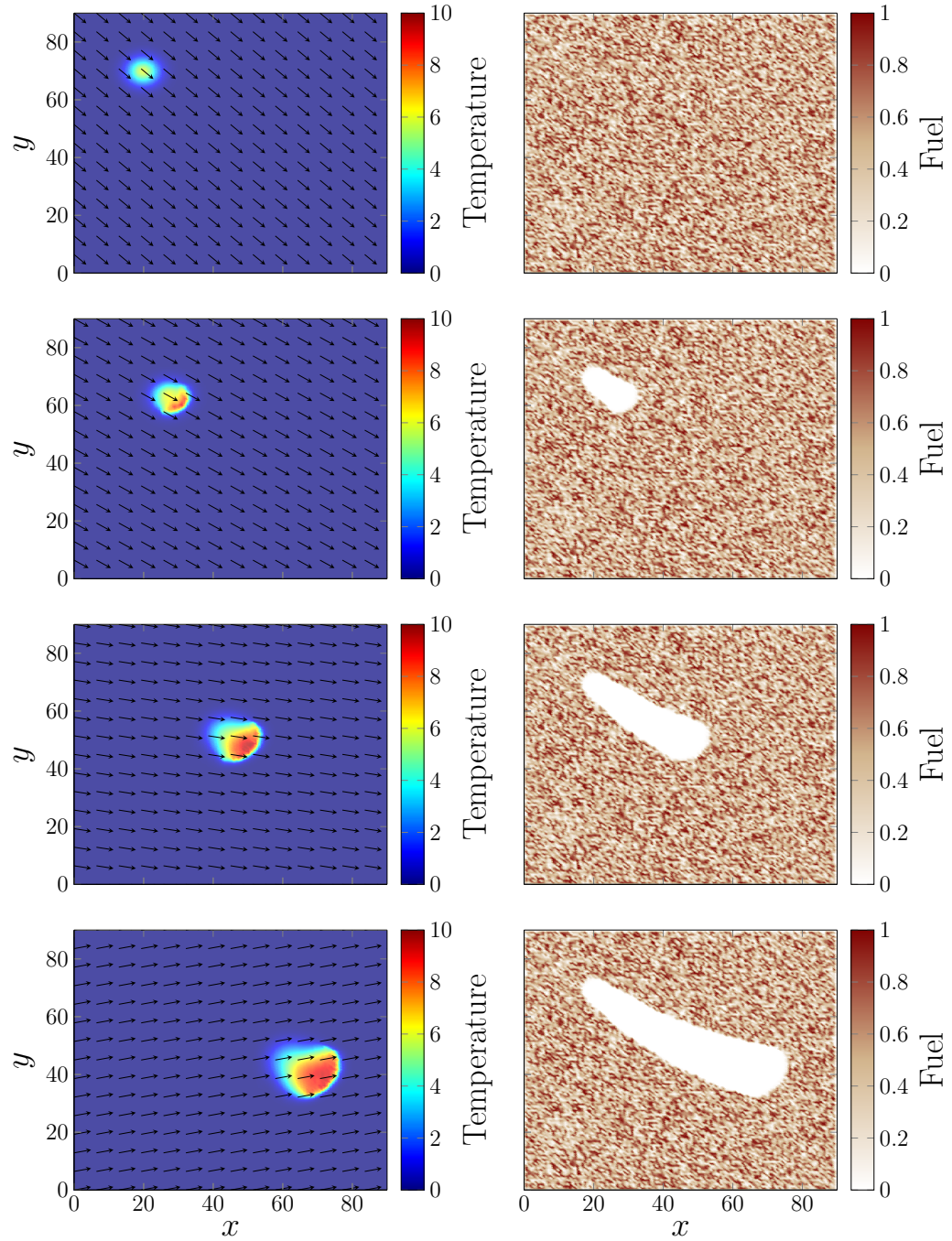


Figure 7.9: Wind effect experiment at $t = \{0, 10, 30, 50\}$. Left column presents the nondimensional temperature and right column presents the fuel fraction.

- The domain $\Omega \times [t_{\min}, t_{\max}] = [-1, 1]^2 \times [0, 0.1]$. In physical domain it is equivalent to a square of 4 [m] \times 4 [m] and 12 [h] of simulation.
- Temperature and fuel were set to zero at boundary, $u_\Gamma(\mathbf{x}, t) = \beta_\Gamma(\mathbf{x}, t) = 0$, on $\Gamma \times [0, 30]$.
- Initial fire profile was $u_0(\mathbf{x}) = 6 \exp(-100((x + x_r)^2 + (y + y_r)^2))$, with $x_r = y_r = \{d : d = -0.8 + 0.05 \times r, r = \{0, 1, \dots, 32\}\}$. Notice that there are 33×33 ignition points. Initial fuel condition was set as

$$\begin{aligned} \beta_0(\mathbf{x}) = & 0.5 \exp(-(x + 0.75)^2 + (y - 0.75)^2) / 0.6 \\ & + 0.9 \exp(-(x - 0.75)^2 + (y + 0.75)^2) \\ & + 0.016 \exp(-(x + 0.65)^2 + (y + 0.65)^2) / 0.3 \\ & + 0.015 \exp(-(x - 0.65)^2 + (y - 0.65)^2) / 0.7. \end{aligned}$$

- The vector field used in this experiment includes wind effect with $\mathbf{w}(\mathbf{x}, t) = 10 (\cos(7/4 \pi), \sin(7/4 \pi))$, and terrain effect with

$$\begin{aligned} Z(\mathbf{x}) = & 4 \left(\exp(-(x + 0.5)^2 + (y + 0.5)^2) / 0.6 \right. \\ & \left. + \exp(-(x - 0.9)^2 + (y - 0.9)^2) / 0.9 \right). \end{aligned}$$

- Model parameters $\kappa = 0.01$, $\varepsilon = 0.1$, $u_{pc} = 1$, $q = 0.005$, $\alpha = 0.001$.
- Numerical parameters:
 - Number of space nodes for *FDM*, $N_x = N_y = 127$
 - Number of time nodes for *RK4*, $N_t = 1000$.

Figure 7.10 shows the results of the first experiment. The top plot shows initial conditions for fuel, topography, and wind using parameters described before. The middle plot shows the time it took each ignition point to consume 10% of the total fuel available, where white indicates that 10% was not reached within the numerical simulation time. The bottom plot shows the percentage of burned fuel at the end of the numerical simulation for each ignition point.

The second experiment shows fuel consumption rates for applications such as vulnerability zone analysis. To build this map *Forward Difference FDM* was used with the approximations of fuel as follow $\frac{B^{n+1} - B^n}{\Delta t}$, using the second experiment of Section 7.2. The output of this maps is presented in Figure 7.11. This example of a vulnerability map allows the identification of prone areas according to the amount of existing fuel and the characteristics of the fire front.

The experiments carried out in this subsection show that the algorithm allows developing an analysis of the effects of different forest fire scenarios.

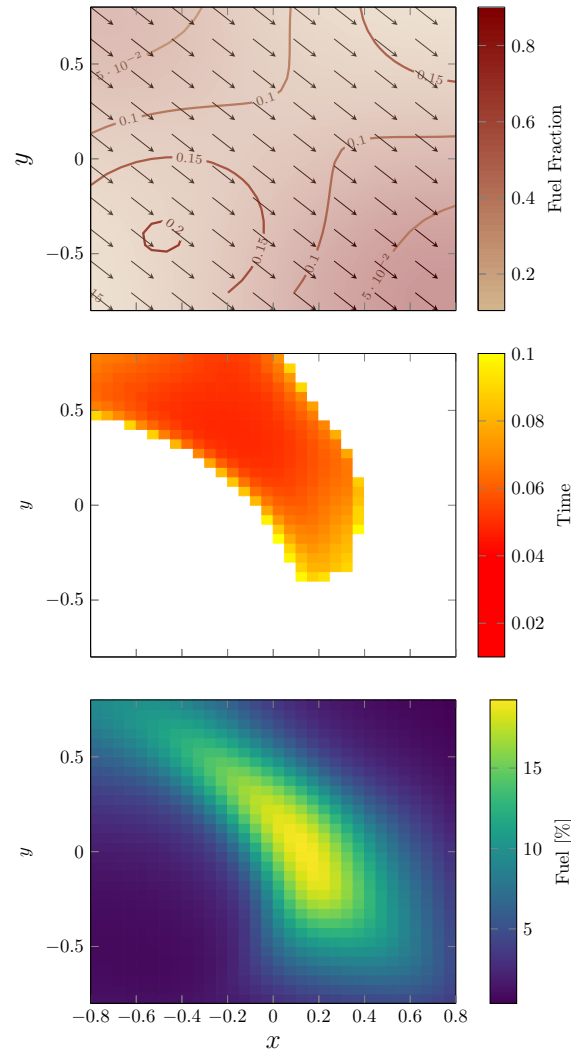


Figure 7.10: Examples of *risk maps* generated from multiple numerical simulations. Top plot shows the initial conditions for the experiment. Middle plot shows the time that 10% of fuel was burnt. The darkest red denotes a more dangerous fire ignition zone. Bottom plot shows the percentage of fuel burned at end of numerical simulation. The yellow zones indicates a more dangerous fire ignition zone.

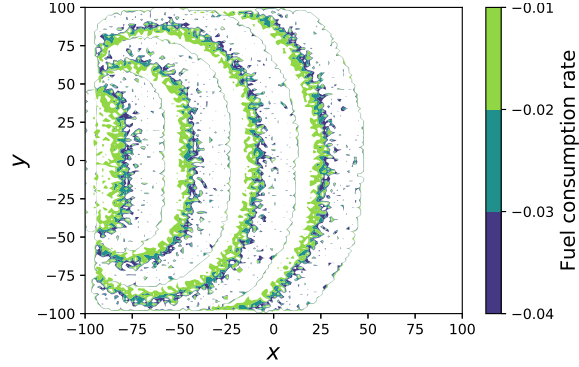


Figure 7.11: Burnt rate map example. Blue values indicates a faster fuel consumption rate.

7.4 CPU-GPU Performance

To compare the computational capacity of each of the implementations, an equi-spaced grid of ignition points is defined, as described in Figure 4.1. Under this definition, experiments are performed for $N_s = \{1, 10, 100, 1000, 8100\}$ numerical simulations. The goal is to process a high number of simulations, in order to compare the execution times of *CPU* and *GPU* implementations.

The code was executed in the cluster provided by the Centro Científico Tecnológico de Valparaíso of the Universidad Técnica Federico Santa María (CCTVal). The available nodes have an Intel(R) Xeon(R) E5-2643 v2 *CPU* of 3.90 [GHz] frequency, 6 physical cores, and 12 logical cores. These include a Tesla K20m graphics card, which has 2496 cores with 706 [MHz] frequency and 5 [GB] memory. They also have 64 [GB] of RAM.

For the implementation in *OpenMP*, 12 threads were used because they had the smallest average execution time. For the parameters selection of *CUDA kernel*, 10 experiments and 8100 simulations were performed. The results are presented in Table 7.1. Therefore, the choice of 512 blocks with 256 threads gets the lowest average

Table 7.1: Average execution time [s] of the algorithm for a different number of threads and blocks.

		Blocks				
		32	64	128	256	512
Threads	128	116.667	74.175	60.518	62.10	61.138
	256	74.009	59.104	59.689	58.829	57.66
	512	73.375	65.806	60.688	60.89	59.026

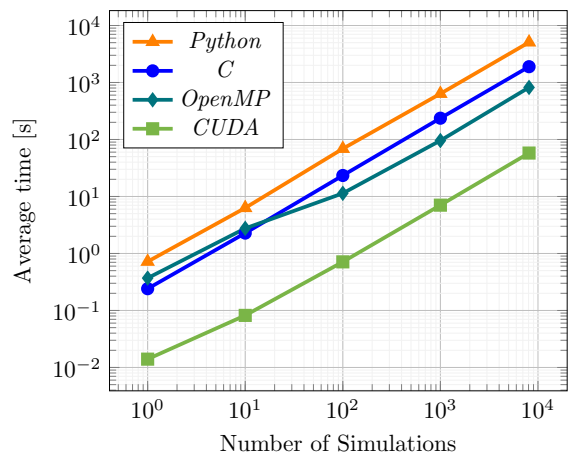
execution time, being 49.42 less than the worst-case with 32 blocks and 128 threads. After obtaining the best parameters for *GPU*, Table 7.2 and Figure 7.12a presents the result of the experiments for *Python*, *C*, *OpenMP*, and *CUDA* implementations. Analyzing the curves of Figure 7.12a, the implementation of *GPU* significantly re-

Table 7.2: Algorithm implementations average times in [s].

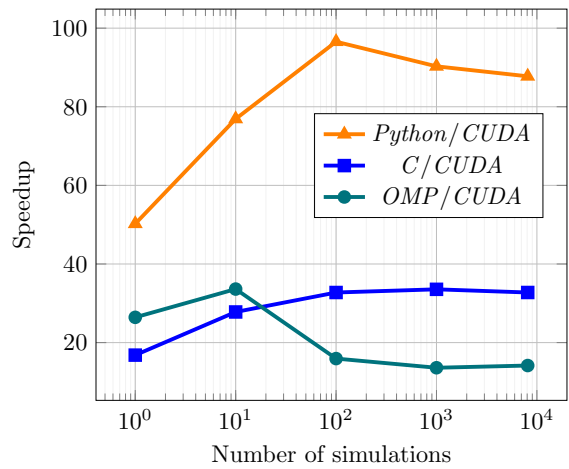
N_s	<i>Python</i>	<i>C</i>	<i>OpenMP</i>	<i>CUDA</i>
1	0.72	0.241	0.37	0.014
10	6.306	2.279	2.757	0.082
100	68.872	23.359	11.362	0.713
1000	635.772	236.292	95.792	7.042
8100	5073.632	1893.581	819.075	57.821

duces the computing time required to process different scenarios. Note that for less than 10 simulations, *OpenMP* shows worse performance due to the fixed cost associated with synchronization work. Additionally, to compare the results obtained the *speedup* is introduced as the ratio between the *CPU* implementations run time over *GPU*. The *speedups* obtained from numerical simulations times of Table 7.2 are shown in Figure 7.12b

For $N_s \geq 100$, the execution on *GPU* is about 80 times faster than the prototype version implemented in *Python*, 30 times faster than a *C* implementation, and approximately 15 times faster than *OpenMP* implementation. This shows that a GPU implementation outperforms *CPU* versions and improves performance when a large number of numerical simulations are required.



(a) Average times.



(b) Speedup.

Figure 7.12: CPU-GPU performance comparison.

Chapter 8

Open-source Framework

Finally, based on the results presented in the previous chapters, the framework proposed by this work is presented below. This framework is named *Ngen-Kütral*, which means Spirit of Fire in Mapudungun (Biblioteca Nacional de Chile, n.d.), the language of the Mapuche people. Two versions are available, the first one developed in *Python* and the second one in *CUDA*. Both versions are publicly available on *GitHub* and can be accessed in (San Martín & Torres, 2018c) and (San Martín & Torres, 2018b).

8.1 *Python*

8.1.1 Prerequisites

The following libraries must be installed to execute the *Framework*:

- Python \geq 3.6.5
- Numpy \geq 1.13.3
- Scipy \geq 1.1.0
- Matplotlib \geq 2.2.2

8.1.2 Installation

Clone the repository to some user-defined directory,

```
user@host:~ $ git clone https://github.com/dsanmartin/ngen-kutral.git
```

Add the following line in `~/.profile`, `~/.bash_profile`, or `~/.bashrc` file according to the Operating System configuration.

```
export PYTHONPATH=$PYTHONPATH:/path/of/the/ngen-kutral/
```

8.1.3 Usage

The framework can be used as a module, importing it.

```
import wildfire
```

Also, it can be used as a wrapper script inside the `bin` folder:

```
usage: main.py -sm SM -Nx NX -Ny NY -tm TM -Nt NT [-h] [-k K]
        [-e E] [-p P] [-a A] [-q Q] [-x XMIN XMAX] [-y YMIN YMAX]
        [-t TMIN TMAX] [-u0 U0] [-b0 B0] [-vf VF] [-w W] [-T Tx Ty]
        [-acc ACC] [-sps S] [-lst LST] [-plt PLT]
```

Create and execute a wildfire simulation.

required arguments:

<code>-sm SM, --space SM</code>	Space method approximation, FD (Finite Difference) or FFT (Fast Fourier Transform).
<code>-Nx NX, --xnodes NX</code>	Number of nodes in x.
<code>-Ny NY, --ynodes NY</code>	Number of nodes in y.
<code>-tm TM, --time TM</code>	Time method approximation, Euler or RK4.
<code>-Nt NT, --tnodes NT</code>	Number of nodes in t.

optional arguments:

<code>-h, --help</code>	Show this help message and exit.
<code>-k K, --kappa K</code>	Kappa parameter (default: 0.1).
<code>-e E, --epsilon E</code>	Epsilon parameter (default: 0.3).
<code>-p P, --phase P</code>	Phase change threshold parameter (default: 3.0).
<code>-a A, --alpha A</code>	Alpha parameter (default: 1e-3).
<code>-q Q, --reaction Q</code>	Reaction heat coefficient (default: 1.0).
<code>-x XMIN XMAX, --xlimits XMIN XMAX</code>	x domain limits (default: [0, 90]).
<code>-y YMIN YMAX, --ylimits YMIN YMAX</code>	y domain limits (default: [0, 90]).
<code>-t TMIN TMAX, --tlimits TMIN TMAX</code>	t domain limits (default: [0, 30]).
<code>-u0 U0, --initial-temperature U0</code>	Initial temperature file. Only .txt and .npz supported (default lambda testing function).
<code>-b0 B0, --initial-fuel B0</code>	Initial fuel file. Only .txt and .npz supported (default lambda testing function).
<code>-vf VF, --vector-field VF</code>	Vector Field. Only .txt and .npz supported (default lambda testing function).
<code>-w W, --wind W</code>	Wind component. Only .txt and .npz supported (default lambda testing function).
<code>-T Tx Ty, --terrain Tx Ty</code>	Topography gradient effect. Only .txt and .npz supported (default no topography effect).
<code>-acc ACC, --accuracy ACC</code>	Finite difference accuracy (default: 2).

```
-sps S, --sparse S      Finite difference sparse matrices (default: 0).  
-lst LST, --last LST    Only last approximation (default: 1).  
-plt PLT, --plot PLT    Plot result (default: 0).
```

8.2 *CUDA*

8.2.1 Prerequisites

The computer must have a compatible *CUDA* version. It depends on the Operating System and the Graphic Card. Please use this link <https://docs.nvidia.com/cuda/> and follow the instructions according to your configuration.

8.2.2 Installation

Clone the repository to some user-defined directory,

```
user@host:~ $ git clone https://github.com/dsanmartin/ngen-kutral-gpu.git
```

Go inside the folder and compile with `make`

```
cd ngen-kutral-gpu && make
```

8.2.3 Usage

Execute the following command

```
./bin/ngen-kutral SM TM Nx Ny Nt x_ign_n y_ign_n \  
threads blocks save_output exp_n
```

where

- SM the spatial approximation method. Only *FD* at the moment.
- TM the time integration method. *Euler* or *RK4*.
- Nx Ny Nt number of spatial and time nodes.
- x_ign_n y_ign_n number of simulations per *x* and *y* axis.
- threads blocks *CUDA* parameters.
- save_output 1 to save or 0 to does not save the output.
- exp_n experiment identifier.

8.3 Data

Currently, the data manipulation is only available for *Python* implementation. The user can put the input/output files inside the `data` folder. The framework supports both `.npy` and `.txt` file formats for fuel and temperature initial conditions, and wind/terrain vector field.

8.3.1 How to use

- Temperature initial condition: Use one file with supported file extensions. File size must match with N_x and N_y parameters.
- Fuel initial condition: Use one file with supported file extensions. File size must match with N_x and N_y parameters.
- Vector field, terrain + wind:
 - If the vector field is constant in space, use a supported file with shape $(N_t + 1) \times 2$, one column per component.
 - If the vector field depends on space and time, a `.npy` file with shape $(N_t + 1) \times 2 \times N_y \times N_x$ can be used. Also, a folder with `.txt` files can be used with the following format: `V1_timestep.txt` and `V2_timestep.txt` for each time step, that is for 0 to N_t .
 - If the wind and topography effect are independent files, the framework supports:
 1. One file for wind effect with shape $(N_t + 1) \times 2$. Wind is assumed to be time-dependent only.
 2. Two files for topography effect (gradient of topography), both with shapes $N_y \times N_x$. Topography only depends on space variables.

8.3.2 Example files

An example folder is provided for some testing files.

- `U0.npy` and `U0.txt` for temperature initial condition. The shape of both files is 128×128 .
- `B0.npy` and `B0.txt` for fuel initial condition. The shape of both files is 128×128 .
- For vector field it is included:
 - `V.npy` and `V.txt` files with shape 101×2 .

- `VV.npy` file of shape $101 \times 2 \times 128 \times 128$. Also includes a `V` directory with 101 files of shape 128×128 for both vector components. The data is the same but `.npy` can handle different shapes, while `.txt` does not.
- `W.npy` and `W.txt` for independent wind effect with shape 101×2 , `Tx.npy/Tx.txt`, and `Ty.npy/Ty.txt` for independent terrain effect both with shape 128×128 .

Chapter 9

Conclusions

This thesis work has successfully presented an *Open-source Framework* to support wildfire spreading and effects analysis in Chile. Using the numerical implementation of a two-dimensional simplified wildfire model, this work allows the numerical simulation of several fire scenarios. These simulations depend on components such as temperature, fuel fraction, wind, and topography which are parts of the mathematical developed by Asensio and Ferragut (2002), using the assumptions of Eberle et al. (2015). For the numerical approximation of the mathematical model, the proposed algorithm is based on *Method of Lines*, which splits the space and time dependency allowing to approximate independently using different classes of numerical methods. In this implementation was used *CD FDM* for space and *RK4* for time. Regarding the proposed objectives of this work:

1. Provided analysis and description of the model used, including physical derivation, its assumptions, and simplifications.
2. Presented an algorithm with the numerical method used, the stability, convergence, complexity, and simulation capabilities. Since the results were positive, we can experimentally assure that the proposed algorithm is valid for the approximation of the mathematical model.
3. Developed three *CPU* implementations: a prototype in *Python*, a sequential version in *C*, a multi-threaded using *OpenMP*, and a *GPU* implementation using *CUDA*. The last one is required by a computationally intensive task such as *risk map* generation. The implementations can load data using the format detailed in the source code repositories.
4. Showed the performance and comparison of both *CPU* and *GPU* implementations.

Satisfactorily, this work goes in the direction to keep improving the current open-source computational tools for wildfire spreading analysis. As results, the developed framework is publicly available in San Martín and Torres (2018c) and San Martín and Torres (2018b). It is important to highlight that the development of this work allowed

the publication of two conference papers presented at the *37th International Conference of the Chilean Computer Science Society* (San Martín & Torres, 2018d) and the *38th International Conference of the Chilean Computer Science Society* (San Martín & Torres, 2019).

Finally, from the analysis conducted in the section 7.4, the hypothesis of this work which established that: “*Based on a mathematical model of wildfire spreading, a parallel algorithm is proposed on a GPU approach that will reduce the computational time of multiple simulations execution compared to a CPU implementation*” has been successfully validated.

9.1 Future Work

Following the guidelines developed in this thesis, as future work at least three possible routes can be detached:

1. Explore the use of spectral methods in the approximation of partial derivatives with respect to spatial variables. For instance, the use of Chebyshev’s differentiation matrices (Trefethen, 2000), which require a smaller number of nodes to obtain better approximations when computing the derivatives. Also, the use of the *Fourier Transform* can be explored, taking advantage of the optimized libraries for the computation of the *Fast Fourier Transform* in both *CPU* and *GPU*. A prototype *CPU* version was developed in San Martín and Torres (2019).
2. Optimize the code in *GPU* using the different types of memory provided by *CUDA*. It is possible to improve the performance of *GPU* codes by correctly using the *CUDA* memory model. In particular, the use of shared memory can significantly accelerate the performance of some operations as shown in Appendix A.2.
3. Improve the quality of the mathematical model used, coupling a weather model that allows a more realistic wind behavior. For example, the use of the Navier-Stokes equations or some variant that enables the consideration of the atmospheric variation generated by the increase of temperature in a wildfire episode. It is necessary to consider that the numerical methods involved in fluid dynamics are computationally expensive, so they require the use of parallel programming techniques such as those explored in this work.

References

- Albini, F. A., Brown, J. K., Reinhardt, E. D., & Ottmar, R. D. (1995). Calibration of a large fuel burnout model. *International Journal of Wildland Fire*, 5(3), 173–192. doi: 10.1071/WF9950173
- Almeida, R. M., & Macau, E. E. N. (2011, 3). Stochastic cellular automata model for wildland fire spread dynamics. *Journal of Physics: Conference Series*, 285(1), 12038. Retrieved from <http://stacks.iop.org/1742-6596/285/i=1/a=012038> doi: 10.1088/1742-6596/285/1/012038
- Ambroz, M., Balažovjeh, M., Medl’a, M., & Mikula, K. (2018). Numerical modeling of wildland surface fire propagation by evolving surface curves. *Advances in Computational Mathematics*, 45(2), 1067–1103. doi: 10.1007/s10444-018-9650-4
- Andrews, P. L. (2014). Current status and future needs of the BehavePlus Fire Modeling System. *International Journal of Wildland Fire*, 23(1), 21–33. doi: 10.1071/WF12167
- Archibald, S., Lehmann, C. E. R., Gómez-Dans, J. L., & Bradstock, R. A. (2013, 4). Defining pyromes and global syndromes of fire regimes. *Proceedings of the National Academy of Sciences of the United States of America*, 110(16), 6442–7. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/23559374><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3631631> doi: 10.1073/pnas.1211466110
- Asensio, M. I., & Ferragut, L. (2002, 5). On a wildland fire model with radiation. *International Journal for Numerical Methods in Engineering*, 54(1), 137–157. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.420> doi: 10.1002/nme.420
- Balbi, J. H., Santoni, P. A., & Dupuy, J. L. (2000). Dynamic Modelling of Fire Spread Across a Fuel Bed. *International Journal of Wildland Fire*, 9(4), 275–284. doi: 10.1071/wf00005
- Baranovskiy, N. V., & Nee, A. E. (2015). Mathematical simulation of heat transfer in surface soil layer at forest fire influence in three-dimension statement. *EPJ Web of Conferences*, 82, 01004. doi: 10.1051/epjconf/20158201004
- Beezley, J. D., & Mandel, J. (n.d.). *WRF-Fire - Open Wildland Fire Modeling E community Wiki*. Retrieved from <https://www.openwfm.org/wiki/WRF-Fire>
- Biblioteca Nacional de Chile. (n.d.). *Mapudungun - Memoria Chilena, Biblioteca Nacional de Chile*. Retrieved from <http://www.memoriachilena.gob.cl/602/>

- w3-article-3594.html
- Bürger, R., Gavilán, E., Inzunza, D., Mulet, P., & Villada, L. M. (2020a). Exploring a Convection–Diffusion–Reaction Model of the Propagation of Forest Fires: Computation of Risk Maps for Heterogeneous Environments. *Mathematics*, 8(10), 1674. doi: 10.3390/math8101674
- Bürger, R., Gavilán, E., Inzunza, D., Mulet, P., & Villada, L. M. (2020b). Implicit-explicit methods for a convection-diffusion-reaction model of the propagation of forest fires. *Mathematics*, 8(6). doi: 10.3390/math8061034
- Chan, T. F. (1984, 4). Stability Analysis of Finite Difference Schemes for the Advection-Diffusion Equation. *SIAM Journal on Numerical Analysis*, 21(2), 272–284. Retrieved from <https://doi.org/10.1137/0721020> doi: 10.1137/0721020
- Cheng, J., Grossman, M., & McKercher, T. (2013). *Professional CUDA C Programming* (Vol. 53) (No. 9).
- Clark, T. L., Coen, J., & Latham, D. (2004). Description of a coupled atmosphere-fire model. *International Journal of Wildland Fire*, 13(1), 49–63. doi: 10.1071/WF03043
- Clark, T. L., Jenkins, M. A., Coen, J., & Packham, D. (1996, 6). A Coupled Atmosphere-Fire Model: Convective Feedback on Fire-Line Dynamics. *Journal of Applied Meteorology*, 35(6), 875–901. Retrieved from <http://journals.ametsoc.org/doi/abs/10.1175/1520-0450%281996%29035%3C0875%3AACAMCF%3E2.0.CO%3B2> doi: 10.1175/1520-0450(1996)035<0875:ACAMCF>2.0.CO;2
- Coen, J. (n.d.). *The CAWFE Modeling System for Wildland Fires*. Retrieved from https://www2.mmm.ucar.edu/people/coen/files/newpage_c.html
- Coen, J. L. (2005). Simulation of the Big Elk Fire using coupled atmosphere-fire modeling. *International Journal of Wildland Fire*, 14(1), 49–59. doi: 10.1071/WF04047
- Coen, J. L. (2013). *Modeling Wildland Fires : of the Coupled Atmosphere- Wildland Fire Environment Model (CAWFE)* (Tech. Rep.). doi: 10.1071/WF04047
- CONAF. (2021). *Incendios Forestales en Chile*. Retrieved from <http://www.conaf.cl/incendios-forestales/incendios-forestales-en-chile/>
- Costafreda-Aumedes, S., Comas, C., & Vega-Garcia, C. (2017). Human-caused fire occurrence modelling in perspective: A review. *International Journal of Wildland Fire*, 26(12), 983–998. doi: 10.1071/WF17026
- Doerr, S., & Santin, C. (2017). Global trends in wildfire-perceptions and realities in a changing world. *Geophysical Research Abstracts*, 19, 2017–17979.
- Eberle, S. (2014). Modeling and Simulation of Forest Fire Spreading. In Pardo-Igúzquiza Eulogio et al. (Eds.), *Mathematics of planet earth* (pp. 811–814). Berlin, Heidelberg: Springer, Berlin, Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-32408-6_175 doi: 10.1007/978-3-642-32408-6_{-}175
- Eberle, S., Freeden, W., & Matthes, U. (2015). Forest fire spreading. In W. Free-

References

- den, M. Z. Nashed, & T. Sonar (Eds.), *Handbook of geomathematics* (pp. 1349–1385). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-54551-1_70 doi: 10.1007/978-3-642-54551-1{_}70
- Eskandari, S. (2016). Application of a CA-based model to predict the fire front in Hyrcanian forests of Iran. *Arabian Journal of Geosciences*, 9(17). doi: 10.1007/s12517-016-2717-y
- FARSITE — Fire, Fuel, and Smoke Science Program. (n.d.). Retrieved from <https://www.firelab.org/project/farsite>
- FDS-SMV. (n.d.). Retrieved from <https://pages.nist.gov/fds-smv/>
- Fečkan, M., & Pačuta, J. (2018, 9). Existence of solution of a forest fire spread model. *Applied Mathematics Letters*, 83, 227–231. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0893965918301162?via%3Dihub> doi: 10.1016/J.AML.2018.03.035
- Fernandez-Anez, N., Christensen, K., & Rein, G. (2017, 7). Two-dimensional model of smouldering combustion using multi-layer cellular automaton: The role of ignition location and direction of airflow. *Fire Safety Journal*, 91, 243–251. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0379711217301753> doi: 10.1016/j.firesaf.2017.03.009
- Ferragut, L., Asensio, M. I., Cascón, J. M., & Prieto, D. (2015, 1). A Wildland Fire Physical Model Well Suited to Data Assimilation. *Pure and Applied Geophysics*, 172(1), 121–139. Retrieved from <http://link.springer.com/10.1007/s00024-014-0893-9> doi: 10.1007/s00024-014-0893-9
- Ferragut, L., Asensio, M. I., & Monedero, S. (2007a). Modelling radiation and moisture content in fire spread. *Communications in Numerical Methods in Engineering*, 23(9), 819–833. Retrieved from <https://doi.org/10.1002/cnm.927> doi: 10.1002/cnm.927
- Ferragut, L., Asensio, M. I., & Monedero, S. (2007b, 6). A numerical method for solving convection-reaction-diffusion multivalued equations in fire spread modelling. *Advances in Engineering Software*, 38(6), 366–371. Retrieved from <https://www.sciencedirect.com/science/article/pii/S096599780600127X> doi: 10.1016/j.advengsoft.2006.09.007
- Filippi, J.-B. (n.d.). *ForeFire API: javascript interface for fire simulation integration in your system*. Retrieved from <http://forefire.univ-corse.fr/>
- Filippi, J. B. (2018). ForeFire open source wildfire front propagation model solver and programming interface. *CEUR Workshop Proceedings*, 2146(2004), 87–91.
- Finney, M. A. (1998). *FARSITE : Fire Area Development and Evaluation* (Vol. 3) (No. March). US Department of Agriculture, Forest Service, Rocky Mountain Research Station Ogden, UT.
- FlamMap — Fire, Fuel, and Smoke Science Program. (n.d.). Retrieved from <https://www.firelab.org/project/flammap>
- FRAMES. (n.d.). *BehavePlus*. Retrieved from <https://www.frames.gov/behaveplus/home>

- Ghisu, T., Arca, B., Pellizzaro, G., & Duce, P. (2015, 1). An Improved Cellular Automata for Wildfire Spread. *Procedia Computer Science*, 51, 2287–2296. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1877050915011965> doi: 10.1016/J.PROCS.2015.05.388
- Gouveia Freire, J., & Castro Dacamara, C. (2019). Using cellular automata to simulate wildfire propagation and to assist in fire management. *Natural Hazards and Earth System Sciences*, 19(1), 169–179. doi: 10.5194/nhess-19-169-2019
- Grasso, P., & Innocente, M. S. (2020). Physics-based model of wildfire propagation towards faster-than-real-time simulations. *Computers and Mathematics with Applications*, 80(5), 790–808. Retrieved from <https://doi.org/10.1016/j.camwa.2020.05.009> doi: 10.1016/j.camwa.2020.05.009
- Green, M. E., Deluca, T. F., & Kaiser, K. W. (2020). Modeling wildfire using evolutionary cellular automata. *GECCO 2020 - Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 1089–1097. doi: 10.1145/3377930.3389836
- Grishin, A. M. (1996). General mathematical model for forest fires and its applications. *Combustion, Explosion and Shock Waves*, 32(5), 503–519. doi: 10.1007/BF01998573
- Guelpa, E., Sciacovelli, A., Verda, V., & Ascoli, D. (2016). Faster prediction of wildfire behaviour by physical models through application of proper orthogonal decomposition. *International Journal of Wildland Fire*, 25(11), 1181. doi: 10.1071/wf15150
- Hansen, P. B. (1993, 8). Parallel cellular automata: A model program for computational science. *Concurrency: Practice and Experience*, 5(5), 425–448. Retrieved from <http://doi.wiley.com/10.1002/cpe.4330050504><https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.4330050504> doi: 10.1002/cpe.4330050504
- Iudin, D., Sergeyev, Y., & Hayakawa, M. (2015, 3). Infinity computations in cellular automaton forest-fire model. *Communications in Nonlinear Science and Numerical Simulation*, 20(3), 861–870. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1007570414002949?via%3Dihub> doi: 10.1016/J.CNSNS.2014.06.031
- Karafyllidis, I., & Thanailakis, A. (1997, 6). A model for predicting forest fire spreading using cellular automata. *Ecological Modelling*, 99(1), 87–97. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0304380096019424> doi: 10.1016/S0304-3800(96)01942-4
- Lei, W., & Zhiqiang, D. (2017). Modeling and simulation of forest fire detection and fire spread. In *2017 2nd international conference on advanced robotics and mechatronics, icarm 2017* (pp. 65–69). doi: 10.1109/ICARM.2017.8273136
- Levy, D., & Tadmor, E. (1998). From semidiscrete to fully discrete: Stability of Runge-Kutta schemes by the energy method. *SIAM Review*, 40(1), 40–73. doi: 10.1137/S0036144597316255
- Liu, Y., Liu, H., Zhou, Y., & Sun, C. (2018, 10). Spread vec-

- tor induced cellular automata model for real-time crown fire behavior simulation. *Environmental Modelling & Software*, 108, 14–39. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1364815217305613?via%3Dihub> doi: 10.1016/J.ENVSOFT.2018.07.005
- Lopes, A. M., Ribeiro, L. M., Viegas, D. X., & Raposo, J. R. (2017). Effect of two-way coupling on the calculation of forest fire spread: Model development. *International Journal of Wildland Fire*, 26(9), 829–843. doi: 10.1071/WF16045
- Mandel, J., Beezley, J. D., Coen, J. L., & Kim, M. (2009). Data assimilation for wildland fires. *IEEE Control Systems Magazine*, 29(3), 47–65. doi: 10.1109/MCS.2009.932224
- Mandel, J., Beezley, J. D., & Kochanski, A. K. (2011). Coupled atmosphere-wildland fire modeling with WRF 3.3 and SFIRE 2011. *Geoscientific Model Development*, 4(3), 591–610. doi: 10.5194/gmd-4-591-2011
- Mandel, J., Bennethum, L. S., Beezley, J. D., Coen, J. L., Douglas, C. C., Kim, M., & Vodacek, A. (2008, 12). A wildland fire model with data assimilation. *Mathematics and Computers in Simulation*, 79(3), 584–606. doi: 10.1016/j.matcom.2008.03.015
- Marzaeva, S. I., & Galtseva, O. V. (2018). A numerical solution of the problem of crown forest fire initiation and spread. *IOP Conference Series: Materials Science and Engineering*, 363(1). doi: 10.1088/1757-899X/363/1/012020
- Mattes, L., & Kofuji, S. T. (2014). An Efficiently Use of GPU to Implement High Accuracy FDTD Algorithms. *IEEE Latin America Transactions*, 12(4), 825–830. doi: 10.1109/TLA.2014.6868889
- McArthur, A. G. (1966). *Weather and Grassland Fire Behaviour*. Forest Research Institute, Forestry and Timber Bureau. Retrieved from <https://books.google.es/books?id=H1onHQAACAAJ>
- McArthur, A. G. (1967). *Fire Behaviour in Eucalypt Forests*. Forestry and Timber Bureau. Retrieved from <https://books.google.cl/books?id=Y032GwAACAAJ>
- McGrattan, K. B., Hostikka, S., Floyd, J., McDermott, R., & Vanella, M. (2019). *Fire Dynamics Simulator Technical Reference Guide Volume 1: Mathematical Model*. Gaithersburg, MD: National Institute of Standards and Technology. Retrieved from <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication1018.pdf> doi: 10.6028/NIST.SP.1018
- Mell, W., Charney, J., Jenkins, M. A., Cheney, P., & Gould, J. (2013). Numerical Simulations of Grassland Fire Behavior from the LANL-FIRETEC and NIST-WFDS Models. In J. J. Qu, W. T. Sommers, R. Yang, & A. R. Riebau (Eds.), *Remote sensing and modeling applications to wildland fires* (pp. 209–225). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-32530-4_15 doi: 10.1007/978-3-642-32530-4_{15}
- Montenegro, R., Plaza, A., Ferragut, L., & Asensio, M. I. (1997, 12). Application of a nonlinear evolution model to fire propagation. *Nonlinear Analysis, Theory*,

- Methods and Applications*, 30(5), 2873–2882. doi: 10.1016/S0362-546X(97)00341-6
- Mulyani, Putri, N. D., & Gunawan, P. H. (2017). The performance of OpenMP architecture for simulating fire spreading in forest area by cellular automata. *2017 5th International Conference on Information and Communication Technology, ICoICT 2017*, 0(c). doi: 10.1109/ICoICT.2017.8074661
- Mutthulakshmi, K., Wee, M. R. E., Wong, Y. C. K., Lai, J. W., Koh, J. M., Acharya, U. R., & Cheong, K. H. (2020). Simulating forest fire spread and fire-fighting using cellular automata. *Chinese Journal of Physics*, 65(April), 642–650. Retrieved from <https://doi.org/10.1016/j.cjph.2020.04.001> doi: 10.1016/j.cjph.2020.04.001
- National Center of Atmospheric Research. (n.d.). *Weather Research and Forecasting Model — MMM: Mesoscale & Microscale Meteorology Laboratory*. Retrieved from <https://www.mmm.ucar.edu/weather-research-and-forecasting-model>
- Nobile, M. S., Cazzaniga, P., Besozzi, D., Pescini, D., & Mauri, G. (2014, 3). cuTauLeaping: A GPU-powered tau-leaping stochastic simulator for massive parallel analyses of biological systems. *PLoS ONE*, 9(3). doi: 10.1371/journal.pone.0091963
- NVIDIA. (2018). *CUDA Zone — NVIDIA Developer*. Retrieved from <https://developer.nvidia.com/cuda-zone>
- NVIDIA. (2021). *CUDA Toolkit Documentation*. Retrieved from <https://docs.nvidia.com/cuda/index.html>
- Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science and Engineering*, 9(3), 10–20. doi: 10.1109/MCSE.2007.58
- Pastor, E., Zárate, L., Planas, E., & Arnaldos, J. (2003, 1). Mathematical models and calculation systems for the study of wildland fire behaviour. *Progress in Energy and Combustion Science*, 29(2), 139–153. doi: 10.1016/S0360-1285(03)00017-0
- Patankar, S. (2018). *Numerical heat transfer and fluid flow*. Taylor & Francis.
- Patankar, S. V., & Spalding, D. B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10), 1787–1806. doi: 10.1016/0017-9310(72)90054-3
- Peña-Fernández, E., & Valenzuela-Palma, L. (2004). Incremento de los Incendios Forestales en Bosques Naturales y Plantaciones Forestales en Chile. *Segundo Simposio Internacional Sobre Políticas, Planificación y Economía de los Programas de Protección Contra Incendios Forestales: Una Visión Global.*, 595–612.
- Perminov, V. (2017). Mathematical Modeling of Forest Fires Initiation, Spread and Impact on Environment. *International Journal of GEOMATE*, 13(35), 93–99. doi: 10.21660/2017.35.6704
- Perminov, V. (2018). Mathematical modelling of wildland fires initiation and

- spread using a coupled atmosphere-forest fire setting. *Chemical Engineering Transactions*, 70(2002), 1747–1752. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85051425579&doi=10.3303%2FCET1870292&partnerID=40&md5=6dfb8d74cb3f13c4312c695ee8937db5> doi: 10.3303/CET1870292
- Perminov, V., & Soprunenko, E. (2016). Numerical solution of crown forest fire initiation and spread problem. In *2016 11th international forum on strategic technology (ifost)* (pp. 400–404). doi: 10.1109/IFOST.2016.7884279
- Preisler, H. K., & Ager, A. A. (2014, 9). *Forest-Fire Models*. Wiley Online Library. Retrieved from <https://doi.org/10.1002/9781118445112.stat07705> doi: doi:10.1002/9781118445112.stat07705
- Quartieri, J., Mastorakis, N. N. E., Iannone, G., & Guarnaccia, C. (2010). *A Cellular Automata Model for Fire Spreading Prediction* (No. ISSN: 1792-4286). Retrieved from http://www.wseas.org/mastorakishttp://www.researchgate.net/publication/256662471_A_Cellular_Automata_Model_for_Fire_Spreading_Prediction/file/e0b495238e6df428e7.pdf
- Ramírez, J., Monedero, S., & Buckley, D. (2011). New approaches in fire simulations analysis with Wildfire Analyst. *South Africa*(May), 9–13. Retrieved from http://s3.amazonaws.com/academia.edu.documents/43690793/New_approaches_in_fire_simulations_analy20160313-6064-1mqwqkl.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1496060004&Signature=%2B31NP3MoD7bPYZOVv6RnhK1TDmY%3D&response-content-disposition=inline doi: 10.13140/2.1.2045.7766
- Reddy, S. C., & Trefethen, L. N. (1992). Stability of the method of lines. *Numerische Mathematik*, 267, 235–267. doi: 10.1007/BF01396228
- Richards, G. D. (1995). A General Mathematical Framework For Modelling Two-Dimensional Wildland Fire Spread. *International Journal of Wildland Fire*, 5(2), 63–72. doi: 10.1071/WF9950063
- Rothermel, R. C. (1972). *A mathematical model for predicting fire spread in wildland fuels* (Tech. Rep. No. INT-115). Retrieved from <http://www.srs.fs.usda.gov/pubs/32533> doi: <http://www.snap.uaf.edu/webshared/JenNorthway/AKFireModelingWorkshop/AKFireModelingWkshp/FSPProAnalysisGuideReferences/Rothermel1972INT-115.pdf>
- San Martín, D., & Torres, C. (2018b). *Ngen Kütral GPU: Open Source Framework for Chilean Wildfire Spreading GPU Implementation*. Retrieved from <https://github.com/dsanmartin/ngen-kutral-gpu>
- San Martín, D., & Torres, C. (2018c). *Ngen-Kutral: Open Source Framework for Chilean Wildfire Spreading*. Retrieved from <https://github.com/dsanmartin/ngen-kutral>
- San Martín, D., & Torres, C. E. (2018a). *dsanmartin/ngen-kutral: Open Source Framework for Chilean Wildfire Spreading*. Retrieved from <https://github.com/dsanmartin/ngen-kutral>
- San Martín, D., & Torres, C. E. (2018d). *Ngen-Kütral: Toward an Open Source*

- Framework for Chilean Wildfire Spreading. In *2018 37th international conference of the chilean computer science society (sccc)* (pp. 1–8). doi: 10.1109/SCCC.2018.8705159
- San Martín, D., & Torres, C. E. (2019, 11). Exploring a Spectral Numerical Algorithm for Solving a Wildfire Mathematical Model. In *2019 38th international conference of the chilean computer science society (sccc)* (pp. 1–7). doi: 10.1109/SCCC49216.2019.8966412
- Santoni, P. A., & Balbi, J. H. (1998, 10). Modelling of two-dimensional flame spread across a sloping fuel bed. *Fire Safety Journal*, 31(3), 201–225. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0379711298000113?via%3Dihub> doi: 10.1016/S0379-7112(98)00011-3
- Sarmin, E., & Chudov, L. (1963, 1). On the stability of the numerical integration of systems of ordinary differential equations arising in the use of the straight line method. *USSR Computational Mathematics and Mathematical Physics*, 3(6), 1537–1543. doi: 10.1016/0041-5553(63)90256-8
- Sauer, T. (2011). *Numerical Analysis* (2nd ed.). USA: Addison-Wesley Publishing Company.
- Tecnosylva. (n.d.). *Wildfire Analyst — Tecnosylva*. Retrieved from <https://tecnosylva.es/wildfire-analyst>
- Toivanen, J., Engel, C. B., Reeder, M. J., Lane, T. P., Davies, L., Webster, S., & Wales, S. (2019). Coupled Atmosphere-Fire Simulations of the Black Saturday Kilmore East Wildfires With the Unified Model. *Journal of Advances in Modeling Earth Systems*, 11(1), 210–230. doi: 10.1029/2017MS001245
- Trefethen, L. (2000). *Spectral Methods in MATLAB*. Society for Industrial and Applied Mathematics. doi: doi:10.1137/1.9780898719598
- Trucchia, A., D’andrea, M., Baghino, F., Fiorucci, P., Ferraris, L., Negro, D., ... Severino, M. (2020). Propagator: An operational cellular-automata based wildfire simulator. *Fire*, 3(3), 1–24. doi: 10.3390/fire3030026
- Úbeda, X., & Sarricolea, P. (2016, 11). Wildfires in Chile: A review. *Global and Planetary Change*, 146, 152–161. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0921818116302090> doi: 10.1016/j.gloplacha.2016.10.004
- Verwer, J. G., & Sanz-Serna, J. M. (1984). Convergence of method of lines approximations to partial differential equations. *Computing*, 33(3-4), 297–313. doi: 10.1007/BF02242274
- Wang, X., Chang, L., Liu, J., Qin, X., Ning, W., & Zhou, W. (2017). A cellular automata model for forest fire spreading simulation. *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*(71533001), 1–6. doi: 10.1109/SSCI.2016.7849971
- Weber, R. O. (1991a, 1). Modelling fire spread through fuel beds. *Progress in Energy and Combustion Science*, 17(1), 67–82. doi: 10.1016/0360-1285(91)90003-6
- Weber, R. O. (1991b). Toward a comprehensive wildfire spread model. *International Journal of Wildland Fire*, 1(4), 245–248. doi: 10.1071/WF9910245

References

- Weisstein, E. W. (2002). *Cellular Automaton*. Retrieved from <http://mathworld.wolfram.com/CellularAutomaton.html>
- Zafarullah, A. (1970). Application of the Method of Lines to Parabolic Partial Differential Equations With Error Estimates. *Journal of the ACM*, 17(2), 294–302. doi: 10.1145/321574.321583

Appendix A

Appendix

A.1 Mathematical Stuff

A.1.1 Reynolds theorem for an incompressible flow

In the context of thermal energy conservation detailed in subsection 3.1.3, Reynolds theorem states that

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = \int_V \frac{\partial}{\partial t} (\rho c T) dV + \int_S \rho c T \mathbf{u} \cdot \hat{\mathbf{n}} dS.$$

By *Divergence Theorem*,

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = \int_V \frac{\partial}{\partial t} (\rho c T) dV + \int_V \nabla \cdot (\rho c T \mathbf{u}) dV$$

Since c , ρ are assumed constants,

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = \int_V \rho c \frac{\partial T}{\partial t} dV + \int_V \rho c \nabla \cdot (T \mathbf{u}) dV \quad (\text{A.1})$$

By product rule for multiplication by scalar,

$$\nabla \cdot (T \mathbf{u}) = T \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla T.$$

Assuming that \mathbf{u} is an incompressible flow, that is $\nabla \cdot \mathbf{u} = 0$, then

$$\nabla \cdot (T \mathbf{u}) = \mathbf{u} \cdot \nabla T. \quad (\text{A.2})$$

Replacing (A.2) into (A.1),

$$\frac{d}{dt} \left(\int_V \rho c T dV \right) = \int_V \rho c \frac{\partial T}{\partial t} dV + \int_V \rho c \mathbf{u} \cdot \nabla T dV \quad (\text{A.3})$$

A.1.2 Taylor expansion for radiation term

For one dimension, the Taylor series expansion of $f(x + \delta)$ about x is

$$f(x + \delta) = f(x) + f'(x)\delta + \frac{f''(x)}{2!}\delta^2 + \dots$$

Truncating high order terms,

$$f(x + \delta) \approx f(x) + f'(x)\delta. \quad (\text{A.4})$$

In the context of radiation flux, let $f(x) = T^4(x)$, then,

$$f'(x) = 4T^3(x)\frac{dT(x)}{dx}, \quad (\text{A.5})$$

since T is assumed a non linear function. Replacing equation A.5 in (A.4),

$$\begin{aligned} T^4(x + \delta) &\approx T^4(x) + 4\delta T^3(x)\frac{dT(x)}{dx} \\ T^4(x + \delta) - T^4(x) &\approx 4\delta T^3(x)\frac{dT(x)}{dx} \end{aligned} \quad (\text{A.6})$$

To reach the expression (3.18), the expression $dT(x)/dx$ is extended to several variables, therefore the right hand side of (A.6) becomes to $4\delta T^3\nabla T$.

A.2 Kernel Memory experiment

The goal of this experiment is to show that the use of shared memory improves considerably the performance of a *GPUs* code. To do this, the experimental bandwidth of the shared memory use will be measure indirectly, using the example codes developed in Cheng, Grossman, and McKercher (2013).

A.2.1 Experiments

The experiments involve reading and writing into global and shared memory by the computation of a matrix transpose. Experiments are executed in the Guacolda-Leftraru cluster of the *National Laboratory of High Performance Computing (NLHPC)*. The *GPU* nodes have the *NVIDIA* Tesla V100 graphics card, whose memory description is detailed in the Table A.1.

Temporal variables studied are:

1. R_c := Coalesced read in global memory.
2. W_c := Coalesced write in global memory.
3. W_n := Non-coalesced write in global memory.

Table A.1: Tesla V100 memory description.

Memory	Size	Theoretical bandwidth
Global	16 [GB]	897 [GB/s]
Shared	96 [KB]	~ 12759 [GB/s]
Register	255 [units]	
Constant	64 [KB]	

4. RS := Read in shared memory.
5. WS := Write in shared memory.

Execution time are measured for 3 kinds of *kernels*:

- **copyGmem**: this *kernel* implements a coalesced copy and is used as a time lower bound. Time is computed by $R_c + W_c$.
- **naiveGmem**: this *kernel* implements a naive algorithm for transpose matrix. Time is computed by $R_c + W_n$.
- **transposeSmem**: this *kernel* implements a block transpose matrix using shared memory. Time is computed by $R_c + W_c + RW + WS$.

Matrix sizes used in experiments are 4096×4096 , 8192×8192 and 16384×16384 .

A.2.2 Times

Table A.2 presents the best times achieved for each *kernel*, using the different matrix sizes.

Table A.2: Times [ms] for each matrix size using different *kernels*.

<i>Kernel</i>	4096×4096	8192×8192	16384×16384
copyGmem	0.1772	0.6661	2.7764
naiveGmem	0.3776	2.6769	10.6870
transposeSmem	0.2053	0.7219	3.0085

A.2.3 Experimental bandwidths

Using the formula,

$$\text{Bandwidth}_{\text{effective}} = \frac{(\text{bytes read} + \text{bytes written}) \times 10^{-9}}{\text{time elapsed}} [\text{GB/s}],$$

the computed bandwidths using Table A.2, are detailed in Table A.3.

Note that the experimental values of global memory bandwidth of **copyGmem** and **transposeSmem** are close to the theoretical reported in Table A.1.

Table A.3: Experimental bandwidths [GB/s] computed for each matrix size.

<i>Kernel</i>	4096 × 4096	8192 × 8192	16384 × 16384
copyGmem	757.479	805.991	773.478
naiveGmem	355.487	355.487	200.944
transposeSmem	653.637	743.681	713.805

A.2.4 Analysis

Estimated shared memory bandwidth is derived from the following relationship,

$$\text{time}_{\text{total}} = \frac{\text{memory}_{\text{global}}}{\text{bandwidth}_{\text{global}}} + \frac{\text{memory}_{\text{shared}}}{\text{bandwidth}_{\text{shared}}}.$$

Using the fact that the amount of global and shared total memory used is the same, and rearranging the equation we obtain:

$$\text{bandwidth}_{\text{shared}} = \left(\frac{\text{total}_{\text{time}}}{\text{memory}} + \frac{1}{\text{bandwidth}_{\text{global}}} \right)^{-1} \quad (\text{A.7})$$

Table A.4 shows the estimated bandwidths using expression (A.7).

Table A.4: Estimated bandwidth [GB/s]

	4096 × 4096	8192 × 8192	16384 × 16384
Bandwidth	4767.948	9619.619	9252.407

The estimated values are closer enough to the theoretical shared memory reported in Table A.1.

A.2.5 Summary

Table A.5 presents the summary of NVIDIA V100 estimated bandwidth.

Table A.5: NVIDIA V100 summary.

Size	4096 × 4096	8192 × 8192	16384 × 16384
Threads	16 × 32	32 × 32	32 × 32
Blocks	256 × 128	256 × 256	512 × 512
Estimated SM bandwidth	~ 4.7 [TB/s]	~ 9.6 [TB/s]	~ 9.2 [TB/s]

A.2.6 Comments

- Medium-sized matrix gives a 80% of the theoretical shared memory performance indicated by the graphic card.

- Also, the medium matrix gets 90% of the theoretical global memory, indicated by the card.
- The results are sensitive to the block and thread configuration, so these parameters should be tuned depending on the size of the problem. Depending on the card, it is suggested to start with a certain number of threads per block and move to adjust the performance.
- This experiment shows that the use of shared memory can achieve better performance. It is very useful when there is a non-coalescent accesses in global memory and it is necessary to avoid this kind of access because it slows down the program execution time.

A.3 Figure Reuse Licenses

The following documents are the License Agreements for reuse of Figures 7.6 and 7.8 retrieved from Asensio and Ferragut (2002), and Mell et al. (2013) respectively.

APPENDIX A. APPENDIX



This is a License Agreement between Daniel San Martin ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	23-Jun-2021	Type of Use	Republish in a
Order License ID	1128135-1		thesis/dissertation
ISSN	1097-0207	Publisher Portion	JOHN WILEY & SONS LTD. Image/photo/illustration

LICENSED CONTENT

Publication Title	International journal for numerical methods in engineering	Publication Type	e-Journal
Article Title	On a wildland fire model with radiation	Start Page	137
Date	01/01/1969	End Page	157
Language	English	Issue	1
Country	United Kingdom of Great Britain and Northern Ireland	Volume	54
		URL	http://www3.interscience.wiley.com/cgi-bin/jhome/110541304
Rightsholder	John Wiley & Sons - Books		

REQUEST DETAILS

Portion Type	Image/photo/illustration	Distribution	Worldwide
Number of images / photos / illustrations	1	Translation	Original language of publication
Format (select all that apply)	Print, Electronic	Copies for the disabled?	No
Who will republish the content?	Academic institution	Minor editing privileges?	Yes
Duration of Use	Life of current edition	Incidental promotional use?	No
Lifetime Unit Quantity	Up to 499	Currency	USD
Rights Requested	Main product		

NEW WORK DETAILS

Title	Open Source Framework for Chilean Wildfire Spreading and Effects Analysis	Institution name	Universidad Técnica Federico Santa María
Instructor name	Claudio E. Torres	Expected presentation date	2021-07-01

ADDITIONAL DETAILS

The requesting person / organization to appear on the license Daniel San Martin

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Figure 4	Title of the article/chapter the portion is from	On a wildland fire model with radiation
Editor of portion(s)	Asensio, M. I.; Ferragut, L.	Author of portion(s)	Asensio, M. I.; Ferragut, L.
Volume of serial or monograph	54	Publication date of portion	2002-02-20
Page or page range of portion	137-157		

RIGHTSHOLDER TERMS AND CONDITIONS

No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto. You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley material. This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process. In no instance may the total amount of Wiley Materials used in any Main Product, Compilation or Collective work comprise more than 5% (if figures/tables) or 15% (if full articles/chapters) of the (entirety of the) Main Product, Compilation or Collective Work. Some titles may be available under an Open Access license. It is the Licensors' responsibility to identify the type of Open Access license on which the requested material was published, and comply fully with the terms of that license for the type of use specified. Further details can be found on Wiley Online Library <http://olabout.wiley.com/WileyCDA/Section/id-410895.html>.

CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.
2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.
3. Scope of License; Limitations and Obligations.
 - 3.1. All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.
 - 3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of

APPENDIX A. APPENDIX

the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

- 3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).
- 3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.
- 3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.
- 3.6. User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.
4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.
5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event,

the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.
7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.
8. Miscellaneous.
 - 8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.
 - 8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:<https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy>
 - 8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.
 - 8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.
 - 8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties

APPENDIX A. APPENDIX

expressly submit to the personal jurisdiction and venue of each such federal or state court. If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1



This is a License Agreement between Daniel San Martin ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	23-Jun-2021	Type of Use	Republish in a
Order License ID	1128148-1		thesis/dissertation
ISBN-13	9783642325298	Publisher Portion	Springer Berlin Heidelberg Image/photo/illustration

LICENSED CONTENT

Publication Title	Remote Sensing Modeling and Applications to Wildland Fires	Rightsholder	Springer
Article Title	Numerical Simulations of Grassland Fire Behavior from the LANL-FIRETEC and NIST-WFDS Models. 15	Publication Type	Book
Author/Editor	Qu, John J., EastFIRE Conference (1st : 2005 : Fairfax, Va.)	Start Page	209
Date	08/05/2013	End Page	225
Language	English	URL	http://link.springer.com/10.1007/978-3-642-32530-4
Country	Germany		

REQUEST DETAILS

Portion Type	Image/photo/illustration	Distribution	Worldwide
Number of images / photos / illustrations	1	Translation	Original language of publication
Format (select all that apply)	Print, Electronic	Copies for the disabled?	No
Who will republish the content?	Academic institution	Minor editing privileges?	No
Duration of Use	Life of current edition	Incidental promotional use?	No
Lifetime Unit Quantity	Up to 499	Currency	USD
Rights Requested	Main product		

NEW WORK DETAILS

Title	Open-source Framework for Chilean Wildfire Spreading and Effects Analysis	Institution name	Universidad Técnica Federico Santa María
Instructor name	Claudio E. Torres	Expected presentation date	2021-07-01

APPENDIX A. APPENDIX

ADDITIONAL DETAILS

Order reference number	N/A	The requesting person / organization to appear on the license	Daniel San Martin
-------------------------------	-----	--	-------------------

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Figure 15.4	Title of the article/chapter the portion is from	Numerical Simulations of Grassland Fire Behavior from the LANL-FIRETEC and NIST-WFDS Models. 15
Editor of portion(s)	Cheney, Phil; Mell, William; Jenkins, Mary Ann; Gould, Jim; Charney, Joseph	Author of portion(s)	Cheney, Phil; Mell, William; Jenkins, Mary Ann; Gould, Jim; Charney, Joseph
Volume of serial or monograph	N/A	Issue, if republishing an article from a serial	N/A
Page or page range of portion	209-225	Publication date of portion	2013-01-01

RIGHTSHOLDER TERMS AND CONDITIONS

A maximum of 10% of the content may be licensed for republication. The user is responsible for identifying and seeking separate licenses for any third party materials that are identified anywhere in the work. Without a separate license, such third party materials may not be reused.

If you are placing a request on behalf of/for a corporate organization, please use RightsLink. For further information visit <http://www.nature.com/reprints/permission-requests.html> and <https://www.springer.com/gp/rights-permissions/obtaining-permissions/882>. If the content you are requesting to reuse is under a CC-BY 4.0 licence (or previous version), you do not need to seek permission from Springer Nature for this reuse as long as you provide appropriate credit to the original publication. <https://creativecommons.org/licenses/by/4.0/>

CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.
2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.
3. Scope of License; Limitations and Obligations.
 - 3.1. All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set

forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

- 3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.
 - 3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).
 - 3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.
 - 3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.
 - 3.6. User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.
4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

APPENDIX A. APPENDIX

5. **Limitation of Liability.** UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.
6. **Limited Warranties.** THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.
7. **Effect of Breach.** Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.
8. **Miscellaneous.**
 - 8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.
 - 8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here: <https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy>
 - 8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.
 - 8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.
 - 8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts

of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court. If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1