# UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II 



Scuola Politecnica e delle Scienze di Base
Area Didattica di Scienze Matematiche Fisiche e Naturali
Corso di Laurea Magistrale in Matematica

Tesi Sperimentale in Statistica Matematica

# Beta Distribution in Wildfire Spreading 

Relatori<br>Prof.ssa Enrica Pirozzi<br>Dott. Gianni Pagnini

Marzia Canzaniello
N98000604

The research presented in this dissertation has been partially funded by BCAM - Basque Center for Applied Mathematics - Bilbao (Spain), where the candidate carried out a 6 months internship in the research line of Statistical Physics.

## Contents

Introduzione ..... 1
Introduction ..... 4
1 On some special function ..... 7
1.1 Historical overview ..... 7
1.1.1 The seventeenth century ..... 8
1.1.2 The eighteenth century ..... 8
1.1.3 The nineteenth century ..... 9
1.1.4 The twentieth century ..... 9
1.2 The Gamma function ..... 10
1.2.1 Introduction ..... 10
1.2.2 Definition ..... 11
1.2.3 Analyticity ..... 12
1.2.4 Properties ..... 12
1.2.5 Euler measures ..... 14
1.3 The Beta function ..... 14
1.3.1 Introduction ..... 14
1.3.2 The beta function of two variables ..... 15
1.3.3 The beta function of several variables ..... 16
1.3.4 Incomplete beta function ..... 17
1.4 Beta distribution ..... 18
1.4.1 Mean ..... 20
1.4.2 Variance ..... 21
1.4.3 Skewness ..... 21
1.4.4 Kurtosis ..... 22
1.5 Generalized logistic distribution ..... 23
2 PROPAGATOR ..... 25
A cellular-automata based wildfire simulator ..... 25
2.1 Introduction ..... 25
2.2 Mathematical modeling ..... 26
2.2.1 Cellular Automatas models ..... 26
2.3 History of the development ..... 27
2.4 PROPAGATOR model ..... 28
2.5 The role of the slope in wildfire propagation ..... 32
3 Results ..... 37
3.1 Introduction ..... 37
3.1.1 Modelling ..... 37
3.1.2 Analysis ..... 38
3.2 Slope 10 ..... 39
3.3 Slope 15 ..... 46
3.4 Slope20 ..... 52
3.5 Slope30 ..... 58
3.6 Slope40 ..... 64
3.7 Slope45 ..... 70
Conclusions ..... 76
Conclusioni ..... 78
A Basic on simulation ..... 80
A. 1 Wiener Process ..... 80
A.1.1 General facts ..... 80
A.1.2 Application ..... 84
A. 2 Ornstein-Uhlenbeck Process ..... 104
A.2.1 General facts ..... 104
A.2.2 Definition ..... 106
A.2.3 Simulation ..... 107
A. 3 Langevin Equation ..... 129
A.3.1 Fluctuation dissipation theorem ..... 131
A.3.2 Simulation ..... 132
B Algorithms ..... 146
C Software ..... 159
C. 1 Functions ..... 159
C. 2 Hypatia ..... 165
C. 3 Minitab ..... 166
Bibliography ..... 168

## Introduzione

Il presente elaborato è frutto del lavoro di ricerca svolto in collaborazione con il team di Statistical Physics del Basque Center for Applied Mathematics di Bilbao, superviosionato dal Dott. Gianni Pagnini.

Principale oggetto di studio è PROPAGATOR, algoritmo per la simulazione di incendi basato su un modello ad automi cellulari (CA).

La scelta di tale approccio modellistico non è casuale, gli automi cellulari, infatti, grazie alla loro natura modulare, sono in grado di semplificare i processi fisici che influenzano la propagazione di fuochi, pur conservando la possibilità di raggiungere qual si voglia livello di complessità e precisione. Inoltre essi costituiscono uno dei più conosciuti esempi di modelli stocastici a reticolo.
Il modello di PROPAGATOR, infatti, è basato su implementazione raster, che discretizza lo spazio in una griglia composta da celle rettangolari di lunghezza arbitraria, e la propagazione è modellizzata come un processo di contaminazione tra celle adiacenti del dominio considerato.

L'obiettivo è quello di individuare come si distribuisce l'area bruciata in un intervallo di osservazione limitato.

A tal fine, si considera un caso di propagazione di incendi semplificato. Non sono, infatti, oggetto di studio i combustibili e il possibile intervento di elicotteri antincendio, che possono condizionarne la dimensione, l'intensità e la durata.

Inoltre è escluso il fenomeno del fire-spotting: propagazione di incendi al di fuori del perimetro del fuoco principale causata da particelle ardenti che, sollevate in
aria dalle correnti convettive e sospinte dal vento, generano fuochi secondari con distanze dell'ordine di decine di metri.
Si focalizza dunque l'attenzione sui seguenti parametri: intervallo di osservazione; perimetro di propagazione; tipologia di vegetazione; intensità e direzione del vento; inclinazione del territorio.

Nel presente lavoro di tesi, in particolare, si riportano i risultati ottenuti studiando il fenomeno di propagazione al variare della pendenza del territorio e fissando i restanti parametri. Per ottenere tali risultati è stata apportata una modifica all'algoritmo di PROPAGATOR, in quanto quest'ultimo è programmato in modo da restituire in output, per ogni istante di tempo, la media aritmetica sul numero di realizzazioni dei valori di area bruciata, mentre per l'analisi si necessitava di tali valori per ogni realizzazione, in quanto interessati, non solo alla media, ma anche alla varianza, la skewness, la kurtosis e più in generale, appunto, alla distribuzione degli stessi. Questa accuratezza è costata in termini di sforzo computazionale. Per un elevato numero di realizzazioni, infatti, si è dovuto ricorrere all'utilizzo del server Hypatia.

L'elaborato è organizzato come segue:

- Nel primo capitolo, dopo una prima introduzione alle funzioni speciali e alla loro storia, con riferimenti a quelle maggiormente note e utilizzate, sono definite, a partire dalla funzione Beta e la funzione Gamma, la Distribuzione Beta e la Distribuzione Logistica Generalizzata.

Per approfondimenti: [7], [14], [20], [19] e [13];

- Nel secondo capitolo si introduce il modello di PROPAGATOR, la storia dello sviluppo dell'algoritmo, per poi arrivare al ruolo dell'inclinazione del territorio nella propagazione di incendi.

Per approfondimenti sul fenomeno: [4], [5], [23], [22], [21], [9] e [10].
Per approfondimenti sul modello: [18], [2] e [1];

- Nel terzo capitolo è descritta dettagliatamente l'analisi dati svolta e sono ri-
portati i risultati grafici per ogni caso di studio: slope $10^{\circ}$, slope $15^{\circ}$, slope $20^{\circ}$, slope $30^{\circ}$, slope $40^{\circ}$ e slope $45^{\circ}$;
- Nell'appendice A alcuni lavori preliminari sulla simulazione di processi stocastici noti, per sviluppare una sorta di senso critico nei confronti dei risultati, così da prepararsi all'uso di PROPAGATOR.
Per approfondimenti: [6], [12] e [17];
- Nell'appendice B sono riportati i codici elaborati per svolgere l'analisi;
- Nell'appendice C sono elencati i software, le app e le routine utilizzate.

Infine si rende noto che l'algoritmo di PROPAGATOR è attualmente in uso dal Dipartimento della Protezione Civile Nazionale e che la versione utilizzata nel seguito è quella del 2020, se pur già disponibile un suo aggiornamento al 2022.

## Introduction

This paper is the result of research work carried out in collaboration with the Statistical Physics team of the Basque Center for Applied Mathematics in Bilbao, supervised by Dr. Gianni Pagnini.

Main subject is PROPAGATOR, an algorithm for fires simulation based on a cellular automata model (CA).

The choice of such a modelling approach is not random, the cellular automata, in fact, thanks to their modular nature, are able to simplify the physical processes that influence the propagation of fires, while retaining the ability to achieve whatever you want level of complexity and accuracy. They are also one of the most widely known examples of stochastic lattice models.
The PROPAGATOR model, in fact, is based on raster implementation, which discretizes the space in a grid composed of rectangular cells of arbitrary length, and the propagation is modeled as a contamination process between adjacent cells of the considered domain.

The aim is to identify how the burned area is distributed in a limited observation interval.

For this purpose, a simplified case of fire propagation is considered. In fact, fuels and possible intervention of fire-fighting helicopters, which can affect fire size, intensity and duration, are not being studied.

In addition, the fire-spotting phenomenon is excluded. It consists in the propagation of fires outside the perimeter of the main fire caused by burning particles that,
raised in the air by convective currents and driven by the wind, generate secondary fires with distances of the order of tens meters.

The focus is therefore on the following parameters: observation interval; propagation perimeter; vegetation type; wind intensity and direction; inclination of the territory.

In this thesis, in particular, we report the results obtained by studying the phenomenon of propagation as the slope of the territory changes and fixing the remaining parameters. To obtain these results, a modification to the PROPAGATOR algorithm was made, in that the latter is programmed to return in output, for each instant of time, the arithmetic mean over the number of realizations of the burned area values, whereas for the analysis these values were needed for each realization, since they were interested, not only in the mean, but also in the variance, skewness, kurtosis and more generally in their distribution. This accuracy has cost in terms of computational effort. For a large number of realizations, in fact, it was necessary to use the Hypatia server.

The work is organized as follow:

- In the first chapter, after a first introduction to the special functions and their history, with references to those most known and used, are defined, starting from the Beta function and the Gamma function, the Beta Distribution and the General Logistics Distribution.

For further information: [7], [14], [20], [19] and [13];

- The second chapter introduces the PROPAGATOR model, the story of the algorithm development, and then arrive at the role of the territory inclination in the fires propagation.

For further information on the phenomenon: [4], [5], [23], [22], [21], [9] and [10]. For further information on the model: [18], [2] and [1];

- The third chapter describes in detail the data analysis carried out and gives the graphic results for each case studied: slope $10^{\circ}$, slope $15^{\circ}$, slope $20^{\circ}$,
slope $30^{\circ}$, slope $40^{\circ}$ e slope $45^{\circ}$;
- In Appendix A some preliminary work on the simulation of known stochastic processes, to develop a kind of critical sense for the results, in order to get prepared for the use of PROPAGATOR;
For further information: [6], [12] and [17];
- Appendix B shows the codes developed for the analysis;
- Appendix C lists the software, apps and routines used.

Finally, we inform that the PROPAGATOR algorithm is currently in use by the Department of National Civil Protection and that the version used in the following is of 2020, even if an update to 2022 is already available.

## Chapter 1

## On some special function

### 1.1 Historical overview

A special function is a real or complex valued function of one or more real or complex variables which is specified so completely that its numerical values could in principle be tabulated. Although this definition includes elementary functions such as $x^{n}, e^{x}, \log x$ and $\sin x[7]$.
The study of special function grew up with the calculus and is consequently one of the oldest branches of analysis. It flourished in the nineteenth century as part of the theory of complex variables, but lost much of its prestige in the first half of the twentieth when the study of general classes of function superseded that of individual functions. In the second half of the twentieth century it has received a new impetus from a connection with Lie groups, and a connection with averages of elementary functions.
A branch of mathematics which has many applications in other fields is likely to survive, and the subject of special functions owes much of its durability and growth to its usefulness in physical science. The history of special functions is closely tied to the problems of terrestrial and celestial mechanics that were solved in the eighteenth and early nineteenth centuries, the boundary-value problems of electromagnetism and heat in the nineteenth, and the eigenvalue problems of quantum mechanics in the twentieth.

In these cases the problem is to solve an ordinary or partial differential equation, but many special functions have been studied in order to evaluate various integrals not primarily connected with differential equations (for example, in the seventeenth and eighteenth centuries, integrals needed to find the arc length of an ellipse or hyperbola).

Special functions provide a valuable testing ground for analytical methods, for instance in the theory of complex variables or asymptotic expansions.
Finally, the great variety of special functions that have been studied and the enormous number of their known properties, expansions and interrelations raise intriguing and difficult questions: What relations are most fundamental for connections with other parts of mathematics? Can one find an underlying structure? How can the known results best be codified for practical use or numerical computation? These questions were not much considered before the twentieth century, and as yet there is no general agreement about how or in what sense they can be answered.
The subsequent sections list briefly some of the most important developments

### 1.1.1 The seventeenth century

England was the birthplace of special functions. John Wallis at Oxford took two first steps toward the theory of the gamma function long before Euler reached it. One was his formula for $\pi$, published in 1656 . The other step, more important because it later led Euler to a representation of the gamma function, was the idea of interpolating between numbers $a_{1}, a_{2}, a_{3}, \ldots$ by an integral which depends continuously on a parameter $v$ and has the value $a_{n}$ if $v=n$.

A more sophisticated calculus, which made possible the real flowering of special functions, was developed by Newton at Cambridge and by Leibnitz in Germany.

### 1.1.2 The eighteenth century

In 1703 James Bernoulli solved a differential equation by an infinite series which would now be called the series representation of a Bessel function. Al-
though Bessel functions were met by Euler and others in various mechanics problems and the principal achievements in the eighteenth century were the gamma function and the theory of elliptic integrals. In 1772 Euler evaluated the beta function integral in terms of the gamma function. Only the duplication and multiplication theorems remained to be discovered by Legendre and Gauss in the next century.

### 1.1.3 The nineteenth century

The golden age of special functions, which was centered in the nineteenth century Germany and France, was the result of developments in both mathematics and physics: the theory of analytic functions of a complex variable on one hand, and on the other hand, the field theories of physics (e.g. heat and electromagnetism) which required solutions of partial differential equations containing the Laplacian operator. The discovery of the elliptic functions (the inverses of the elliptic integrals) and their property of double periodicity was published by Abel in 1827. Elliptic functions grew up in symbiosis with the general theory of analytic functions and flourished throughout the nineteenth century, especially in the hands of Jacobi and Weierstrass.
Another major development was the theory of hypergeometric series, near the end of the century Appell introduced hypergeometric functions of two variables and Lauricella generalized them to several variables in 1893.
The gamma function was put in a new light in 1876 by Weierstrass' theory of entire functions, and then zeta function became celebrated because of Riemann's hypothesis and its connection with the prime number theorem.
At the end of the century special functions were familiar to every analyst and were part of the mathematics curriculum in the universities.

### 1.1.4 The twentieth century

During the first half of the century the growth of functional analysis drew the interest of pure mathematicians away from special functions. They were consid-
ered to be part of applied mathematics in 1950.
In physical science special functions gained added importance as solutions of the Schrödinger equation of quantum mechanics, but there were important developments of a purely mathematical nature also.

In 1907 Barnes used the gamma function to develop a new theory of Gauss' hypergeometric function. In 1922 the gamma function was approached from the new viewpoint of logarithmic convexity.
An important recent development has connected special functions and the matrix representations of Lie groups. It is too early to say whether any serious rapprochement between special functions and pure mathematics will result from this algebraic approach.

### 1.2 The Gamma function

### 1.2.1 Introduction

In 1729 Euler undertook the problem of interpolating $n$ ! between the positive integral values of $n$ in order to give a meaning to $x$ ! when $x$ is any positive number. Euler's solution, one of an infinite number of possibilities, consists in showing that

$$
\int_{0}^{\infty} t^{x} e^{-t} d t
$$

is well defined for all positive $x$ and has the value $n!$ if $x$ is a positive integer $n$. This integral (now called the gamma function) even gives a meaning to $x$ ! for many complex values of $x$.
Certain properties of Appell's symbol such as the reflection and duplication formulas correspond to analogous formulas for the gamma function. Conversely, Appell's symbol can be written as a ratio of gamma functions and its properties deduced from the ratio.

In the study of special function the gamma function is fundamental, practically because it is encountered in nearly all parts of the subject and theoretically because many special functions can be expressed in terms of gamma functions either directly or by contour integration. Although either property can be made into a
definition, the significance of the gamma function for applied mathematics comes chiefly from the practical importance of Euler's integral, and accordingly this integral will be the starting point of our discussion.

### 1.2.2 Definition

If $n$ is a positive integer, let

$$
f(n)=\int_{0}^{\infty} t^{n} e^{-t} d t
$$

integration by part shows that

$$
f(n)=\left[-t^{n} e^{-t}\right]_{0}^{\infty}+n \int_{0}^{\infty} t^{n-1} e^{-t} d t
$$

and so $f(n)=n f(n-1)$ if $n=2,3, \ldots$
Since $f(1)=1$, it follows that $f(n)=n(n-1) \cdots 3 \cdot 2 \cdot 1=n$ !. Thus $n$ ! has the integral representation

$$
\begin{equation*}
n!=\int_{0}^{\infty} t^{n} e^{-t} d t, \quad n=1,2,3, \ldots \tag{1.1}
\end{equation*}
$$

However, the integral converges also for non integral values of $n$ and even for complex values provided $\operatorname{Re} n>-1$. This suggests that we define the factorial function by

$$
\begin{equation*}
z!=\int_{0}^{\infty} t^{z} e^{-t} d t, \quad \operatorname{Re} z>-1 \tag{1.2}
\end{equation*}
$$

note that $0!=1$. The notation

$$
\begin{equation*}
\Gamma(z)=(z-1)! \tag{1.3}
\end{equation*}
$$

was introduced in 1809 by Legendre, who named the right-hand side of (1.2) the Eulerian integral of the second kind.

Definition 1.1. Let $\mathbb{C}_{>}=\{z \in \mathbb{C}: \operatorname{Re} z>0\}$. We define

$$
\begin{equation*}
\Gamma(z)=\int_{0}^{\infty} t^{z-1} e^{-t} d t, \quad z \in \mathbb{C}_{>} \tag{1.4}
\end{equation*}
$$

where $t^{z-1}$ means $\exp [(z-1) \log t]$ and $\log t$ has its real value.

### 1.2.3 Analyticity

A function is analytic on an open set in $\mathbb{C}$ if it has a derivative at every point of the set. When we wish to emphasize that a function is single-valued as well as analytic, we shall call it a holomorphic function.

Theorem 1.1. The function $\Gamma(z)$ is holomorphic on $U=\mathbb{C}-\{0,-1,-2, \ldots\}$. If $n \in \mathbb{N}$, it has a simple pole at $z=-n$ with residue

$$
\operatorname{Res}(\Gamma,-n)=\frac{(-1)^{n}}{n!}
$$

A function which is holomorphic on $\mathbb{C}$ except for poles is called meromorphic, and therefore the gamma function is meromorphic according to the theorem just proved.

A function $f$ is said to have a singularity at infinity if the function $g=f(1 / w)$ has a singularity at $w=0$. Now $\Gamma(1 / w)$ has simple poles at $w=-1,-1 / 2,-1 / 3, \ldots$ and hence $w=0$ is an accumulation point of poles. A singularity of this kind also is called essential, and therefore the gamma function is said to have an essential singularity at infinity.

### 1.2.4 Properties

The gamma function can be obtained from Appell's symbol by a limiting process. The connection with Appell's symbol leads to several of the most important properties of the gamma function.

- Let $\mathbb{R}_{>}=\{x \in \mathbb{R}: x>0\}$. If $x \in \mathbb{R}_{>}$and $\alpha+x, \beta+x \in U$, then

$$
\lim _{z \rightarrow \infty} \frac{\Gamma(\alpha+x)}{\Gamma(\beta+x)} x^{\beta-\alpha}=1
$$

- Euler's limit formula. Let $z \in U, n-1 \in \mathbb{N}$, and $(z, n)=z(z+1) \cdots(z+$ $n-1)$. Then

$$
\begin{equation*}
\Gamma(z)=\lim _{n \rightarrow \infty} \frac{(1, n)}{(z, n)} n^{z-1} \tag{1.5}
\end{equation*}
$$

- Euler's infinite product. If $z \in U$, then

$$
\begin{equation*}
\Gamma(z)=\frac{1}{z} \prod_{n=1}^{\infty} \frac{(1+(1 / n))^{z}}{1+(z / n)} \tag{1.6}
\end{equation*}
$$

- Logarithmic convexity. The gamma function is strictly log-convex on $\mathbb{R}_{>}$.

The property of log-convexity allows the gamma function to be characterized uniquely by its functional properties without the help of an explicit representation.

- Duplication theorem. If $2 z \in U$, then

$$
\Gamma(2 z)=\pi^{-1 / 2} 2^{2 z-1} \Gamma(z) \Gamma(z+1 / 2)
$$

- Stirling's formula. Given any $\varepsilon$ such that $0<\varepsilon \leq \pi$, let $z$ tend to infinity in the sector $|p h z| \leq \pi-\varepsilon$. Then

$$
\lim _{z \rightarrow \infty} \Gamma(z) z^{1 / 2-z} e^{z}=(2 \pi)^{1 / 2}
$$

Stirling's formula provides one method of calculating numerical values of the gamma function.

- Reflection formula. For every complex $z$,

$$
\sin \pi z=\frac{\pi}{\Gamma(z) \Gamma(1-z)}
$$

By means of the reflection formula, any circular, hyperbolic, or exponential function can be expressed in terms of gamma functions.

## - Inequalities:

(1) $\Gamma(x)>\Gamma\left(x_{0}\right) \quad x \neq x_{0}$
(2) $\Gamma(x)>\Gamma(y)$ if $x<y \leq x_{0}$
(3) $\Gamma(x)<\Gamma(y)$ if $x_{0} \leq x<y$
(4) $\left|\frac{\Gamma(x+i y)}{\Gamma(x)}\right|<\left(\frac{x^{2}}{x^{2}+y^{2}}\right)^{\frac{1}{2}}<1$ if $x, y \in \mathbb{R}$ and $y \neq 0$
(5) $|\Gamma(x+i y)| \geq \Gamma(x)(\operatorname{sech} \pi y)^{\frac{1}{2}}>\Gamma(x) e^{-\pi|y| / 2} \quad$ if $x, y \in \mathbb{R}, x \geq \frac{1}{2}, y \neq 0$ the first inequality being strict if $x>1 / 2$.

### 1.2.5 Euler measures

If $z$ is real and positive, the integrand of Euler's integral is known in statistics as the frequency function of the gamma distribution. Division by $\Gamma(z)$ normalizes the distribution to unit total probability. Even in the complex case the same procedure yields a complex measure on the positive real line which is normalized to unit total measure.

Definition 1.2. Let $\alpha \in \mathbb{C}_{>}$. The complex measure $\lambda_{\alpha}$ defined on $\mathbb{R}_{+}$by

$$
\begin{equation*}
d \lambda_{\alpha}(t)=\frac{1}{\Gamma(\alpha)} t^{\alpha-1} e^{-t} d t \tag{1.7}
\end{equation*}
$$

will be called an Euler measure.
Note that

$$
\lambda\left(\mathbb{R}_{+}\right)=\int_{0}^{\infty} d \lambda_{\alpha}(t)=1
$$

The moments of $\lambda_{\alpha}$ are given by

$$
\begin{aligned}
& t^{m} \lambda_{\alpha}(t)=(\alpha, m) d \lambda_{\alpha+m}(t) \quad m \in \mathbb{N} \\
& \int_{0}^{\infty} t^{m} d \lambda_{\alpha}(t)=(\alpha, m)
\end{aligned}
$$

### 1.3 The Beta function

### 1.3.1 Introduction

A particular combination of gamma functions is given a name because it has a simple and useful integral representation. The beta function, or Eulerian integral of the first kind, is

$$
\begin{equation*}
B(x, y)=\int_{0}^{1} u^{x-1}(1-u)^{y-1} d u \tag{1.8}
\end{equation*}
$$

where $x$ and $y$ have positive real parts [7].
With various changes of integration variable it is seen frequently in applied mathematics, for example in the theory of high-energy particle physics.
If $x$ and $y$ are real and positive, the integrand (1.8) occurs in statistics as the frequency function of the beta distribution. To normalize the distribution to unit total
probability, we divide by $B(x, y)$.
Complex values of $x$ and $y$ hold little interest for statistics, but the integrand still defines a complex measure on the unit interval which can be normalized as done previously:

$$
\begin{gathered}
d \mu(u)=\frac{u^{x-1}(1-u)^{y-1}}{B(x, y)} d u \\
\int_{0}^{1} d \mu(u)=1 \quad x, y \in \mathbb{C}_{>}
\end{gathered}
$$

The beta distribution for more than one random variable can likewise be extended to complex values of the parameters. For exemple we may define the complex measure

$$
\begin{align*}
& d \mu(u, v)=\frac{u^{x-1} v^{y-1}(1-u-v)^{z-1}}{B(x, y, z)} d u d v  \tag{1.9}\\
& \int d \mu(u, v)=1 \quad x, y, z \in \mathbb{C}_{>}
\end{align*}
$$

where the triangular region of the integration is $\{(u, v): u \geq 0, v \geq 0, u+v \leq 1\}$. Equations (1.9) determine $B(x, y, z)$, which we shall call the beta function of three variables.

Because Dirichlet (1839) evaluated an integral of this type in several variables, we shall call the complex measure, or its analog in any number of variables, a Dirichlet measure. Dirichlet measures will be fundamental in defining special functions.

### 1.3.2 The beta function of two variables

To avoid restricting the real parts of the variables, we define the beta function in terms of gamma functions

Definition 1.3. Let $x, y \in U$. Then beta function is defined by

$$
\begin{equation*}
B(x, y)=\frac{\Gamma(x) \Gamma(y)}{\Gamma(x+y)} \tag{1.10}
\end{equation*}
$$

Representation 1.1. If $x, y \in \mathbb{C}_{>}$, then

$$
\begin{aligned}
& B(x, y)=\int_{0}^{1} u^{x-1}(1-u)^{y-1} d u, \\
& B(x, y)=2 \int_{0}^{\pi / 2}(\cos \Theta)^{2 x-1}(\sin \Theta)^{2 y-1} d \Theta, \\
& B(x, y)=\int_{0}^{\infty} \frac{t^{x-1}}{(t+1)^{x+y}} d t .
\end{aligned}
$$

The symmetry property

$$
B(x, y)=B(y, x)
$$

is obvious from (1.10). For the integral representation it means that there is no essential distinction between the two ends of the interval of integration. Symmetries of similar kind will be important in subsequent sections.

### 1.3.3 The beta function of several variables

The definition and integral representation of the beta function will now be extended to three or more variables.

Definition 1.4. Let $\beta=\left(\beta_{1}, \ldots, \beta_{k}\right) \in U^{k}, k \geq 2$. The beta function of $k$ variables is defined by

$$
\begin{equation*}
B(\beta)=B\left(\beta_{1}, \ldots, \beta_{k}\right)=\frac{\Gamma\left(\beta_{1}\right) \cdots \Gamma\left(\beta_{k}\right)}{\Gamma\left(\beta_{1}+\cdots+\beta_{k}\right)} \tag{1.11}
\end{equation*}
$$

For $k=2$ the ordinary beta function is represented by an integral (1.8) over the unit interval $0 \leq u \leq 1$.
If $k=3$, we integrate over a triangular region in the $(u, v)$ plane with vertices $(0,0),(1,0)$ and $(0,1)$. This is the set $\{(u, v): u \geq 0, v \geq 0, u+v \leq 1\}$ and will be called the standard simplex in $\mathbb{R}^{2}$. The points $(u, v)$ of the simplex are in one-toone correspondence with the triples ( $u, v, 1-u-v$ ) of non negative weights with unit sum.
In general we denote the standard simplex in $\mathbb{R}^{n}, n \geq 1$, by

$$
E=E_{n}=\left\{\left(u_{1}, \ldots, u_{n}\right): u_{1} \geq 0, \ldots, u_{n} \geq 0, u_{1}+\cdots+u_{n} \leq 1\right\}
$$

The points $\left(u_{1}, \ldots, u_{n}\right)$ of the simplex are in one-to-one correspondence with the


Figure 1.1: The standard simplex $E_{n}$ [7].
( $n+1$ )-tuples $\left(u_{1}, \ldots, u_{n}, 1-u_{1}-\cdots-u_{n}\right)$ of non negative weights with unit sum. The interior of $E$ is denoted by

$$
\operatorname{int}(E)=\left\{\left(u_{1}, \ldots, u_{n}\right): u_{1}>0, \ldots, u_{n}>0, u_{1}+\cdots+u_{n}<1\right\}
$$

According to (1.11), $B\left(\beta_{1}, \ldots, \beta_{k}\right)$ is symmetric by $\beta_{1}, \ldots, \beta_{k}$.

### 1.3.4 Incomplete beta function

Definition 1.5. The incomplete beta function, a generalization of the beta function, is defined as

$$
\begin{equation*}
B(x ; \alpha, \beta)=\int_{0}^{x} u^{\alpha-1}(1-u)^{\beta-1} d u \tag{1.12}
\end{equation*}
$$

for $x=1$ the incomplete beta function coincides with the complete one.
The regularized incomplete beta function is defined in terms of the incomplete beta function and the complete one:

$$
\begin{equation*}
I_{x}(\alpha, \beta)=\frac{B(x ; \alpha, \beta)}{B(\alpha, \beta)} \tag{1.13}
\end{equation*}
$$

The regularized incomplete beta function is the cumulative distribution function of the beta distribution, and is related to the cumulative distribution function $F(k ; n, p)$ of a random variable $X$ following a binomial distribution with probability of single success $p$ and number of Bernoulli trials $n$ :

$$
\begin{equation*}
F(k ; n, p)=P(X \leq k)=I_{1-p}(n-k, k+1)=1-I_{p}(k+1, n-k) \tag{1.14}
\end{equation*}
$$

### 1.4 Beta distribution

The beta distribution is a family of continuous probability distributions defined on the interval $[0,1]$ parameterized by two positive shape parameters, denoted by $\alpha$ and $\beta$, that appear as exponents of the random variable and control the shape of the distribution. The generalization to multiple variables is called a Dirichlet distribution. The formulation of the beta distribution discussed below is also known as the beta distribution of the first kind, whereas beta distribution of the second kind is an alternative name for the beta prime distribution.

Beta distributions are very versatile and a variety of uncertainties can be usefully modeled by them. Many of the finite range distributions encountered in practice can be easily transformed into the standard distributions.
The probability density function (pdf) of the standard beta distribution is given by:

$$
\begin{equation*}
f(x ; \alpha, \beta)=\frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{1.15}
\end{equation*}
$$

for $0 \leq x \leq 1$, where $\alpha>0, \beta>0, B(\alpha, \beta)$ denotes the beta function and $x$ is a realization of a random process $X$ [16].

This definition includes both ends $x=0$ and $x=1$, which is consistent with definitions for other continuous distributions supported on a bounded interval which are special cases of the beta distribution, for example the arcsine distribution. However, the inclusion of $x=0$ and $x=1$ does not work for $\alpha, \beta<1$; accordingly, several authors, choose to exclude the ends $x=0$ and $x=1$, (so that
the two ends are not actually part of the domain of the density function) and consider instead $0<x<1$.

Several authors use the symbols $p$ and $q$, instead of $\alpha$ and $\beta$, for the shape parameters of the beta distribution, reminiscent of the symbols traditionally used for the parameters of the Bernoulli distribution, because the beta distribution approaches the Bernoulli distribution in the limit when both shape parameters approach the value of zero.

The parameters $\alpha$ and $\beta$ are symmetrically related by

$$
f(x ; \alpha, \beta)=f(1-x ; \beta, \alpha)
$$

this implies that if $X$ has the beta distribution with parameters $\alpha$ and $\beta$ then $1-X$ has the beta distribution with parameters $\beta$ and $\alpha$. The pdf (1.15) corresponds to type I distribution in the system of Pearson curves. The special case of (1.15) for $\alpha=\beta=1$ is the uniform distribution, when $\beta=1$ the beta distribution is known as the power function distribution.

The cumulative distribution function (cdf) of (1.15) is:

$$
\begin{equation*}
F(x ; \alpha, \beta)=\frac{B(x ; \alpha, \beta}{B(\alpha, \beta)}=I_{x}(\alpha, \beta) \tag{1.16}
\end{equation*}
$$

where $B(x ; \alpha, \beta)$ is the incomplete beta function and $I_{x}(\alpha, \beta)$ is the regularized incomplete beta function.

The $\boldsymbol{n}$ th moment about zero associated with (1.15) is [3]

$$
\begin{equation*}
E\left(X^{n}\right)=\frac{B(\alpha+n, \beta)}{B(\alpha, \beta)} \tag{1.17}
\end{equation*}
$$

- $\alpha>1$ and $\beta>1$ : the pdf (1.15) has a single mode at $x=(\alpha-1) /(\alpha+\beta-$ 2);
- $\alpha<1$ and $\beta<1$ : there is an anti-mode at this same value of $x$, and this corresponds to a $U$-shaped beta distribution;
- $(\alpha-1)(\beta-1) \leq 0$ : the pdf does not have a mode or anti-mode.

The aim is to discuss the properties of order statistics and record values associated with (1.15), to provide a review of the known variations of (1.15), to discuss the relationship of (1.15) to other well-known distributions and to illustrate real life applications

### 1.4.1 Mean

The expected value (mean) $\mu$ of a Beta distribution random variable $X$ with two parameters $\alpha$ and $\beta$ is a function of only the ratio $\beta / \alpha$ of these parameters:

$$
\begin{align*}
\mu=E[X] & =\int_{0}^{1} x f(x ; \alpha, \beta) d x= \\
& =\int_{0}^{1} x \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} d x  \tag{1.18}\\
& =\frac{\alpha}{\alpha+\beta}=\frac{1}{1+(\beta / \alpha)}
\end{align*}
$$

Letting $\alpha=\beta$ in the above expression one obtains $\mu=1 / 2$, showing that for $\alpha=\beta$ the mean is at the center of the distribution: it is symmetric.
Also, the following limits can be obtained from the above expression:

$$
\begin{aligned}
& \lim _{\frac{\beta}{\alpha} \rightarrow 0} \mu=1 \\
& \lim _{\frac{\beta}{\alpha} \rightarrow \infty} \mu=0
\end{aligned}
$$

Therefore, for $\beta / \alpha \rightarrow 0$ or for $\alpha / \beta \rightarrow \infty$, the mean is located at the right end $x=1$. For these limit ratios, the beta distribution becomes a one-point degenerate distribution with a Dirac delta function spike at the right end, $x=1$, with probability 1 , and zero probability everywhere else. There is $100 \%$ probability (absolute certainty) concentrated at the right end, $x=1$.
Similarly, for $\beta / \alpha \rightarrow \infty$ or for $\alpha / \beta \rightarrow 0$, the mean is located at the left end $x=0$. The beta distribution becomes a one-point Degenerate distribution with a Dirac delta function spike at the left end, $x=0$, with probability 1 , and zero probability everywhere else. There is $100 \%$ probability (absolute certainty) concentrated at the left end, $x=0$.

### 1.4.2 Variance

The variance, second moment centered on the mean, of a Beta distribution random variable $X$ with parameters $a$ and $b$ is

$$
\begin{equation*}
\operatorname{var}(X)=E\left[(X-\mu)^{2}\right]=\frac{\alpha \beta}{(\alpha+\beta)^{2}(\alpha+\beta+1)} \tag{1.19}
\end{equation*}
$$

letting $\alpha=\beta$ in the above expression one obtains

$$
\operatorname{var}(X)=\frac{1}{4(2 \beta+1)}
$$

showing that for $\alpha=\beta$ the variance decreases monotonically as $\alpha=\beta$ increase. Setting $\alpha=\beta=0$ in this expression, one finds the maximum variance $\operatorname{var}(X)=1 / 4$ which only occurs approaching the limit, at $\alpha=\beta=0$.
The beta distribution may also be parametrized in terms of its mean $0<\mu<1$ and sample size $v=\alpha+\beta>0$ :

$$
\alpha=\mu v, \quad \beta=(1-\mu) v
$$

using this parametrization, one can express the variance in terms of the mean $\mu$ and the sample size $v$ as follows:

$$
\operatorname{var}(X)=\frac{\mu(1-\mu)}{1+v}
$$

it follows that $\operatorname{var}(X)<\mu(1-\mu)$.
For a symmetric distribution, the mean is at the middle of the distribution, $\mu=$ $1 / 2$, and therefore:

$$
\operatorname{var}(X)=\frac{1}{4(1+v)} \quad \text { if } \mu=\frac{1}{2}
$$

### 1.4.3 Skewness

The skewness, third moment centered on the mean, normalized by the $3 / 2$ power of the variance, of (1.15) is

$$
\begin{equation*}
\gamma_{1}=\frac{E\left[(X-\mu)^{3}\right]}{(\operatorname{var}(X))^{3 / 2}}=\frac{2(\beta-\alpha)}{\alpha+\beta+2} \sqrt{\frac{\alpha+\beta+1}{\alpha \beta}} \tag{1.20}
\end{equation*}
$$

If $\alpha=\beta$ then $\gamma_{1}=0$, showing once again that the distribution is symmetric.
If $\beta>\alpha$ then $\gamma_{1}>0$ and the pdf becomes skewed to the right. Similarly, $\beta<\alpha$ gives a left skewed pdf.
In term of mean $\mu$ and the sample size $v=(\alpha+\beta)>0$, using the parametrization

$$
\alpha=\mu v \quad \beta=(1-\mu) v
$$

one can express the skewness

$$
\begin{equation*}
\gamma_{1}=\frac{E\left[(X-\mu)^{3}\right]}{(\operatorname{var}(X))^{3 / 2}}=\frac{2(1-2 \mu)}{2+v} \sqrt{\frac{1+v}{\mu(1-\mu)}} \tag{1.21}
\end{equation*}
$$

The skewness can also be expressed just in terms of the variance var and the mean $\mu$ as follows

$$
\begin{equation*}
\gamma_{1}=\frac{E\left[(X-\mu)^{3}\right]}{(\operatorname{var}(X))^{3 / 2}}=\frac{2(1-2 \mu) \sqrt{\operatorname{var}}}{\mu(1-\mu)+\operatorname{var}} \tag{1.22}
\end{equation*}
$$

if $\operatorname{var}<\mu(1-\mu)$.
The accompanying plot of skewness as a function of variance and mean shows that maximum variance ( $1 / 4$ ) is coupled with zero skewness and the symmetry condition ( $\mu=1 / 2$ ), and that maximum skewness (positive or negative infinity) occurs when the mean is located at one end or the other, so that the "mass" of the probability distribution is concentrated at the ends (minimum variance).

### 1.4.4 Kurtosis

The kurtotis, fourth moment centered on the mean normalized by the square of the variance, of (1.15) is

$$
\begin{align*}
\gamma_{2} & =\frac{E\left[(X-\mu)^{4}\right]}{(\operatorname{var}(X))^{2}}=  \tag{1.23}\\
& =\frac{3(\alpha+\beta+1)}{\alpha \beta(\alpha+\beta+2)(\alpha+\beta+3)}\left[2(\alpha+\beta)^{2}+\alpha \beta(\alpha+\beta-6)\right]
\end{align*}
$$

The description of kurtosis as a measure of the "extreme values" of the probability distribution, is correct for all distributions including the beta distribution.

When rare, extreme values can occur in the beta distribution, the higher its kurtosis; otherwise, the kurtosis is lower.
Minimum kurtosis takes place when the mass density is concentrated equally at each end (and therefore the mean is at the center), and there is no probability mass density in between the ends.

Note that both the skewness and the kurtosis are symmetrical functions of $a$ and $b$ are symmetrical functions of $a$ and $b$; so, interchanging the parameters in the pdf yields its mirror image.

### 1.5 Generalized logistic distribution

The aim of this chapter is to compare the beta distribution with the genlogistic ditribution. For this purpose we introduce it theoretically.

Because of their flexibility, recently, much attention has been given to the study of generalized distributions [13].

Verhulst used the logistic function for economic demographic purposes. Gumbel found that the logistic distribution arises in a purely statistical manner as the limiting distribution (as $n \rightarrow \infty$ ) of the standardized midrange (average of largest and smallest values) of random samples of size $n$ from a symmetric distribution of exponential type. Gumbel and Keeney showed that a logistic distribution is obtained as the limiting distribution of an appropriate multiple of the "extremal quotient", that is, (largest value)/(smallest value). Talacko proved that the logistic distribution is the limiting distribution (as $r \rightarrow \infty$ ) of the standardized variable corresponding to $\sum_{j=1}^{r} j^{-1} X_{j}$, where $X_{j}$ are independent random variables each having a type I extreme value distribution.

A number of authors discussed important applications of the logistic distribution in many fields including survival analysis, growth model and public health. Several different forms of generalizations of the logistic distribution have been
proposed in the literature, i.e. types I, II, III and IV.

The type I generalized logistic distribution has the following pdf

$$
\begin{equation*}
f(x ; \alpha)=\frac{\alpha e^{-x}}{\left(1+e^{-x}\right)^{\alpha+1}}, \quad \alpha>0 \tag{1.24}
\end{equation*}
$$

this type has also been called the skew-logistic distribution.
If $X$ has type I generalized logistic distribution in (1.24), then $X$ has a type II generalized logistic distribution.

The type III generalized logistic distribution has the pdf

$$
\begin{equation*}
f(x ; \alpha)=\frac{1}{B(\alpha, \alpha)} \frac{e^{-\alpha x}}{\left(1+e^{-x}\right)^{2 \alpha}}, \quad \alpha>0,-\infty<x<+\infty \tag{1.25}
\end{equation*}
$$

Last but not least, the type IV beta generalized logistic distribution is given by the pdf

$$
\begin{equation*}
f(x ; \alpha, \beta)=\frac{1}{B(\alpha, \beta)} \frac{e^{-\beta x}}{\left(1+e^{-x}\right)^{\alpha+\beta}}, \quad \alpha, \beta>0 \tag{1.26}
\end{equation*}
$$

where $B(\alpha, \beta)$ is the complete beta function.

Type IV is the most general form of the distribution: the Type III distribution can be obtained from Type IV by fixing $\beta=\alpha$; the Type II distribution can be obtained from Type IV by fixing $\alpha=1$ (and renaming $\beta$ to $\alpha$ ); the Type I distribution can be obtained from Type IV by fixing $\beta=1$.

It is well known, in general, that a generalized model is more flexible than the ordinary model and it is preferred by many data analysts in analyzing statistical data. Here, we will be concerned mostly with the beta generalized logistic distribution type I, i.e. the skew-logistic distribution.

## Chapter 2

## PROPAGATOR

## A cellular-automata based wildfire simulator

### 2.1 Introduction

Wildfire emergencies, especially in the southern EU Countries, are related with extreme weather conditions, characterized by persistent dry strong winds over flammable land cover species. In this case, the ignition probability increases and, in case it happens, the fire propagation is rapid and difficult to cope with: in most of the recent wildfire emergencies, casualties happened in few hours after the fire ignition. For this reason, it is extremely urgent to support first responders and Civil Protection Authorities (CPAs) with operational tools in emergency response, based on reliable wildfire risk maps and efficient emergency plans. This behavior requires CPAs to improve their ability of anticipation, discrimination, and selection of the best strategies and the most appropriate decisions in the first phase of the event, in order to ensure security to exposed people.
New technologies and computer modeling represent a great opportunity, supporting wildfire emergency managers sharing information useful in the coordination of civil protection and fire fighting activities. The recent dramatic events occurred in Greece and Portugal made evident the need of tools able to anticipate the behavior of fire in order to implement prevention and communication activities in
time to save lives. This can be achieved using ad hoc mathematical and numerical models.

### 2.2 Mathematical modeling

Physical processes influencing wildfire propagation are complex, meaning that the effects of slopes, wind conditions and fuel moisture interconnect and combine together, determining the evolution of the fire event. Such factors make wildfires multi-scale, multi-physics and nonlinear phenomena. This makes the formulation of efficient and reliable mathematical models particularly hard, as well as their computational implementation.
Nevertheless, in literature, there are many different approaches and models dedicated to this specific task. Such modeling efforts are usually divided into three main approaches [1]:

1. empirical and semi-empirical models, which rely on statistically derived laws of fire propagation;
2. macroscopic-deterministic models, where the fire spread is modeled in a continuum;
3. stochastic lattice or grid-based models, where the evolving quantities are usually described adopting a discretization in space and time [1],[11].

In any case, it should be remembered that the distinction between such categories may not be strict as expected, since, in many works, different approaches are mixed together.

### 2.2.1 Cellular Automatas models

Cellular Automatas (CA) [8] constitute one of the most well known examples of the latter category of models. CA models for wildfire simulation model
discretize spatial interactions by adopting a square or hexagonal grid. The macroscopic fire spread dynamics is simulated by the means of an ensemble of different realization of a stochastic process. In every realization, the spreading of the fire front from burning cells to neighboring ones is modeled by the means of probabilistic rules. Although CA models may simplify the underlying physical processes, their modular nature allows them to reach the desired level of complexity and accuracy.

### 2.3 History of the development

The implementation of PROPAGATOR has spanned across more than a decade, and it is recapitulated in the following [22]

- The first implementation of PROPAGATOR started from a request of the Italian Civil Protection Department, to support the organization of the G8 summit 2009, originally planned in La Maddalena, Sardinia, a region frequently affected by severe forest fires in summer season in order to evaluate the best prevention measures and support the fire fighting activities in case of a forest fire event.

Since its first release, it was able to reproduce burned areas up to 10,000 ha in a few minutes of computational time. Given an ignition point, it highlighted the zones more likely to be affected by fire propagation.

- In 2011, the second release of PROPAGATOR was operational. In this release, the model has been implemented in a 3D environment named NAZCA. While in the previous version algorithm and server code were mixed, at this stage, the algorithm was running as a standalone MATLAB ${ }^{\circledR}$ script, much easier to maintain and to develop.
Timing algorithms were added at this stage, taking into account wind and orography. Probability maps were at last time dependent and isochrones are added as visual output. At this stage, PROPAGATOR did not include a real parametrization of the propagation speed.
- In 2014, the third release was completed. The web interface was redesigned and it was integrated into the multi purpose MyDewetra platform, a tool for the forecasting, monitoring and real-time surveillance of all the environmental risks (http://www.mydewetra.org).
- In 2017, the fourth release saw a total rewriting of the code in the Python programming language, with a new server. Some of the algorithms for the treatment of slope and wind data have been rewritten from scratch, and it has been made possible to change wind conditions over time. An open API had been released to several developers during the ANYWHERE (EnhANcing emergencY management and response to extreme WeatHER and climate Events) European Project and the fuel-DEM dataset had been extended from the sole Italian territory to Finland, Portugal, Spain (Catalonia, Cantabria, Asturias) France (Corsica and Cote D'Azur) and Switzerland.
- In 2020, the fifth release of PROPAGATOR saw the implementation of a Rate of Spread (RoS) model in order to give the isochrones a more realistic time parametrization, and the introduction of the fuel moisture into the computational core. The 2020 version also saw the introduction of fire fighting actions (lines and polygons where some kind of fire fighting procedure is going to be put in charge) that may be prescribed by the user in a time-dependent way.


### 2.4 PROPAGATOR model

The PROPAGATOR model is a quasi-empirical stochastic CA model based on a raster implementation, which discretizes the space into a grid composed of square cells of arbitrary length $\Delta x=\Delta y=L$. The cell size reflects the resolution in space of the analysis and the final results. In this work, $L$ has been fixed to 20 Km , allowing PROPAGATOR to give high resolution output [22].

For each time step, each cell of the domain can assume one out of three different
possible states:

- State 1: corresponds to cells that are burning during the current simulation step;
- State 0: corresponds to cells that are already burned in previous steps of the simulation;
- State -1: corresponds to cells that are unburned, but that can burn in the following steps of the simulation.

The fire propagation is modeled as a contamination process between adjacent cells of the considered domain; the probability of fire spreading from a cell to one of its neighborhood, $p_{i j}$, is calculated starting from the nominal fire spread probability (named $p_{n}$ ), which is then modified considering several factors. Such factors account for the topography, wind vector, and the fuel moisture content.

For each cell of the simulation, corresponding to a point $x_{P}=(x, y)$ of the spatial domain, the model calculates the probability $u\left(x_{P}, t\right)$ of being burnt at time $t$ and space $x$ evaluating the fire frequency for each cell, based on a significant number of stochastic simulations and each simulation is performed for the same ignitions and wind conditions. This procedure is resumed in Figure (2.1)


Figure 2.1: Averaging procedure of single realization adopted in PROPAGATOR.

The fire spreading is stochastically calculated considering the directions between the center of the $i$-cell and the ones of the neighboring cells, the slopes between the cells and the possible different moisture conditions. Each cell is characterized by a vegetation type.

Fuel models are adopted by widespread fire propagation models to classify the physical characteristics such as fuel load, heat content, and height of live and dead biomass that contribute to the size, intensity, and duration of a fire.
PROPAGATOR adopts a manageable simplified custom fuel model with seven available fuel types corresponding to seven different types of vegetation.

The considered fuel types are the following:

- broad-leaves;
- shrubs;
- grasslands;
- fire-prone conifers;
- agro-forestry areas;
- non-fire prone forest;
- non-vegetated areas.

The class called "non-vegetated areas" includes man-made buildings and infrastructures (e.g., streets, villages and towns) and the non-vegetated terrains, such as natural bare soil. Fire propagation cannot take place in this class. Rivers, lakes, and seas are considered by default as non-burnable areas as well.

The fire propagates from a cell $i$ to the neighbor cell $j$ with a probability $p_{i j}$, called Fire Spread Probability, which depends heavily on the involved vegetation types. The $p_{i j}$ is also influenced by the slope between the two cells, the wind effect (direction and velocity), and the fuel moisture content of the $j$-cell. The probability of the fire propagation $p_{i j}$ from an ignited $i$-cell at the time $t_{k}$ to a $j$-cell is calculated applying the cumulative binomial probability formula

$$
\begin{equation*}
p_{i j}=\left(1-\left(1-p_{n}\right)^{\alpha_{w h}}\right) \cdot e_{m} \tag{2.1}
\end{equation*}
$$

where: $p_{n}$ is the Nominal Fire Spread Probability, which represents the possibility for the i-cell, characterized by a certain vegetation cover, to ignite an adjacent j cell, characterized by the same, or another, vegetation cover; $\alpha_{w h}$ is the factor that combines the topographic and wind influence on the probability; $e_{m}$ is the factor that simulates the effect of the fine fuel moisture content.

The model takes into account the vegetation of the cell that is burning and the cells where the fire can propagate and it analyzes how a certain type of vegetation can ignite other types of vegetation, or also the same vegetation type.
These probability values are given in input through a fire spread probability table, Table (2.2), which considers all the possible combinations between the different vegetation and land-cover types.

|  |  | Burning Cell |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Broadleaves | Shrubs | Grassland | Fire-Prone Conifers | Agro-Forestry Areas | Not Fire-Prone Forest |
|  | Broadleaves | 0.3 | 0.375 | 0.25 | 0.275 | 0.25 | 0.25 |
|  | Shrubs | 0.375 | 0.375 | 0.35 | 0.4 | 0.3 | 0.375 |
|  | Grassland | 0.45 | 0.475 | 0.475 | 0.475 | 0.375 | 0.475 |
|  | Fire-prone conifers | 0.225 | 0.325 | 0.25 | 0.35 | 0.2 | 0.35 |
|  | Agro-forestry areas | 0.25 | 0.25 | 0.3 | 0.475 | 0.35 | 0.25 |
|  | Not fire-prone forest | 0.075 | 0.1 | 0.075 | 0.275 | 0.075 | 0.075 |
| Nominal Fire Spread Velocity [m/min] |  | 100 | 140 | 120 | 200 | 120 | 60 |

Figure 2.2: In the first six rows, the values of the nominal fire spread probability $p_{n}$ between all the species are given. In the last row, nominal fire spread velocity $v_{n}$ is reported.

Not fire-prone forest class represents the low-flammable forests: its probability of being ignited is quite low, except if the burning cell is a Fire-prone conifers cell. Medium flammable tall vegetation is considered in the Broadleaves class, while Fire-prone conifers class included the highly flammable tall vegetation. There are also three classes which represent the medium to low vegetation: Agro-forestry areas represent areas with a low vegetable density characterized by low probability of propagation; the Shrubs class includes the medium-flammable low vegetation; Grassland class represent the high-flammable very low vegetation.

### 2.5 The role of the slope in wildfire propagation

The slope and the wind speed and direction can modify the initial value of $p_{n}$, increasing or decreasing the nominal value depending on the direction of propagation [10].

The influence of the topography is taken into account through the slope between the two cells. The slope increases the propagation probability $p_{n}$ when the slope increases in the direction of propagation (uphill case) and it decreases $p_{n}$ if slope decreases in the direction of propagation (downhill case).

In Figure (2.3), it is possible to notice how $\alpha_{w h}$ impacts the Fire Spread Probability, $p$ : when $\alpha_{w h}$ is equal to 1 , the nominal transitional probability is obtained; when this factor is not unitary, it is possible to evaluate the effect of the possible combinations of slope, wind speed, and direction.


Figure 2.3: Influence of the combined slope-wind factor on the Fire Spread Probability. The plot portrays the dependence of $p_{i j}$ of Formula (2.1) on the slope-wind factor $\alpha_{w h}$, given a fixed $e_{m}=1$ and several values for $p_{n}$.

When a cell is ignited, the transition time of the fire is modeled by combining the Rate of Spread $v_{\text {prop }}$ and the fuel moisture factor $f_{m}$, with the distance $d$ from the center of the $i$-cell which propagated the fire to the center of the newly ignited $j$-cell. In particular, the transition time $\Delta t$ is calculated as:

$$
\begin{equation*}
\Delta t=\frac{d}{v_{\text {prop }} \cdot f_{m}} \tag{2.2}
\end{equation*}
$$

It is acknowledged that the flammability of the vegetal fuel and, consequently, the rate of the spread of a fire depends exponentially on the fuel moisture content, $f_{m}$ is calculated using the formulation proposed by Marino:

$$
\begin{equation*}
f_{m}=e^{c \cdot M_{n}} \tag{2.3}
\end{equation*}
$$

where $c$ is a constant that has been set at -0.014 and $M_{n}$ is the fuel moisture (ranging from 0 to 1 ).
The Rate of Spread $v_{\text {prop }}$ is calculated starting from the nominal $v_{n}$, which stands for the Fire Spread Velocity for each vegetation type without slope and wind effects, and then modifying it by considering the slope and the wind effects (values in Table (2.2)).

Slope and wind effects have been evaluated through the formulations proposed by Sun, which are calculated as follow:

- the wind speed factor $K_{w}$ is evaluated as

$$
\begin{equation*}
K_{w}=\exp (0.1783 V) \tag{2.4}
\end{equation*}
$$

where $V$ is the wind velocity in the direction of propagation, in $[\mathrm{m} / \mathrm{s}]$

- the slope factor $K_{\phi}$ is evaluated as

$$
\begin{equation*}
K_{\phi}=\exp \left(3.533(\tan \phi)^{1.2}\right) \tag{2.5}
\end{equation*}
$$

where $\phi$ is the terrain slope angle in the direction of propagation.
The Rate of Spread $v_{\text {prop }}$ is then evaluated multiplying the nominal Fire Spread Velocity $v_{n}$ by the two factors, $K_{w}$ and $K_{\phi}$.

In the second part of this section, we focus on the role of the mesoscopic characteristics of wildfires. We refer, in particular, to flame geometry, which strongly affects dire spreading, and to terrain slope. Additionally, an accurate estimation of geometrical properties allows a determination of how wildfires can be controlled: indeed, flame length is used to determine the size of fire control lines, while flame height is used to predict the heat flux exposure. Moreover, flame geometry is a descriptor of the surrounding vegetation; therefore, it is considered in fire-fighting strategies.

Therefore, motivated by the lack of this important foundation, we theoretically establish a formula for estimating the flame height and length in wildfires from the fireline intensity. The derivation is based on the energy conservation principle and on the concept of the energy flow rate in the convection column above a fireline, the latter was originally introduced by Byram in 1959.
Evidently, flame geometry is strongly affected by wind and terrain slope. In our formulation, we assume that, in the no-wind no-slope condition, flame geometry is fully characterised by the process's energy, Figure (2.4), while the wind and slope rule the flame tilting angle and they cause a stretching of the flame, Figure (2.5).


Figure 2.4: Flame geometry of real wildland fire in flat terrain.

In this study, we adopt, for our convenience, definitions that allow for stressing the separation between flame angle and terrain slope angle.
We define flame geometry as follows:
Definition 2.1 (Flame height). The flame height, $h(m)$, is measured along the axis perpendicular to the terrain that can be sloped.

Definition 2.2 (Flame length). The flame length $L_{f}(m)$ is defined as the distance between the flame height tip and the midpoint of the flame depth.

Definition 2.3 (Flame tilt). The flame tilt is defined as the sum of the terrain slope angle $\omega$ and the tilting angle $\Theta$.

These flame geometry characteristics are related by the formula (see Fig.2.5)

$$
\begin{equation*}
h=L_{f} \cos (\Theta+\omega) \tag{2.6}
\end{equation*}
$$

## UP-SLOPE WIND



Figure 2.5: Flame geometry in cases of up-slope wind.

The tilting angle $\Theta$ accounts for the concurrent effects of wind and slope that are not joined in a simple additive formula, i.e.

$$
\Theta=\Theta(U, \omega)=\Theta\left(c_{U} U+c_{\omega} \omega\right)
$$

where $U$ is the mean wind while the parameters $c_{U}$ and $c_{\omega}$ represent the corresponding dimensional scales; and it reduces to the corresponding angle in the limiting cases:

$$
\Theta=\Theta(U, \omega)=\left\{\begin{array}{l}
\Theta(U, 0)=\Theta_{w}(U), \text { with wind but no-slope },  \tag{2.7}\\
\Theta(U, \omega), \text { with slope and wind } \\
\Theta(0, \omega)=\Theta_{s}(\omega), \text { with slope but no-wind }
\end{array}\right.
$$

Several experimental measurements display a power-law relationship between the flame length $L_{f}$ and the Fireline Intensity $I_{f}$.

The fireline intensity $I_{f}[\mathrm{kWm}-1]$, was established by Byram from measurements of fire spread and fuel consumption. Notwithstanding, fireline intensity is of paramount importance in quantifying wildfire behaviour, both in applied and in theoretical studies.

It is also related to flame geometry. A widely used approximated empirical relation is given by

$$
\begin{equation*}
L_{f}=\beta_{0} I_{f} \beta_{1} \tag{2.8}
\end{equation*}
$$

where $\beta_{0}$ and $\beta_{1}$ are two positive parameters. Unfortunately, the values of $\beta_{0}$ and $\beta_{1}$ tend to be mostly scattered, while the sole constraint that emerges is that the power-law exponent is $0<\beta_{1}<1$.

To conclude, fireline intensity is related to the propagation of a front and drives fire-spotting, which accelerates the spreading of a fire; therefore, fire-spotting is crucial for simulating the evolution of a burning area, but this is not a topic of this treatment. For further information [15], [9].

## Chapter 3

## Results

### 3.1 Introduction

In this chapter will be analysed the results obtained from the study of fires propagation by areas of uniform vegetation and variable slope.
The aim will be to demonstrate, as already anticipated above, that, despite two types of distribution approach well the process pdf, the evolution of data is better described by the beta distribution.

### 3.1.1 Modelling

Before starting the analysis, let's describe the initial state of the process and the results provided by PROPAGATOR.

## Initial conditions

- ignitions: point;
- wind speed: 0 [km/h];
- moisture: 2;
- vegetation: uniform;
- slope: floating;
- number of threads ( n ): 1000 ;
- space limit: 20 [km];
- time limit: 100 [min];
- time resolution: 10 [min];
- discretization step $(d t): 10$.

Note that the process is being analyzed excluding the phenomenon of firespotting and possible external interventions to stop the fire.

## Results

- matrix $X[(n+1)(d t+1) \times 2]$ : containing in the first column the values of burned area for each realization and in the second the corresponding time instants. Output that PROPAGATOR provides thanks to a modification we made;
- matrix $A[(d t+1) \times 2]$ : containing in the first column the values of average area burned every 1000 realizations and in the second the corresponding instants of time; it is equivalent to a report of the situation every 10 minutes. Output that PROPAGATOR provides by default.


### 3.1.2 Analysis

For greater clarity we divide the statistical analysis into several sections:

- Data cleansing: given the $X$ matrix of output, we remove the burned area values that do not vary with time. Therefore, from a phenomenological point of view, cells that do not contribute to the extension of fire are being removed;
- Histogram representation: for each realization the corresponding histogram is constructed, from which a first purely empirical pdf is obtained. Then these histograms are plotted all on the same graph to appreciate the trend over time.
- Study of the following variables trend: mean area (considering matrix $A$ ); variance; correlation; maximum of the pdf.
- Data scaling: the data of the $X$ matrix are scaled and we indicate them with $Y$. In order to use beta.fit it is necessary a data scaling in interval $[0,1]$.
- Parameters calculation: are calculated, from the data, therefore empirically, the parameters required for beta and generalized logistic distributions.
- Data fitting: thanks to the calculated parameters a fit of the data is performed, using both the beta pdf and generalized logistic pdf, to compare them.
- Analysis of moments of various order: considering the four moments (mean, variance, skewness and kurtosis), these are calculated both for the initial normalized data and for the two distributions used later. The objective is to compare the results and determine which of the two distributions best describes the phenomenon in analysis.

The analysis carried out was illustrated in broad terms, now we deepen the individual cases studied. In particular, the following: slope10, slope15, slope20, slope 30 , slope 40 and slope 45.

### 3.2 Slope10

### 3.2.1 Data cleansing

After "cleaning" the data, we get the following results at every instant of observation:

## Starting Data

| $X 0[1000]$ | $t=0$ | $X 6[939]$ | $t=60$ |
| :---: | :---: | :---: | :---: |
| $X 1[928]$ | $t=10$ | $X 7[941]$ | $t=70$ |
| $X 2[946]$ | $t=20$ | $X 8[941]$ | $t=80$ |
| $X 3[945]$ | $t=30$ | $X 9[941]$ | $t=90$ |
| $X 4[943]$ | $t=40$ | $X 10[941]$ | $t=100$ |
| $X 5[940]$ | $t=50$ |  |  |

Table 3.1: The $X$ matrix is divided into column vectors representing the number of threads for each time $t$.

### 3.2.2 Histogram representation

For each vector we consider the corresponding histogram and its empirical distribution:


Figure 3.1: $X 1[928], t=10$.
Figure 3.2: $X 2[946], t=20$.


Figure 3.3: $X 3[945], t=30$.
Figure 3.4: $X 4[943], t=40$.



Figure 3.5: $X 5[940], t=50$.


Figure 3.6: $X 6[939], t=60$.



Figure 3.9: $X 9[941], t=90$.


Figure 3.10: $X 10[941], t=100$.

Then the histograms and the empirical distributions obtained for each individual vector are plotted on the same graph to study its trend over time


Figure 3.11: Trend over the time. Legend: blue $=0 \mathrm{~m}$; orange $=10 \mathrm{~m}$; green $=20 \mathrm{~m}$; red $=30 \mathrm{~m}$; violet $=40 \mathrm{~m}$; brown $=50 \mathrm{~m}$; pink $=60 \mathrm{~m}$; grey $=70 \mathrm{~m}$; light green $=80 \mathrm{~m}$; light blue $=90 \mathrm{~m}$; blue $=100 \mathrm{~m}$.

### 3.2.3 Other variables trend

Considering the outputs in matrix $A$, we study the trend of mean area. Instead, with the data of the $X$ matrix we study the trends of variance, maximum of the distribution and correlation. Note that maximum of the distribution means the element belonging to the dataset to which the highest peak of the curve corresponds.


Figure 3.12: Mean Area.


Figure 3.14: Correlation.

Figure 3.13: Variance.


Figure 3.15: Pdf maximum.

### 3.2.4 Data scaling

Let now consider the variable

$$
Y=\frac{X-A}{\sqrt{\operatorname{var}(X-A)}}
$$

as well as the $X$ matrix, the $Y$ matrix is divided into column vectors corresponding to the instants of observation.

Moreover, on the vectors of the matrix $Y$ has been operated the same analysis that is not reported, in order to focus the attention on the successive more important results.

### 3.2.5 Parameters calculation and data fitting

After normalizing the data, considering the variable Y of scaled data into the range $[0,1]$. The parameters, $\alpha$ and $\beta$, for the beta and generalized logistic distributions are then calculated. Using these parameters a fit of the data is performed obtaining the following graphs


Figure 3.16: Linear beta fit.


Figure 3.18: Linear genlog fit.


Figure 3.17: Logarithmic beta fit.

Figure 3.19: Logarithmic genlog fit.

Therefore graphically both distributions are presented as good candidates for the description of the phenomenon. Now, to determine which of the two retains the properties of the case studied, we analyze the moments and compare them with the initial data.

### 3.2.6 Analysis of moments of various order

Considering the four moments (mean, variance, skewness and kurtosis), they are calculated on the scaled $Y$ vectors and their distribution. The results are shown in the following graphs:


Figure 3.20: Mean. On the left the comparison between the mean calculated on the data (green) and the mean calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.21: Variance. On the left the comparison between the variance calculated on the data (green) and the variance calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.22: Skewness. On the left the comparison between the skewness calculated on the data (green) and the skewness calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.23: Kurtosis. On the left the comparison between the variance calculated on the data (green) and the variance calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.

### 3.3 Slope15

### 3.3.1 Data cleansing

After "cleaning" the data, we get the following results at every instant of observation. Note that in this case the vector $X 1$ does not appear, this indicates that after the first 10 minutes no cell still burns.

## Starting Data

| $X 0[1000]$ | $t=0$ | $X 6[939]$ | $t=60$ |
| :---: | :---: | :---: | :---: |
| $X 2[908]$ | $t=20$ | $X 7[941]$ | $t=70$ |
| $X 3[864]$ | $t=30$ | $X 8[944]$ | $t=80$ |
| $X 4[941]$ | $t=40$ | $X 9[943]$ | $t=90$ |
| $X 5[936]$ | $t=50$ | $X 10[943]$ | $t=100$ |

Table 3.2: The $X$ matrix is divided into column vectors representing the number of threads for each time $t$.

### 3.3.2 Histogram representation

For each vector we consider the corresponding histogram and its empirical distribution:



Figure 3.24: $X 2[908], t=20$.



Figure 3.25: $X 3[864], t=30$.


Figure 3.26: $X 4[941], t=40$.


Figure 3.28: $X 6[939], t=60$.
Figure 3.29: $X 7[941], t=70$.


Figure 3.30: $X 8[944], t=80$.
Figure 3.31: $X 9[943], t=90$.


Figure 3.32: $X 10[943], t=100$

Then the histograms and the empirical distributions obtained for each individual vector are plotted on the same graph to study its trend over time


Figure 3.33: Trend over the time. Legend: blue $=20 \mathrm{~m}$; orange $=30 \mathrm{~m}$; green $=40 \mathrm{~m}$; red $=50 \mathrm{~m}$; violet $=60 \mathrm{~m}$; brown $=70 \mathrm{~m}$; pink $=80 \mathrm{~m}$; grey $=90 \mathrm{~m}$; light green $=100 \mathrm{~m}$.

### 3.3.3 Other variables trend

Considering the outputs in matrix $A$, we study the trend of mean area. Instead, with the data of the $X$ matrix we study the trend of variance, maximum of the distribution and correlation. Note that maximum of the distribution means the element belonging to the dataset to which the highest peak of the curve corresponds.


Figure 3.34: Mean Area.


Figure 3.36: Correlation.


Figure 3.35: Variance.


Figure 3.37: Pdf maximum.

### 3.3.4 Data scaling

Let now consider the variable

$$
Y=\frac{X-A}{\sqrt{\operatorname{var}(X-A)}}
$$

as well as the $X$ matrix, the $Y$ matrix is divided into column vectors corresponding to the instants of observation.

Moreover, on the vectors of the matrix $Y$ has been operated the same analysis that is not reported, in order to focus the attention on the successive more important results.

### 3.3.5 Parameters calculation and data fitting

After normalizing the data, considering the variable Y of scaled data into the range $[0,1]$. The parameters, $\alpha$ and $\beta$, for the beta and generalized logistic distributions are then calculated. Using these parameters a fit of the data is performed obtaining the following graphs


Figure 3.38: Linear beta fit.


Figure 3.40: Linear genlog fit.


Figure 3.39: Logarithmic beta fit.

Figure 3.41: Logarithmic genlog fit.

Therefore graphically both distributions are presented as good candidates for the description of the phenomenon. Now, to determine which of the two retains the properties of the case studied, we analyze the moments and compare them with the initial data.

### 3.3.6 Analysis of moments of various order

Considering the four moments (mean, variance, skewness and kurtosis), they are calculated on the sceled $Y$ vectors and their distribution. The results are shown in the following graphs:



Figure 3.42: Mean. On the left the comparison between the mean calculated on the data (green) and the mean calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.43: Variance. On the left the comparison between the variance calculated on the data (green) and the variance calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.44: Skewness. On the left the comparison between the skewness calculated on the data (green) and the skewness calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.45: Kurtosis. On the left the comparison between the kurtosis calculated on the data (green) and the kurtosis calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.

### 3.4 Slope20

### 3.4.1 Data cleansing

After "cleaning" the data, we get the following results at every instant of observation:

## Starting Data

| $X 0[1000]$ | $t=0$ | $X 6[954]$ | $t=60$ |
| :---: | :---: | :---: | :---: |
| $X 1[912]$ | $t=10$ | $X 7[949]$ | $t=70$ |
| $X 2[900]$ | $t=20$ | $X 8[954]$ | $t=80$ |
| $X 3[892]$ | $t=30$ | $X 9[954]$ | $t=90$ |
| $X 4[948]$ | $t=40$ | $X 10[953]$ | $t=100$ |
| $X 5[950]$ | $t=50$ |  |  |

Table 3.3: The $X$ matrix is divided into column vectors representing the number of threads for each time $t$.

### 3.4.2 Histogram representation

For each vector we consider the corresponding histogram and its empirical distribution:


Figure 3.49: $X 4[948], t=40$


Figure 3.52: $X 7[949], t=70$.
Figure 3.53: $X 8[954], t=80$.


Figure 3.54: $X 9[954], t=90$.


Figure 3.55: $X 10[953], t=100$.

Then the histograms and the empirical distributions obtained for each individual vector are plotted on the same graph to study its trend over time


Figure 3.56: Trend over the time. Legend: blue $=10 \mathrm{~m}$; orange $=20 \mathrm{~m}$; green $=30 \mathrm{~m}$; red $=40 \mathrm{~m}$; violet $=50 \mathrm{~m}$; brown=60m; pink $=70 \mathrm{~m}$; grey $=80 \mathrm{~m}$; light green $=90 \mathrm{~m}$; light blue $=100 \mathrm{~m}$.

### 3.4.3 Other variables trend

Considering the outputs in matrix $A$, we study the trend of mean area. Instead, with the data of the $X$ matrix we study the trend of variance, maximum of the distribution and correlation. Note that maximum of the distribution means the element belonging to the dataset to which the highest peak of the curve corresponds.


Figure 3.57: Mean Area.


Figure 3.59: Correlation.


Figure 3.58: Variance.


Figure 3.60: Pdf maximum.

### 3.4.4 Data scaling

Let now consider the variable

$$
Y=\frac{X-A}{\sqrt{\operatorname{var}(X-A)}}
$$

as well as the $X$ matrix, the $Y$ matrix is divided into column vectors corresponding to the instants of observation.

Moreover, on the vectors of the matrix $Y$ has been operated the same analysis that is not reported, in order to focus the attention on the successive more important results.

### 3.4.5 Parameters calculation and data fitting

After normalizing the data, considering the variable Y of scaled data into the range $[0,1]$. The parameters, $\alpha$ and $\beta$, for the beta and generalized logistic distributions are then calculated. Using these parameters a fit of the data is performed obtaining the following graphs


Figure 3.61: Linear beta fit.


Figure 3.62: Logarithmic beta fit.


Figure 3.63: Linear genlog fit.


Figure 3.64: Logarithmic genlog fit.

Therefore graphically both distributions are presented as good candidates for the description of the phenomenon. Now, to determine which of the two retains the properties of the case studied, we analyze the moments and compare them with the initial data.

### 3.4.6 Analysis of moments of various order

Considering the four moments (mean, variance, skewness and kurtosis), they are calculated on the scaled $Y$ vectors and their distribution. The results are shown in the following graphs:


Figure 3.65: Mean. On the left the comparison between the mean calculated on the data (green) and the mean calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.66: Variance. On the left the comparison between the variance calculated on the data (green) and the variance calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.67: Skewness. On the left the comparison between the skewness calculated on the data (green) and the skewness calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.68: Kurtosis. On the left the comparison between the kurtosis calculated on the data (green) and the kurtosis calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.

### 3.5 Slope30

### 3.5.1 Data cleansing

After "cleaning" the data, we get the following results at every instant of observation:

## Starting Data

| $X 0[1000]$ | $t=0$ | $X 6[906]$ | $t=60$ |
| :---: | :---: | :---: | :---: |
| $X 1[920]$ | $t=10$ | $X 7[902]$ | $t=70$ |
| $X 2[885]$ | $t=20$ | $X 8[901]$ | $t=80$ |
| $X 3[875]$ | $t=30$ | $X 9[903]$ | $t=90$ |
| $X 4[878]$ | $t=40$ | $X 10[904]$ | $t=100$ |
| $X 5[906]$ | $t=50$ |  |  |

Table 3.4: The $X$ matrix is divided into column vectors representing the number of threads for each time $t$.

### 3.5.2 Histogram representation

For each vector we consider the corresponding histogram and its empirical distribution:


Figure 3.69: $X 1[920], t=10$.
Figure 3.70: $X 2[885], t=20$.


Figure 3.71: $X 3[875], t=30$.
Figure 3.72: $X 4[878], t=40$


Figure 3.73: $X 5[906], t=50$.


Figure 3.75: $X 7[902], t=70$.
Figure 3.74: $X 6[906], t=60$.

Figure 3.76: $X 8[901], t=80$

Figure 3.77: $X 9[903], t=90$.
Figure 3.78: $X 10[904], t=100$.

Then the histograms and the empirical distributions obtained for each individual vector are plotted on the same graph to study its trend over time


Figure 3.79: Trend over the time. Legend: blue $=10 \mathrm{~m}$; orange $=20 \mathrm{~m}$; green $=30 \mathrm{~m}$; red $=40 \mathrm{~m}$; violet $=50 \mathrm{~m}$; brown $=60 \mathrm{~m}$; pink $=70 \mathrm{~m}$; grey $=80 \mathrm{~m}$; light green $=90 \mathrm{~m}$; light blue $=100 \mathrm{~m}$.

### 3.5.3 Other variables trend

Considering the outputs in matrix $A$, we study the trend of mean area. Instead, with the data of the $X$ matrix we study the trend of variance, maximum of the distribution and correlation. Note that maximum of the distribution means the element belonging to the dataset to which the highest peak of the curve corresponds.


### 3.5.4 Data scaling

Let now consider the variable

$$
Y=\frac{X-A}{\sqrt{\operatorname{var}(X-A)}}
$$

as well as the $X$ matrix, the $Y$ matrix is divided into column vectors corresponding to the instants of observation.

Moreover, on the vectors of the matrix $Y$ has been operated the same analysis that is not reported, in order to focus the attention on the successive more important results.

### 3.5.5 Parameters calculation and data fitting

After normalizing the data, considering the variable Y of scaled data into the range $[0,1]$. The parameters, $\alpha$ and $\beta$, for the beta and generalized logistic distributions are then calculated. Using these parameters a fit of the data is performed obtaining the following graphs


Figure 3.84: Linear beta fit.


Figure 3.86: Linear genlog fit.


Figure 3.85: Logarithmic beta fit.


Figure 3.87: Logarithmic genlog fit.

Therefore graphically both distributions are presented as good candidates for the description of the phenomenon. Now, to determine which of the two retains the properties of the case studied, we analyze the moments and compare them with the initial data.

### 3.5.6 Analysis of moments of various order

Considering the four moments (mean, variance, skewness and kurtosis), they are calculated on the scaled $Y$ vectors and their distribution. The results are shown in the following graphs:


Figure 3.88: Mean. On the left the comparison between the mean calculated on the data (green) and the mean calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.89: Variance. On the left the comparison between the variance calculated on the data (green) and the variance calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.90: Skewness. On the left the comparison between the skewness calculated on the data (green) and the skewness calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.91: Kurtosis. On the left the comparison between the kurtosis calculated on the data (green) and the kurtosis calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.

### 3.6 Slope40

### 3.6.1 Data cleansing

After "cleaning" the data, we get the following results at every instant of observation:

## Starting Data

| $X 0[1000]$ | $t=0$ | $X 6[894]$ | $t=60$ |
| :---: | :---: | :---: | :---: |
| $X 1[932]$ | $t=10$ | $X 7[894]$ | $t=70$ |
| $X 2[898]$ | $t=20$ | $X 8[894]$ | $t=80$ |
| $X 3[894]$ | $t=30$ | $X 9[894]$ | $t=90$ |
| $X 4[894]$ | $t=40$ | $X 10[894]$ | $t=100$ |
| $X 5[894]$ | $t=50$ |  |  |

Table 3.5: The $X$ matrix is divided into column vectors representing the number of threads for each time $t$.

### 3.6.2 Histogram representation

For each vector we consider the corresponding histogram and its empirical distribution:


Figure 3.92: $X 1[932], t=10$.
Figure 3.93: $X 2[898], t=20$.


Figure 3.94: $X 3[894], t=30$.
Figure 3.95: $X 4[894], t=40$


Figure 3.96: $X 5[894], t=50$.



Figure 3.98: $X 7[894], t=70$.


Figure 3.100: $X 9[894], t=90$.

Figure 3.97: $X 6[894], t=60$.



Figure 3.101: $X 10[894], t=100$.

Then the histograms and the empirical distributions obtained for each individual vector are plotted on the same graph to study its trend over time


Figure 3.102: Trend over the time. Legend: blue $=10 \mathrm{~m}$; orange $=20 \mathrm{~m}$; green $=30 \mathrm{~m}$; red $=40 \mathrm{~m}$; violet $=50 \mathrm{~m}$; brown=60m; pink $=70 \mathrm{~m}$; gre $y=80 \mathrm{~m}$; light green $=90 \mathrm{~m}$; light blue $=100 \mathrm{~m}$.

### 3.6.3 Other variables trend

Considering the outputs in matrix $A$, we study the trend of mean area. Instead, with the data of the $X$ matrix we study the trend of variance, maximum of the distribution and correlation. Note that maximum of the distribution means the element belonging to the dataset to which the highest peak of the curve corresponds.


### 3.6.4 Data scaling

Let now consider the variable

$$
Y=\frac{X-A}{\sqrt{\operatorname{var}(X-A)}}
$$

as well as the $X$ matrix, the $Y$ matrix is divided into column vectors corresponding to the instants of observation.

Moreover, on the vectors of the matrix $Y$ has been operated the same analysis that is not reported, in order to focus the attention on the successive more important results.

### 3.6.5 Parameters calculation and data fitting

After normalizing the data, considering the variable Y of scaled data into the range $[0,1]$. The parameters, $\alpha$ and $\beta$, for the beta and generalized logistic distributions are then calculated. Using these parameters a fit of the data is performed obtaining the following graphs

Figure 3.107: Linear beta fit.

Figure 3.108: Logarithmic beta fit.


Therefore graphically both distributions are presented as good candidates for the description of the phenomenon. Now, to determine which of the two retains the properties of the case studied, we analyze the moments and compare them with the initial data.

### 3.6.6 Analysis of moments of various order

Considering the four moments (mean, variance, skewness and kurtosis), they are calculated on the scaled $Y$ vectors and their distribution. The results are shown in the following graphs:


Figure 3.111: Mean. On the left the comparison between the mean calculated on the data (green) and the mean calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.112: Variance. On the left the comparison between the variance calculated on the data (green) and the variance calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.113: Skewness. On the left the comparison between the skewness calculated on the data (green) and the skewness calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.114: Kurtosis. On the left the comparison between the kurtosis calculated on the data (green) and the kurtosis calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.

### 3.7 Slope45

### 3.7.1 Data cleansing

After "cleaning" the data, we get the following results at every instant of observation:

## Starting Data

| $X 0[1000]$ | $t=0$ | $X 6[883]$ | $t=60$ |
| :---: | :---: | :---: | :---: |
| $X 1[937]$ | $t=10$ | $X 7[883]$ | $t=70$ |
| $X 2[889]$ | $t=20$ | $X 8[883]$ | $t=80$ |
| $X 3[884]$ | $t=30$ | $X 9[883]$ | $t=90$ |
| $X 4[884]$ | $t=40$ | $X 10[883]$ | $t=100$ |
| $X 5[883]$ | $t=50$ |  |  |

Table 3.6: The $X$ matrix is divided into column vectors representing the number of threads for each time $t$.

### 3.7.2 Histogram representation

For each vector we consider the corresponding histogram and its empirical distribution:


Figure 3.115: $X 1$ [937], $t=10$
Figure 3.116: $X 2[889], t=20$.


Figure 3.117: $X 3[884], t=30$
Figure 3.118: $X 4[884], t=40$.


Figure 3.119: $X 5[883], t=50$.
Figure 3.120: $X 6[883], t=60$.


Figure 3.121: $X 7[883], t=70$.
Figure 3.122: $X 8[883], t=80$.


Figure 3.123: $X 9[883], t=90$.


Figure 3.124: $X 10[883], t=100$.

Then the histograms and the empirical distributions obtained for each individual vector are plotted on the same graph to study its trend over time


Figure 3.125: Trend over the time. Legend: blue $=10 \mathrm{~m}$; orange $=20 \mathrm{~m}$; green $=30 \mathrm{~m}$; red $=40 \mathrm{~m}$; violet $=50 \mathrm{~m}$; brown=60m; pink $=70 \mathrm{~m}$; gre $y=80 \mathrm{~m}$; light green $=90 \mathrm{~m}$; light blue $=100 \mathrm{~m}$.

### 3.7.3 Other variables trend

Considering the outputs in matrix $A$, we study the trend of mean area. Instead, with the data of the $X$ matrix we study the trend of variance, maximum of the distribution and correlation. Note that maximum of the distribution means the element belonging to the dataset to which the highest peak of the curve corresponds.


Figure 3.126: Mean Area.


Figure 3.128: Correlation.


Figure 3.127: Variance.


Figure 3.129: Pdf maximum.

### 3.7.4 Data scaling

Let now consider the variable

$$
Y=\frac{X-A}{\sqrt{\operatorname{var}(X-A)}}
$$

as well as the $X$ matrix, the $Y$ matrix is divided into column vectors corresponding to the instants of observation.

Moreover, on the vectors of the matrix $Y$ has been operated the same analysis that is not reported, in order to focus the attention on the successive more important results.

### 3.7.5 Parameters calculation and data fitting

After normalizing the data, considering the variable Y of scaled data into the range $[0,1]$. The parameters, $\alpha$ and $\beta$, for the beta and generalized logistic distributions are then calculated. Using these parameters a fit of the data is performed obtaining the following graphs


Figure 3.130: Linear beta fit.


Figure 3.132: Linear genlog fit.


Figure 3.131: Logarithmic beta fit.


Figure 3.133: Logarithmic genlog fit.

Therefore graphically both distributions are presented as good candidates for the description of the phenomenon. Now, to determine which of the two retains the properties of the case studied, we analyze the moments and compare them with the initial data.

### 3.7.6 Analysis of moments of various order

Considering the four moments (mean, variance, skewness and kurtosis), they are calculated on the scaled $Y$ vectors and their distribution. The results are shown in the following graphs:


Figure 3.134: Mean. On the left the comparison between the mean calculated on the data (green) and the mean calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.135: Variance. On the left the comparison between the variance calculated on the data (green) and the variance calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.136: Skewness. On the left the comparison between the skewness calculated on the data (green) and the skewness calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.


Figure 3.137: Kurtosis. On the left the comparison between the kurtosis calculated on the data (green) and the kurtosis calculated on their distribution obtained with betafit (red). On the right the same comparison but considering genlogisticfit.

## Conclusions

At the end of the analysis it is clear that the area burned, in a limited range of observation, by varying the slope of the territory, is distributed as a Beta. However, note that this result is not equally valid for very long observation times.

In order to justify the conclusion reached, the type of analytical approach adopted must be specified. There was no use of hand-implemented statistical tests or software for statistical analysis, as there was no interest in deepening this aspect. To identify the distribution in question, that is a limited domain distribution, compatible with the phenomenology of the case study, we have observed the graphs obtained from the data analysis and, by trial and error, the latter were compared with different types of known distributions having the properties involved. After identifying the most faithful, the moments of various order were analyzed, to have a mathematical response of what was supposed.

We consider therefore the two distributions identified: the beta and the generalized logistics. It can be seen that, regardless of the slope value considered, if a betafit ${ }^{1}$ and a genlogisticfit ${ }^{2}$ are performed on the data, once calculated mean, variance, skewness and kurtosis for both the fit, the trend of these last ones, relatively to the genlogisticfit, is more different from the trend of the same calculated

[^0]on the initial data. This has therefore led to a focus on beta distribution as a possible solution to the problem.

After identifying the beta as optimal distribution, it remains to be clarified that the data analysis carried out has repeatedly required a change in the output provided by the code, such as cleaning, scaling, this has led to an accumulation of the error on the results that caused, in the final charts, the increase in the gap between skewness and kurtosis for betafit data and those calculated on initial data.

In this paper, therefore, in line with the phenomenology of the case studies, a new application of the beta distribution has been identified. The newness of this application is evident not only in the type of field, that of the propagation of fires, but also in the type of model, that of cellular automata. Previously, in fact, the beta distribution had found application in various fields, such as financial or demographic, very different from the one in question.

Finally, with this thesis work we want to bring to the attention, in addition to the interesting results obtained, the breadth of skills developed, such as: deepening many aspects of statistical analysis and stochastic simulation, develop a greater critical sense of the results, deepen the knowledge of routine for the use of Python, compare with statistical testing software such as Minitab and many others. All this has been possible thanks to the constant collaboration with the research team of Statistical Physics, with the Dott. Gianni Pagnini and to the continuous stimuli provided by the scientific research environment.

Working in an international research center like the BCAM has certainly allowed to implement the preparation considerably, both from a scientific and personal point of view, if with some initial difficulties due to language and adaptation in a much larger environment than the academic one.

## Conclusioni

A conclusione del lavoro si evince che l'area bruciata, in un intervallo di osservazione limitato, al variare della pendenza del territorio, si distribuisce come una Beta. Si noti, però, che tale risultato non è altrettanto valido per tempi di osservazione molto lunghi.

Per giustificare come si è arrivati a tale conclusione, bisogna specificare il tipo di approccio analitico adottato. Non si è fatto uso di test statistici implementati a mano o di software per l'analisi statistica, in quanto non si aveva interesse ad approfondire questo aspetto. Per individuare la distribuzione in oggetto, ovvero una distribuzione a dominio limitato, compatibilmente con la fenomenologia del caso di studio, si sono osservati i grafici ottenuti dall'analisi dati e, per tentativi, si sono confrontati questi ultimi con diversi tipi di distribuzioni note aventi le proprietà interessate. Dopo aver individuato le maggiormente fedeli, si sono analizzati i momenti di vario ordine, per avere un riscontro matematico di quanto supposto.

Si considerino dunque le due distribuzioni individuate: la beta e la logistica generalizzata. È possibile notare che, indipendentemente dal valore di slope considerato, se si eseguono un betafit ${ }^{3} \mathrm{e}$ un genlogisticfit ${ }^{4}$ sui dati, una volta calcolate media, varianza, skewness e kurtosis per entrambi i fit, l'andamento di queste

[^1]ultime, relativamente al genlogisticfit, si discosta maggiormente rispetto all'andamento delle stesse calcolate sui dati iniziali. Questo ha dunque portato a focalizzare l'attenzione sulla distribuzione beta, in quanto possibile soluzione al problema.

Dopo aver individuato la beta come distribuzione ottimale, resta da chiarire che l'analisi dati effettuata ha richiesto più volte una modifica degli output forniti dal codice, come la pulizia, la scalatura, questo ha portato ad un accumulo dell'errore sui risultati che ha causato, nei grafici finali, l'aumento del gap tra skewness e kurtosis relative ai dati con betafit e quelle calcolate sui dati iniziali.

Nel presente elaborato dunque, in linea con la fenomenologia dei casi studiati, si è individuata una nuova applicazione della distribuzione beta. La novità di tale applicazione si evince non solo nel tipo di ambito, ovvero quello della propagazione di incendi, ma anche nel tipo di modello, quello degli automi cellulari. Precedentemente, infatti, la distribuzione beta aveva trovato applicazione in svariati campi, come quello finanziario o demografico, molto diversi da quello in oggetto.

Infine, con questo lavoro di tesi si intende portare all'attenzione, oltre agli interessanti risultati ottenuti, la vastità di competenze sviluppate, quali: approfondire numerosi aspetti dell'analisi statistica e della simulazione stocastica, sviluppare un maggiore senso critico nei confronti dei risultati, approfondire la conoscenza di routine per l'utilizzo di Python, confrontarsi con software per test statistici come Minitab e molte altre. Tutto questo è stato possibile grazie alla costante collaborazione con il team di ricerca di Statistical Physics, con il Dott. Gianni Pagnini e ai continui stimoli forniti dall'ambiente di ricerca scientifica.

Lavorare in un centro di ricerca internazionale come il BCAM ha sicuramente permesso di implementare notevolmente la preparazione, sia da un punto di vista scientifico che personale, se pur con qualche difficoltà iniziale dovuta alla lingua e all'adattamento in un ambiente molto più grande rispetto a quello accademico.

## Appendix A

## Basic on simulation

In this appendix are some preliminary works on the simulation of known stochastic processes carried out with the aim of developing a certain criticism with respect to the results obtained by a generic stochastic simulation, so as to prepare for fire simulation and PROPAGATOR use.

## A. 1 Wiener Process

## A.1.1 General facts

A stochastic process is a mathematical object that is intended to model the evolution in time of a random phenomenon. As will become clear in the sequel the appropriate setting is the following:

A stochastic process is an object of the form

$$
X=\left(\Omega, \mathscr{F},\left(\mathscr{F}_{t}\right)_{t \in T},\left(X_{t}\right)_{t \in T}, P\right)
$$

where

- $(\Omega, \mathscr{F}, P)$ is a probability space;
- $T$ is a ordered subset of $\mathbb{R}^{+}$;
- $\left(\mathscr{F}_{t}\right)_{t \in T}$ is filtration, i.e. an increasing family of sub- $\sigma$-algebras of $\mathscr{F}$ :
$\mathscr{F}_{s} \subset \mathscr{F}_{t}$ whenever $s \leq t ;$
- $\left(X_{t}\right)_{t \in T}$ is a family of r.v.'s on $(\Omega, \mathscr{F})$ taking values in a measurable space $(E, \mathscr{E})$ such that, for every $\mathrm{t}, X_{t}$ is $\mathscr{F}_{t}$-measurable. This fact is also expressed by saying that $\left(X_{t}\right)_{t}$ is adapted to the filtration $\left(\mathscr{F}_{t}\right)_{t}$

A process is said to be continuous (resp. a.s. continuous) if for every $\omega$ (resp. for almost every $\omega$ ) the map $t \mapsto X_{t}(\omega)$ is continuous. The definitions of a rightcontinuous process, an a.s. right-continuous process, etc., are quite similar.

A process is said to be standard if
a) the filtration $\left(\mathscr{F}_{t}\right)_{t}$ is right-continuous;
b) for every $t, \mathscr{F}_{t}$ contains the negligible events of $\mathscr{F}$.

A Wiener process is a real-valued process $W=\left(\Omega, \mathscr{F},\left(\mathscr{F}_{t}\right)_{t \geq 0},\left(W_{t}\right)_{t \geq 0}, P\right)$ such that
i) $W_{0}=0$ a.s.;
ii) for every $0 \leq s \leq t$ the r.v. $W_{t}-W_{s}$ is independent of $\mathscr{F} s$;
iii) for every $0 \leq s \leq t, W_{t}-W_{s}$ is $N(0, t-s)$-distributed.
moreover a Wiener process always has a Hölder continuous modification.

## Example (Gaussian processes)

An $\mathbb{R}^{m}$-valued process $\left(X_{t}\right)_{t}$ is said to be Gaussian if it is a Gaussian family i.e. if its finite-dimensional distributions are Gaussian. If we define

$$
\begin{aligned}
b_{t} & =E\left(X_{t}\right), \text { its mean function } \\
K_{s, t}^{i, j} & =\operatorname{Cov}\left(X_{i}(s), X_{j}(t)\right), \quad 1 \leq i, j \leq m, \text { its covariance function }
\end{aligned}
$$

then the finite-dimensional distributions of $\left(X_{t}\right)_{t}$ are completely determined by these two quantities.

## Remarks

- A Wiener process is a Gaussian process, i.e. the joint distributions of $W_{t_{1}}, \ldots, W_{t_{m}}$ are Gaussian.
- Sometimes it will be important to specify with respect to which filtration a Wiener process is considered. We shall speak of natural Wiener process when $\left(\mathscr{F}_{t}\right)_{t}$ is the natural filtration.


Figure A.1: A typical image of a path of a two-dimensional Wiener process for $0 \leq t \leq 1$ (a black small circle denotes the origin and the position at time 1).

For computational purposes it is useful to consider discretized Wiener process, where $W(t)$ is specified at discrete $t$ values. We thus set $d t=T / N$ for some positive integer $N$ and let $W_{j}$ denote $W\left(t_{j}\right)$ with $t_{j}=j d t$.

By definition:

- according with (i) $W_{0}=0$ with probability 1 ;
- according with (ii) and (iii)

$$
W_{j}=W_{j-1}+d W_{j}, \quad j=1,2, \ldots, N
$$

where each $d W_{j}$ is an independent random variable of the form $\sqrt{d t} N(0,1)$.

## Simulation

In this section we see the first aspects of the problem of simulating a stochastic process starting from the case of Wiener process.

The first idea (others are possible) for the simulation of a Wiener process is very simple: its increments being independent Gaussian r.v.'s, the problem is solved as soon as we are able to simulate a sequence of independent Gaussian r.v.'s.

If $Z_{1}, Z_{2}, \ldots$ are independent $m$-dimensional and $N(0,1)$-distributed r.v.'s on some probability space $(\Omega, \mathscr{F}, P)$, let us choose a grid of times $0<d t<2 d t<\ldots$ where $d t>0$ is a positive number (typically to be taken small).
Then the r.v. $\sqrt{d t} Z_{1}$ is $N(0, d t I)$-distributed, i.e. has the same distribution as the increment of a Wiener process over a time interval of size $d t$. Hence the r.v.'s

$$
\begin{aligned}
& \bar{W}_{t}(d t)=\sqrt{d t} Z_{1} \\
& \bar{W}_{t}(2 d t)=\sqrt{d t}\left(Z_{1}+Z_{2}\right) \\
& \bar{W}_{t}(k d t)=\sqrt{d t}\left(Z_{1}+\cdots+Z_{k}\right)
\end{aligned}
$$

have the same joint distributions as the positions at times $d t, 2 d t, \ldots, k d t, \ldots$ of a Wiener process.
If, for $k d t \leq t \leq(k+1) d t$, we define $\bar{W}_{t}(t)$ as a linear interpolation of the positions $\bar{W}_{t}(k d t)$ and $\bar{W}_{t}((k+1) d t)$, this is obviously an approximation of a Wiener process.

More precisely the following result holds

## Theorem

Let $T>0, N>0$ and $d t=T / N$. Let us denote by $\bar{P}_{t}$ the law of the process $\bar{W}_{t}$ ( $\bar{P}_{t}$ is therefore a probability on the canonical space $\left.\mathscr{C}=\mathscr{C}\left([0, T], \mathbb{R}^{m}\right)\right)$. Then $\bar{P}_{t}$ converges weakly to the Wiener measure $P^{W}$.

The Theorem ensures that if $\phi: \mathscr{C} \rightarrow \mathbb{R}$ is a bounded continuous function, then

$$
\lim _{t \rightarrow 0} E\left[\phi\left(\bar{W}_{t}\right)\right]=E[\phi(W)]
$$

Of course it would be very important to know how close to the true value $E[\phi(W)]$ the approximation $E\left[\phi\left(\bar{W}_{t}\right)\right]$ is. In other words, it would be very important to determine the speed of convergence, as $t \rightarrow 0$, of the estimator obtained by the simulated process.

## A.1.2 Application

In this section we will use what was previously described in order to simulate a discretized Wiener process. In particular, the simulation will be divided into several steps:

- calculate and show the trajectory of a generalized Wiener process $W(t)$;
- fixed an instant of time $t$, show the trajectories of motion as the number of particles involved varies;
- construct a histogram showing the distribution trend for a fixed time t at the arrival points $x$;
- varying the number of particles in different instants of time, verify that the distribution obtained from the arrival histogram is Gaussian;
- calculate the variance $\sigma^{2}$ and show that it has a proportional trend with respect to time $t$, with the proportionality coefficient given by the diffusion coefficient $D$;
- using the scale law obtained from the variance calculation verify that, as time varies, all distributions tend to the same Gaussian curve.


## Trajectory of a Wiener process

To simulate the process was built a function that, starting from known starting point, first stratifies the terminal value of a Wiener process, and then samples the process from beginning to end by drawing Gaussian samples.

The stratification process assumes that each path is associated with a single stratified terminal value such that the number of paths is equal to the number of strata. This technique is called proportional sampling.

In particular the function requires knowledge of the entire sequence of sample times.

The function implements proportional sampling by partitioning the unit interval into bins of equal probability. The inverse cumulative distribution function of a standard $N(0,1)$ Gaussian distribution then transforms these stratified uniform draws

Finally, the resulting stratified Gaussian draws are scaled by $\sqrt{2 \cdot D} \sqrt{d t}$ to stratify the terminal value of the Wiener process.

```
%The function 'wp' creates and displays a Weiner
    Process. In particular the function first
    stratifies the terminal value of the process, and
    then samples the process from beginning to end by
    drawing Gaussian samples.
2
%INPUT: W0=starting point;
% n=number of particles;
```

```
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient.
%OUTPUT: W=standard Wiener process;
% t=vector of observation instants.
function [W, t]=wp(W0, n,T,N,mu, sigma,D)
dt=T/N; %Discretization
t=(0:dt:T); %Vector of sample times associated with
    all simulated paths
k=sqrt(2*D);
%Construction of the Weiner Process:
for i=1:n
    W(i, 1) =W0;
    for j=1:N
        W(i, j + 1) = W(i, j) +k*sqrt(dt)*normrnd(mu, sigma)
                ;
            %The Matlab function 'normrnd' generates a
                random number from the normal distribution
                with mean parameter mu and standard
                deviation parameter sigma.
    end
end
%Verify the stratification: recreate the uniform draws
    with proportional sampling
U = ((1:n)' - 1 + rand(n,1))/n;
```

${ }_{34}$ \%Plot the output paths:

```
plot(t,W);
```

hold ('on') ;
xlabel('Time'), ylabel('Process State');
title ('Terminal Stratification: Wiener Process');
\%Plot the terminal values on the same figure:
plot(T, WT, '. black', T, WT, 'o black');
hold ('off');
end


Figure A.2: Trajectories for $W 0=0, n=5, T=1, N=500, m u=0$, sigma $=1, D=1$.


Figure A.3: Trajectories for $W 0=0, n=1000, T=1, N=500, m u=0$, sigma $=1, D=1$

## Distribution Trend

In this section we will study the distribution of the process considered.
In particular we use a function that constructs an histogram for a fixed instant of time $t$ at the arrival point $x$.

```
```

%The function 'dis_hist' explores the distribution of

```
```

%The function 'dis_hist' explores the distribution of
points and verifies that it behaves like a Gaussian
points and verifies that it behaves like a Gaussian
3 %INPUT: W0=starting point;
3 %INPUT: W0=starting point;
% n=number of particles;
% n=number of particles;
% T=period of observation;
% T=period of observation;
% N=discretization step;
% N=discretization step;
7% mu=mean;
7% mu=mean;
8% sigma=standard deviation;
8% sigma=standard deviation;
, % D=diffusion coefficient;
, % D=diffusion coefficient;
% h=number of histogram columns.
% h=number of histogram columns.
11
11
12 %OUTPUT: x=arrival points vector.
12 %OUTPUT: x=arrival points vector.
13
13
14 function [x] = dis_hist(W0, n,T,N,mu, sigma,D, h)
14 function [x] = dis_hist(W0, n,T,N,mu, sigma,D, h)
15
15
16 figure(1);
16 figure(1);
17 [W, t]=wp(W0, n,T,N, mu, sigma,D);
17 [W, t]=wp(W0, n,T,N, mu, sigma,D);
18 %[counts,centers]=hist(x,h)
18 %[counts,centers]=hist(x,h)
19 %counts=counts of the number of elements in each
19 %counts=counts of the number of elements in each
column, returned as a row vector.
column, returned as a row vector.
20 %centers=columns centers, returned as a numeric row
20 %centers=columns centers, returned as a numeric row
vector.
vector.
2 1

```
2 1
```

```
2
```

2
[m,x] = hist(W(:,end),h); %Empirical histogram

```
[m,x] = hist(W(:,end),h); %Empirical histogram
```

```
    function
m_norm = (m ./length(W(:,end))) ./(x(2)-x(1)); %
    Normalize histogram function
bar(x,m_norm); %The Matlab function 'bar, creates an
        %histogram plot
hold('on');
%Plot the shape of empirical distribution on the
    histogram:
plot(x,m_norm, 'r');
xlabel('x'), ylabel('Distribution');
title('Empirical Distribution');
figure(2);
pd = fitdist(W(:,end),'Normal'); %The Matlab function
    'fitdist, creates a probability distribution object
        by fitting the distribution specified to the data
        in column vector.
y = pdf(pd, x); %The Matlab function 'pdf' returns the
        probability distribution function of the pd,
        evaluated at the values in x.
%Plot the shape of the empirical distribution on the
%Gaussian one:
semilogy(x,y,'b');
hold ('on');
plot(x,m_norm, 'r');
title('Comparison with Gaussian');
end
```



Figure A.4: Histogram plot of $W(t)$ with shape of the empirical distribution. $W 0=0, n=10000, T=10, N=500$, $m u=0$, sigma $=1, D=1, h=100$.


Figure A.5: The distribution obtained from the arrival histogram is Gaussian. $W 0=0, n=10000, T=10, N=500$, $m u=0$, sigma $=1, D=1, h=100$.

After verifying the result for the vector of arrival points, varying the number of particles $n$ and the time $T$, we show how the empirical distribution step by step becomes more and more similar to the Gaussian distribution.

For this purpose we use the following function which creates a single plot in which both the histogram, the empirical distribution and the Gaussian one appear at each execution.

```
%The function 'tot_hist' creates a single plot in %
    which both the histogram, the empirical
    distribution and the Gaussian one appear.
2
%INPUT: W0=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% h=number of histogram columns.
%OUTPUT: pd=distribution.
    function [pd] = tot_hist(W0, n,T,N,mu,sigma,D,h)
15
16 [W, t]=wp(W0,n,T,N,mu, sigma,D);
17 %[counts,centers]=hist (x,h)
%counts=counts of the number of elements in each
    column, returned as a row vector.
19 %centers=columns centers, returned as a numeric row
        vector.
```

```
[m,x] = hist(W(:, end),h); %Empirical histogram
    function
m_norm = (m ./ length(W(:, end))) ./( x (2)-x (1)); %
    Normalize histogram function
bar(x,m_norm); %The Matlab function 'bar' creates an
    %histogram plot
hold('on');
%Plot the shape of empirical distribution on the
    histogram:
plot(x,m_norm,'r', 'Linewidth', 1);
hold ('on');
pd= fitdist(W(:, end),'Normal'); %The Matlab function
    'fitdist', creates a probability distribution object
        by fitting the distribution specified to the data
    in column vector.
y = pdf(pd, x); %The Matlab function 'pdf, returns the
        probability distribution function of the pd,
    evaluated at the values in x.
%Plot the shape of the Gaussian distribution on the
    histogram:
semilogy(x,y,'Linewidth', 1.5)
xlabel('x'), ylabel('Distribution')
title('Distributions Trend');
end
```



Figure A.6: STEP 1:W0 $=0, n=100, T=100, N=500, m u=0$, sigma $=1, D=1, h=100$


Figure A.7: STEP 2:W0 $: W, n=150, T=150, N=500, m u=0$, sigma $=1, D=1, h=100$


Figure A.8: STEP 3:W0 $=0, n=300, T=300, N=500, m u=0$, sigma $=1, D=1, h=100$


Figure A.9: STEP 4:W0 $=0, n=600, T=600, N=500, m u=0$, sigma $=1, D=1, h=100$


Figure A.10: STEP 5: $W 0=0, n=1000, T=1000, N=500, m u=0$, sigma $=1, D=1, h=100$


Figure A.11: STEP 6: $W 0=0, n=10000, T=1000, N=500, m u=0$, sigma $=1, D=1, h=100$

## Variance Scale Law

In this section we intend to study the scale law of variance, for this purpose we calculate the variance and build its graph to verify the trend.

In particular we use the function 'var_trend' that first calculates the variance both numerically and analytically, the numerical error that derives from these different approaches and then graphically shows the trend.

```
%The function 'var_trend, calculates the variance both
        using the Matlab function 'var' and applying the
        mathematical formula explicitly. Then the function
        makes the plot of the numerical variance trend.
%INPUT: W0=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient.
%OUTPUT: V=variance;
% err=numerical error.
function [V,err] = var_trend(W0, n,T,N,mu,sigma,D)
[W, t]=wp(W0,n,T,N,mu,sigma,D);
V=var(W(:,N),1); %The Matlab function 'var' returns
    the variance of the elements of W
```

2

19
$\mathrm{M}=\boldsymbol{m e a n}(\mathrm{W}(:, \mathrm{N}))$; \%Mean value of all observations
$\mathrm{V} 1=(1 / \mathrm{N}) * \mathbf{S u m}\left(\mathbf{a b s}(\mathrm{~W}(:, \mathrm{N})-\mathrm{M}) .^{\wedge} 2\right) ; \%$ Variance calculated explicitly
err=norm (V-V1); \%Numerical error
\%Plot of the variance scale law:
V_tot=var (W (: , : ) , 1) ;
plot(t, V_tot, 'Linewidth', 1)
xlabel('Time (t)'), ylabel ('Variance (V)');
title ('Variance Scale Law');
hold ('on')
V_num $=2 * D * t$;
plot(t, V_num, 'Linewidth', 1)
end

By varying the number of particles it is also possible to notice how the rumors decrease:


Figure A.12: Variance trend with $W 0=0, n=2000, T=1, N=500, m u=0$, sigma $=1, D=1$ and $E r r=2.996 e-04$.


Figure A.13: Variance trend with $W 0=0, n=4000, T=1, N=500, m u=0$, sigma $=1, D=1$ and Err $=2.611 e-04$.


Figure A.14: Variance trend with $W 0=0, n=6000, T=1, N=500, m u=0$, sigma $=1, D=1$ and $E r r=4.681 e-04$.


Figure A.15: Variance trend with $W 0=0, n=8000, T=1, N=500, m u=0$, sigma $=1, D=1$ and Err $=3.157 e-04$.


Figure A.16: Variance trend with $W 0=0, n=10000, T=1, N=500, m u=0, \operatorname{sigma}=1, D=1$ and $E r r=2.518 e-04$.

Using the function 'var_trend' and modifying the diffusion coefficient D it is possible to obtain the following graph showing the trend of the variance:


Figure A.17: Variance trend with $W 0=0, n=8000, T=10, N=500, m u=0$, sigma $=1$ and $D 1=0, D 2=0.2, D 3=$ $0.4, D 4=0.6, D 5=0.8, D 6=1$.

## Gaussian respect to the Variance

In this section, using the law of scale obtained from the calculation of variance, we intend to verify that, as time varies, all distributions tend to the same Gaussian curve.

In particular, for this purpose we used the function 'gauss_var' that studies the behavior of the Gaussian curve with respect to $\sigma^{2}$.

```
% The function 'gauss_var, studies the trend of the
    Gaussian with respect to the calculated variance.
2
3 %INPUT: W0=starting point;
```

```
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% h= number of histogram columns.
%OUTPUT: sigma_num=empirical standard deviation;
% X=Weiner process respect to the empirical
% standard deviation.
function [sigma_num,X]= gauss_var1(W0, n,T,N,mu, sigma,D,
        h)
18 [W,~}]=wp(W0,n,T,N,mu, sigma,D)
sigma_num=sqrt(var(W(:,:),1)); %Standard deviation
X=W/sigma_num; %Weiner process respect to the standard
    deviation
    [m,x] = hist(X,h); %Empirical histogram function
m_norm = (m./length(X(:, end))) ./(x(2)-x(1)); %
        normalize histogram function
24 pd = fitdist(X,'Normal');%The Matlab function 'fitdist
    , creates a probability distribution object by
        fitting the distribution specified to the data in
        column vector.
26 y = pdf(pd, x); %the Matlab function 'pdf, returns the
        probability distribution function of the pd,
    evaluated at the values in x.
```

17
23
25

27
${ }_{2} 8$ \%Plot the shape of the Gaussian distribution with respect to
\%the variance, as time varies:
plot(x,y, 'Linewidth , , 1)
xlabel('W(t)/sigma(t)'), ylabel('sigma(t)*G')
title ('Gaussian respect to the Variance');
hold ('on') ;
\%Plot the shape of the empirical distribution
plot (x,m_norm) ;
end


Figure A.18: Gaussian distribution with respect to the variance, as time varies. $W 0=0, n=8000, N=500$, $m u=0$, sigma $=1, D=1, h=100$ and $T 1=2, T 2=4, T 3=6, T 4=8, T 5=10$.

## Remark

As expected from the theoretical results, all curves "collapse" on the same Gaussian.

## A. 2 Ornstein-Uhlenbeck Process

## A.2.1 General facts

In many stochastic processes that appear in applications, their statistics remain invariant under time translations. Such stochastic processes are called stationary.

It is possible to develop a quite general theory of stochastic processes that enjoy this symmetry property. It is useful to distinguish between:

- stochastic processes for which all finite dimensional distribution (FDDs) are translation-invariant (strictly stationary processes);
- processes for which this translation invariance holds only for the first two moments (weakly stationary processes).

Let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space. Let $X_{t}, t \in T$ (with $T=\mathbb{R}$ or $\mathbb{Z}$ ), be a real-valued random process on this probability space with finite second moment $\mathbb{E}\left|X_{t}\right|^{2}<+\infty$ (i.e. $X_{t} \in L^{2}(\Omega, \mathbb{P})$ for all $t \in T$ ). Assume that is strictlu stationary. Then

$$
\mathbb{E}\left(X_{t+s}\right)=\mathbb{E} X_{t}, \quad s \in T
$$

from which we conclude that $\mathbb{E} X_{t}$ is constant, and

$$
\mathbb{E}\left(\left(X_{t_{1}+s}-\mu\right)\left(X_{t_{2}+s}-\mu\right)\right)=\mathbb{E}\left(\left(X_{t_{1}}-\mu\right)\left(X_{t_{2}}-\mu\right)\right), \quad s \in T
$$

implies that the covariance function depends on the difference between the two times $t$ and $s$

$$
C(t, s)=C(t-s)
$$

This motivates the following definition

A stochastic process $X_{t} \in L^{2}$ is called second-order stationary, widesense stationary or weakly stationary if the first moment $\mathbb{E} X_{t}$ is a constant and the covariance function $\mathbb{E}\left(X_{t}-\mu\right)\left(X_{s}-\mu\right)$ depends only on the difference $t-s$ :

$$
\mathbb{E} X_{t}=\mu, \quad \mathbb{E}\left(\left(X_{t_{1}}-\mu\right)\left(X_{t_{2}}-\mu\right)\right)=C(t-s)
$$

- the constant $\mu$ is the expectation of the process $X_{t}$;
- the function $C(t)$ is the covariance, also called autocovariance or autocorrelation function of $X_{t}$

Notice that a strictly stationary process with finite second moment is also stationary in the wide sense. The converse is not true, in general. It is true, however, for Gaussian processes.

Continuity properties of the covariance function are equivalent to continuity properties of the paths of $X_{t}$ in the $L^{2}$ sense, i.e.

$$
\lim _{h \rightarrow 0} \mathbb{E}\left|X_{t+h}-X_{t}\right|^{2}=0
$$

In particular, the following result holds

Lemma Assume that the covariance function $C(t)$ of a second-order stationary process is continuous at $t=0$. Then it is continuous for all $t \in \mathbb{R}$. Furthermore, the continuity of $C(t)$ is equivalent to the continuity of the process $X_{t}$ in the $L^{2}$ sense.

The Fourier transform of the covariance function of a second-order stationary process always exists. This enables us to study second-order stationary processes using tools from Fourier analysis.

The autocorrelation function of a second-order stationary process enables us to associate a timescale to $X_{t}$, the correlation time $\tau_{c o r}$ :

$$
\tau_{c o r}=\frac{1}{C(0)} \int_{0}^{\infty} C(\tau) d \tau=\frac{1}{\mathbb{E}\left(X_{0}^{2}\right)} \int_{0}^{\infty} \mathbb{E}\left(X_{\tau} X_{0}\right) d \tau
$$

The slower the decay of the correlation function, the larger the correlation time. Note that when the correlations do not decay sufficiently fast, so that $C(t)$ is not integrable, then the correlation time will be infinite.

Example Consider a mean-zero second-order stationary process with correlation function

$$
\begin{equation*}
C(t)=C(0) e^{-\alpha|t|}, \quad \alpha>0 \tag{A.1}
\end{equation*}
$$

we will write $C(0)=D / \alpha$ where $D>0$. The correlation time is

$$
\tau_{c o r}=\int_{0}^{\infty} e^{-\alpha t} d \tau=\alpha^{-1}
$$

A real-valued Gaussian stationary process defined on $\mathbb{R}$ with correlation function given by (1) is called a stationary Ornstein-Uhlenbeck Process.
The Ornstein-Uhlenbeck process $X_{t}$ can be used as a model for the velocity of a Brownian particle. It is of interest to calculate the statistics of the position of the Brownian particle.

## A.2.2 Definition

The stationary Ornstein-Uhlenbeck process, that was introduced earlier, can be defined through the Brownian motion via a time change.

The process can be considered to be a modification of the random walk in continuous time, or Wiener process, in which the properties of the process have been changed so that there is a tendency of the walk to move back towards a central location, with a greater attraction when the process is further away from the center.

Definition The Ornstein-Uhlenbeck process $X_{t}$ is defined by the following stochastic differential equation:

$$
d X_{t}=-\theta x_{t} d t+\sigma d W_{t}
$$

where $\theta>0$ and $\sigma>0$ are parameters and $W_{t}$ denotes the Wiener process. The stochastic differential equation for $X_{t}$ can be formally solved by variation of parameters. We get

$$
X_{t}=X_{0} e^{-\theta t}+\mu\left(1-e^{-\theta t}\right)+\sigma \int_{0}^{t} e^{-\theta(t-s)} d W_{s}
$$

The Ornstein-Uhlenbeck process is an example of a Gaussian process that has a bounded variance and admits a stationary probability distribution, in contrast to the Wiener process.
The difference between the two is in their "drift" term: for the Wiener process the drift term is constant, whereas for the Ornstein-Uhlenbeck process it is dependent on the current value of the process.

## Remarks

- Brownian motion and the Ornstein-Uhlenbeck process are examples of a diffusion process: a continuous-time Markov process with continuous paths;
- if the initial condition of the Ornstein-Uhlenbeck process is distributed according to the invariant measure, then the Ornstein-Uhlenbeck process is a stationary Gaussian process. Let $X_{t}$ denote the one-dimensional OrnsteinUhlenbeck process with $X_{0} \sim \mathscr{N}(0, D / \alpha)$. Then $X_{t}$ is a mean-zero Gaussian second-order stationary process on $[0, \infty)$ with correlation function and spectral density:

$$
R(t)=\frac{D}{\alpha} e^{-\alpha t} \quad f(x)=\frac{D}{\pi} \frac{1}{x^{2}+(\alpha)^{2}}
$$

- the Ornstein-Uhlenbeck process is the only real-valued mean-zero Gaussian second-order stationary Markov process with continuous paths defined on $\mathbb{R}$. A few paths of the stationary Ornstein-Uhlenbeck process are presented in the follow figure


## A.2.3 Simulation

In this section we will use what was previously described in order to simulate an Ornstein-Uhlenbeck process. In particular, the simulation will be divided into several steps:

- calculate and show the trajectory of an Ornstein-Uhlenbeck process $X(t)$;
- verify that for $\tau \rightarrow+\infty$ we get the Wiener process;


Figure A.19: Sample paths of the Ornstein-Uhlenbeck process

- show that the variance of the process is such that, after a certain instant, it saturates then becomes constant;
- calculate the correlation function and show that it has an exponential trend.


## Trajectory of an Ornstein-Uhlenbeck process

For simulation was built a function that, from a known starting point and a known sequence of sample times, implements the Ornstein-Uhlenbeck process.

In particular the noise part of the process is due to the factor

$$
d W_{n}=\sqrt{2 D} \sqrt{d t} N(0,1)
$$

therefore the process will be

$$
X_{n}=X_{n-1}-\frac{X_{n-1}}{\tau} d t+\sqrt{2 D} d W_{n}
$$

1
\%The function 'oup' creates and displays an OrnsteinUhlenbeck process.
UnIenbeck process.

$$
3
$$

4 \%INPUT: X0=starting point (Ornstein-Uhlenbeck);
5 \% W0=starting point (Wiener);
6 \% n=number of particles;
7 \% $\mathrm{T}=$ period of observation;
8 \% N=discretization step;
\% mu=mean;
\% sigma=standard deviation;
\% D=diffusion coefficient;
\% tau=correlation time.
13
14
15

16
\%OUTPUT: X=Ornstein-Uhlenbeck process;
\% $\quad$ =vector of observation instants.
function $[\mathrm{X}, \mathrm{t}]=\operatorname{oup}(\mathrm{X} 0, \mathrm{~W} 0, \mathrm{n}, \mathrm{T}, \mathrm{N}, \mathrm{mu}$, sigma, $\mathrm{D}, \mathrm{tau})$
$\mathrm{dt}=\mathrm{T} / \mathrm{N} ; \%$ Discretization
$\mathrm{t}=(0: \mathrm{dt}: \mathrm{T}) ; \%$ Vector of sample times associated with all simulated paths
\%Construction of the Ornstein-Uhlenbeck Process:
for $\mathrm{i}=1$ : n
$\mathrm{X}(\mathrm{i}, 1)=\mathrm{X} 0$ ( i$)$;
for $\mathrm{j}=1: \mathrm{N}$
$X(i, j+1)=X(i, j)-(X(i, j) / t a u) * d t+s q r t(2 * D) *$
$\qquad$ normrnd (mu, sigma) $* \mathbf{s q r t}(\mathrm{dt})$;

# \%The Matlab function 'normrnd' generates a random 

\%number from the normal distribution with mean parameter mu and standard deviation parameter sigma. end
end
\%Plot the output paths:
plot(t, X, 'b');
xlabel('Time'), ylabel('Process State');
title ('Terminal Stratification: Ornstein-Uhlenbeck
Process');
\%Verify that for $\tau \rightarrow \infty$ we have the Wiener process:
$[\mathrm{W}, \mathrm{t}]=\mathrm{wp}(\mathrm{W} 0, \mathrm{n}, \mathrm{T}, \mathrm{N}, \mathrm{mu}$, sigma, D$)$;
hold ('on') ;
plot(t, W, 'r');
end


Figure A.20: Trajectories for $X 0=\operatorname{randn}(n, 1), n=5, T=1, N=500, m u=0, \operatorname{sigma}=1, D=1, \tau=1$.


Figure A.21: Trajectories for $X 0=\operatorname{randn}(n, 1), n=1000, T=1, N=500, m u=0, \operatorname{sigma}=1, D=1, \tau=1$.

Using the same function, adding $W 0$ as input, it is possible to verify that for $\tau \rightarrow \infty$ we have the Wiener Process again:


Figure A.22: Trajectories for $X 0=0, W 0=0, n=1000$ for Ornstein - Uhlenbeck (blue), $n=3000$ for Wiener $($ red $), T=1, N=500, m u=0$, sigma $=1, D=1, \tau=\infty$.

## Distribution Trend

In this section we will study the distribution of the process considered.
In particular we use a function that constructs an histogram for a fixed instant of time $t$ at the arrival point $x$.

```
1 %The function 'dis_hist, explores the distribution of
    points and verifies that it behaves like a Gaussian
2
3 %INPUT: X0=starting point;
%. n=number of particles;
5%. T=period of observation;
6%.N=discretization step;
```

```
%. mu=mean;
%. sigma=standard deviation;
%. D=diffusion coefficient;
%. h=number of histogram columns.
%OUTPUT: x=arrival points vector.
    function [x] = dis_hist(X0,n,T,N,mu, sigma,D, tau,h)
    figure(1);
    [X, t]=oup(X0,n,T,N,mu,sigma,D,tau);
    %[counts,centers]=hist(x,h)
    %counts=counts of the number of elements in each
        column, returned as a row vector.
    %centers=columns centers, returned as a numeric row
        vector.
    [m,x] = hist(X(:, end),h); %Empirical histogram
        function
    m_norm = (m./length(X(:, end)))./(x(2) x (1));
        %Normalize histogram function
bar(x,m_norm); %The Matlab function bar creates
        an
            %histogram plot
hold ('on') ;
%Plot the shape of empirical distribution on the
        histogram:
plot(x,m_norm,'r');
xlabel('x'), ylabel('Distribution');
title('Empirical Distribution');
```

${ }_{36} y=\operatorname{pdf}(p d, x) ;$ The Matlab function 'pdf, returns the probability distribution function of the pd, evaluated at the values in $x$.

37
${ }_{38}$ \%Plot the shape of the empirical distribution on the
\%Gaussian one :
semilogy ( $\mathrm{x}, \mathrm{y},{ }^{\prime} \mathrm{b}^{\prime}$ ) ;
hold ('on') ;
plot(x, m_norm , 'r');
title ('Comparison with Gaussian');
end


Figure A.23: Histogram with shape of the empirical distribution for $X 0=$ randn, $n=10000, T=10, N=500$, $m u=0$, sigma $=1, D=1, \tau=1, \mathrm{~h}=100$.


Figure A.24: Distribution obtained from the arrival histogram for $X 0=r a n d n, n=10000, T=10, N=500$, $m u=0$, sigma $=1, D=1, \tau=1, \mathrm{~h}=100$.

After verifying the result for the vector of arrival points, varying the number of particles $n$ and the time $T$, we show how the empirical distribution step by step becomes more and more similar to the Gaussian distribution.

For this purpose we use the following function which creates a single plot in which both the histogram, the empirical distribution and the Gaussian one appear at each execution.

```
%The function 'tot_hist', creates a single plot in %
    which both the histogram, the empirical
    distribution and the Gaussian one appear.
2
%INPUT: W0=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% h=number of histogram columns.
%OUTPUT: pd=distribution.
    function [pd]= tot_hist(W0, n,T,N,mu, sigma,D,h)
    [W, t ] = wp (W0, n,T,N,mu, sigma,D);
%[counts, centers]=hist (x,h)
%counts=counts of the number of elements in each
    column, returned as a row vector.
%centers=columns centers, returned as a numeric row
        vector.
```

```
[m,x] = hist(W(:, end),h); %Empirical histogram
    function
m_norm = (m ./ length(W(:, end))) ./( x (2)-x (1)); %
    Normalize histogram function
bar(x,m_norm); %The Matlab function 'bar' creates an
    %histogram plot
hold('on');
%Plot the shape of empirical distribution on the
    histogram:
plot(x,m_norm,'r', 'Linewidth', 1);
hold ('on');
pd= fitdist(W(:, end),'Normal'); %The Matlab function
    'fitdist', creates a probability distribution object
        by fitting the distribution specified to the data
    in column vector.
y = pdf(pd, x); %The Matlab function 'pdf, returns the
        probability distribution function of the pd,
    evaluated at the values in x.
%Plot the shape of the Gaussian distribution on the
    histogram:
semilogy(x,y,'Linewidth', 1.5)
xlabel('x'), ylabel('Distribution')
title('Distributions Trend');
end
```



Figure A.25: STEP 1: $X 0=$ randn, $n=100, T=100, N=500$, mu $=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.26: STEP 2: $X 0=$ randn, $n=150, T=150, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.27: STEP 3: $X 0=$ randn, $n=300, T=300, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.28: STEP 4: $X 0=$ randn, $n=600, T=600, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.29: STEP 5: $X 0=$ randn, $n=1000, T=1000, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.30: STEP 6:X $0=$ randn, $n=10000, T=1000, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$

## Variance Scale Law

In this section we intend to study the scale law of variance, for this purpose we calculate the variance and build its graph to verify the trend. In particular we use the function 'var_trend' that calculates the variance numerically using the Matlab function 'var'.

We remember that we want to show that the variance of the process is such that, after a certain instant, it saturates then becomes constant.

```
% The function 'var_trend' calculates the variance
        using the Matlab function 'var'. Then the function
        makes the plot of the variance trend.
% INPUT: X0=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% tau=correlation time.
% OUTPUT: V=variance.
    function [V] = var_trend(X0,n,T,N,mu,sigma,D,tau)
%Recall the Ornstein-Uhlenbeck Process:
    [X,t] = oup(X0,W0,n,T,N,mu,sigma,D,tau)
V=var(X(:,:),1); %The Matlab function 'var' returns
```

```
        the variance of the elements of X
%Plot of the variance trend:
plot(t, V_tot)
xlabel('Time (t)'), ylabel('Variance (V)');
title('Variance Trend');
end
```

20


Figure A.31: Trend for $X 0=\operatorname{randn}, T=10, N=5000, m u=0$, sigma $=1, D=1, \tau=1$.

## Correlation Function

We use the following functions as $\tau$ and $D$ vary, in order to find the relationship between variance, correlation time and diffusion coefficient

```
%The script 'vartau, calculates the variance as tau
        varies
2
tau1=1; [Vt1]= var_trend(X0, n,T,N,mu, sigma,1,tau 1);
hold 'on'
tau2=2; [Vt2] = var_trend (X0, n,T,N,mu, sigma, 1, tau2 );
6 tau 3 = 3; [Vt3] = var_trend (X0, n,T,N,mu, sigma, 1, tau 3);
7 tau4 =4; [Vt4] = var_trend(X0,n,T,N,mu,sigma, 1, tau4);
8 tau5 = 5; [Vt5] = var_trend(X0, n,T,N,mu, sigma,1, tau5 );
9 tau6=6; [Vt6] = var_trend(X0, n,T,N,mu, sigma,1, tau6);
tau7 = 7; [Vt7] = var_trend (X0, n,T,N,mu, sigma,1, tau7 );
11 tau8=8; [Vt8] = var_trend(X0, n,T,N,mu, sigma,1,tau8);
12 tau9 =9; [Vt9] = var_trend(X0, n,T,N,mu, sigma,1, tau9 );
13 tau 10=10; [Vt10]= var_trend (X0, n,T,N, mu, sigma, 1, tau 10
    );
```

```
1 %The script 'vardiff, calculates the variance as D
        varies
2
3 n=10000; X0=randn; T=10; N=500; mu=0; sigma=1; tau=1;
4 D1=1; [V1] = var_trend (X0,n,T,N,mu,sigma,D1,tau);
s hold 'on,
6 D2=2; [V2] = var_trend(X0,n,T,N,mu,sigma,D2,tau);
7 D3=3; [V3] = var_trend(X0,n,T,N,mu,sigma,D3,tau);
8 D4=4; [V4] = var_trend(X0,n,T,N,mu,sigma,D4,tau);
9 D5=5; [V5] = var_trend(X0,n,T,N,mu,sigma,D5,tau);
```

${ }_{10}$ D6=6; [V6] = var_trend (X0, n, T,N, mu, sigma, D6,tau);
${ }^{11}$ D7=7; [V7] = var_trend (X0, n, T,N, mu, sigma, D7,tau);
${ }_{12} \mathrm{D} 8=8 ;[\mathrm{V} 8]=$ var_trend $(\mathrm{X} 0, \mathrm{n}, \mathrm{T}, \mathrm{N}, \mathrm{mu}$, sigma, $\mathrm{D} 8, \mathrm{tau})$;
${ }_{13} \mathrm{D} 9=9 ;$ [V9] $=$ var_trend $(\mathrm{X} 0, \mathrm{n}, \mathrm{T}, \mathrm{N}, \mathrm{mu}$, sigma, D9,tau);
${ }_{14} \mathrm{D} 10=10 ;[\mathrm{V} 10]=$ var_trend $(\mathrm{X} 0, \mathrm{n}, \mathrm{T}, \mathrm{N}, \mathrm{mu}$, sigma, D10,tau);


Figure A.32: Variance as $D$ varies for $X 0=$ randn, $n=8000, T=10, N=5000, m u=0$, sigma $=1, \tau=1$ and $D=0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2$.


Figure A.33: Variance as tau varies for $X 0=$ randn, $n=8000, T=10, N=5000, m u=0$, sigma $=1, D=1$ and $\tau=0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2$.


Figure A.34: Variance for $X 0=$ randn, $n=10000, T=10, N=5000, m u=0$, sigma $=1, \tau=[0.2: 0.2: 2]$, $D=[0.2: 0.2: 2]$.

The above figure was obtained using the following script

```
1 \%This script displays the relationship between the
        variance and tau and the variance and D
2
3 tau \(=[\) tau 1 tau2 tau 3 tau 4 tau 5 tau 6 tau 7 tau 8 tau 9
    tau10];
\({ }_{4} \mathrm{~V}\) _t \(=[\mathrm{Vt} 1(1\), end \() \mathrm{Vt2}(1\), end \() \mathrm{Vt} 3(1\), end \() \mathrm{Vt} 4(1\), end \() \mathrm{Vt} 5\)
    (1,end) Vt6(1,end) Vt7(1,end) Vt8(1,end) Vt9(1,end)
        Vt10 (1, end) ];
plot (tau, V_t)
D=[D1 D2 D3 D4 D5 D6 D7 D8 D9 D10];
\(\mathrm{V}=[\mathrm{V} 1(1\), end) \(\mathrm{V} 2(1\), end) \(\mathrm{V} 3(1\), end) \(\mathrm{V} 4(1\), end) \(\mathrm{V} 5(1\), end \()\)
    V6(1, end) V7(1, end) V8(1, end) V9(1, end) V10(1, end)
    ];
plot (D,V)
```

We then used the following function to calculate the empirical correlation (E) and the analytic correlation (C). In particular

$$
E\left[X_{t} X_{s}\right]=\frac{1}{N-1} \sum_{i=1}^{N}\left(\frac{X_{t_{i}}-\bar{X}_{t}}{\sigma_{t}^{2}}\right)\left(\frac{X_{s_{i}}-\bar{X}_{s}}{\sigma_{s}^{2}}\right)
$$

and

$$
C=e^{-|t-s| / \tau} ;
$$

```
% The function 'correlation', calculates the
    correlations E and C. Then the function makes the
    plot of both trends.
% INPUT: X0=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% tau=correlation time.
% OUTPUT: E=empirical correlation;
% C=analytic correlation.
    function [E,C] = correlation( X0, n,T,N,mu, sigma,D,tau)
    dt=T/N; %Discretization
    t=(0:dt:T); %Vector of sample times associated with
        all simulated paths
    %Construction of the Ornstein-Uhlenbeck Process:
    for i=1:n
```

$$
\mathrm{X}(\mathrm{i}, 1)=\mathrm{X} 0(1) ;
$$

for $\mathrm{j}=1: \mathrm{N}$ $X(i, j+1)=X(i, j)-(X(i, j) / t a u) * d t+s q r t(2 * D) *$ normrnd (mu, sigma) $*$ sqrt (dt) ; \%The Matlab function 'normrnd' generates a random number from the normal distribution with mean parameter mu and standard deviation parameter sigma.
end
end
\%variance
$\mathrm{V}=\operatorname{var}(\mathrm{X}(:,:), 1) ;$
\%empirical correlation
$\mathrm{E}=\boldsymbol{z e r o s}(1, \mathrm{~N}+1)$;
$\mathrm{m}=$ mean ( X );
for $i=1: N+1$
for $\mathrm{j}=1$ : n
$\mathrm{E}(\mathrm{i})=\mathrm{E}(\mathrm{i})+(\mathrm{X}(\mathrm{j}, \mathrm{i})-\mathrm{m}(\mathrm{i})) . *(\mathrm{X}(\mathrm{j}, 250)-\mathrm{m}(250))$;
end
$\mathrm{E}(\mathrm{i})=\mathrm{E}(\mathrm{i}) . /(\mathrm{V}(\mathrm{i}) . * \mathrm{~V}(250))$;
$\mathrm{E}(\mathrm{i})=\mathrm{E}(\mathrm{i}) . /(\mathrm{n}-1)$;
end
plot(t, E)
\%analytic correlation
$\mathrm{C}=\boldsymbol{\operatorname { e x p }}(-\mathbf{a b s}(\mathrm{t}-5) . / \mathrm{tau})$;
hold on
plot(t,C)

49 xlabel('Time'), ylabel('Correlation')
title('Correlation Trend')
end


Figure A.35: Correlation for $X 0=$ randn, $n=10000, T=10, N=500, m u=0, \operatorname{sigma}=1, D=1, \tau=1$.

## A. 3 Langevin Equation

The theory of Brownian motion is perhaps the simplest approximate way to treat the dynamics of nonequilibrium systems. The fundamental equation is called the Langevin equation, it contain both frictional forces and random forces.

The random motion of a small particle immersed in a fluid with the same density as the particle is called Brownian motion.
While the motion of a dust particle performing Brownian motion appears to be quite random, it must nevertheless be describable by the same equation of motion as is any other dynamical system. In classical mechanics these are Newton's or Hamiltons equations. For simplicity we will consider motion in one dimension. Newtons equation of motion for the particle in a fluid medium is

$$
m \frac{d v(t)}{d t}=F(t)
$$

where $m$ is the mass of the particle, $x(t)$ the position, $v(t)$ the velocity and $F(t)$ the total instantaneous force on the particle at time $t$. This force is due to the interaction of the Brownian particle with the surrounding medium. It is usually not practical or even desirable to look for an exact expression for $\mathrm{F}(\mathrm{t})$. Experience tells us that in typical cases this force is dominated by a friction force $-\gamma \nu(t)$ proportional to the velocity of the Brownian particle. We also expect a random force $\xi(t)$ due to random density fluctuations in the fluid. The equations of motion of the Brownian particle are:

$$
\begin{align*}
& \frac{d x(t)}{d t}=v(t)  \tag{A.2}\\
& \frac{d v(t)}{d t}=-\frac{\gamma}{m} v(t)+\frac{1}{m} \xi(t)
\end{align*}
$$

This is the Langevin equations of motion for the Brownian particle.

The random force $\xi(t)$ is a stochastic variable giving the effect of background noise due to the fluid on the Brownian particle. The effect of the fluctuating force can be summarized by giving its first and second moments

$$
\begin{equation*}
\langle\xi(t)\rangle_{\xi}=0 \quad\left\langle\xi\left(t_{1}\right) \xi\left(t_{2}\right)\right\rangle_{\xi}=g \delta\left(t_{1}-t_{2}\right) \tag{A.3}
\end{equation*}
$$

The average $\langle\ldots\rangle_{\xi}$ is an average with respect to the distribution of the realizations of the stochastic variable $\boldsymbol{\xi}(t), g$ is a measure of the strength of the fluctuation force and the delta function in time indicates that there is no correlation between impacts in any distinct time intervals $d t_{1}$ and $d t_{2}$ because any memory between forces at different times will be lost due to these frequent collisions.

The remaining mathematical specification of this dynamical model is that the fluctuating force has a Gaussian distribution determined by the moments in A.3. We can obtain an explicit formal solution of A. 2 as

$$
\begin{equation*}
v(t)=e^{-t / \tau_{B}} v(0)+\frac{1}{m} \int_{0}^{t} d s e^{-(t-s) / \tau_{B}} \xi(s) \tag{A.4}
\end{equation*}
$$

How do we know that the integral in A. 4 exists?
To obtain a meaning to A. 2 and A. 4 we write A. 2 as

$$
\begin{equation*}
d v(t)=-\frac{\gamma}{m} v(t) d t+\frac{1}{m} d U(t) \tag{A.5}
\end{equation*}
$$

where

$$
d U(t)=\xi(t) d t
$$

We now discuss the integral noice $U(t)$ for long times $t$. Dividing $t$ into intervals we have

$$
\begin{equation*}
U(t)-U(0)=\sum_{k=1}^{n}\left[U\left(t_{k}\right)-U\left(t_{k-1}\right)\right] \tag{A.6}
\end{equation*}
$$

with $0=t_{0}<t_{1}<\cdots<t_{n}=t . U(t)$ is a continous Markov process with zero mean. Applying the central limit theorem to A. 6 we deduce that $U(t)$ is Gaussian with zero mean. Therefore, it has all the requirements for a Wiener process, i.e.

$$
U(t)=W(t)
$$

and we can write A. 5 as

$$
\begin{equation*}
d v(t)=-\frac{\gamma}{m} v(t) d t+\frac{1}{m} d W(t) \tag{A.7}
\end{equation*}
$$

and the solution A .4 becomes

$$
\begin{equation*}
v(t)=e^{-t / \tau_{B}} v(0)+\frac{1}{m} \int_{0}^{t} d s e^{-(t-s) / \tau_{B}} d W(s) \tag{A.8}
\end{equation*}
$$

## A.3.1 Fluctuation dissipation theorem

Now consider the correlation of the Ornstein-Uhlenbeck process with respect to the velocity expressed as follows

$$
\begin{equation*}
C_{v}\left(t_{2}, t_{1}\right)=\frac{g \tau_{B}}{2 m^{2}}\left[e^{-\left(\left|t_{2}-t_{1}\right|\right) / \tau_{B}}-e^{-\left(t_{2}+t_{1}\right) / \tau_{B}}\right] \tag{A.9}
\end{equation*}
$$

where $g$ denotes the variance of $W(t)$.
We can therefore obtain the distribution functions for the velocity knowing the first and second moments:

- First moment

$$
\langle v(t)\rangle=v_{0} e^{-t / \tau_{B}}
$$

- Second moment

$$
\left\langle v\left(t_{2}\right) v\left(t_{1}\right)\right\rangle_{\xi}=\left[v_{0}^{2}-\frac{g \tau_{B}}{2 m^{m}}\right] e^{-\left(t_{2}+t_{1}\right) / \tau_{B}}+\frac{g \tau_{B}}{2 m^{m}} e^{-\left(\left|t_{2}-t_{1}\right|\right) / \tau_{B}}
$$

Here we can also average over the velocity distribution where in equilibrium

$$
\begin{align*}
& \left\langle v_{0}^{2}\right\rangle_{e q}=\frac{k_{B} T}{m} \\
& \left\langle\left\langle v^{2}\right\rangle_{\xi}\right\rangle_{e q}=\left[\left\langle v_{0}^{2}\right\rangle_{e q}-\frac{g \tau_{B}}{2 m^{2}}\right] e^{-2 t / \tau_{B}}+\frac{g \tau_{B}}{2 m^{2}} \tag{A.10}
\end{align*}
$$

The condition for equilibrium is that $\left\langle\left\langle v^{2}\right\rangle_{\xi}\right\rangle_{e q}=k_{B} T / m$. This requires

$$
\begin{equation*}
g=\frac{2 m k_{B} T}{\tau_{B}}=2 \gamma k_{B} T \tag{A.11}
\end{equation*}
$$

This important result is known as the Fluctuation dissipation theorem. It relates the strength $g$ of the random noise or fluctuating force to the magnitude $\gamma$ of the friction or dissipation.

The variance of $v(t)$ is obtained from A. 9 for $t_{2}=t_{1}=t$ and with $g=2 \gamma k_{B} T$ as

$$
\begin{equation*}
\sigma_{v}^{2}(t)=\frac{k_{B} T}{m}\left[1-e^{-2 t / \tau_{B}}\right] \tag{A.12}
\end{equation*}
$$

We can also get an expression for the displacement of the particle

$$
\begin{equation*}
x(t)=x_{0}+v_{0} \tau_{B}\left[1-e^{-t / \tau_{B}}\right]+\frac{\tau_{B}}{m} \int_{0}^{t}\left[1-e^{-(t-u) / \tau_{B}}\right] d W(u) \tag{A.13}
\end{equation*}
$$

the average displacement is then

$$
\begin{equation*}
\mu_{x}(t)=\langle x(t)\rangle_{\xi}=x_{0}+v_{0} \tau_{B}\left[1-e^{-t / \tau_{B}}\right] \tag{A.14}
\end{equation*}
$$

An important quantity is the mean squared displacement of the particle from the starting point, in equilibrium we get for long times

$$
\left\langle\left\langle\left(x(t)-x_{0}\right)^{2}\right\rangle\right\rangle=\frac{2 k_{B} T}{\gamma} t
$$

that can be compared with the diffusion result

$$
\left\langle\left\langle\left(x(t)-x_{0}\right)^{2}\right\rangle\right\rangle=2 D t
$$

which gives the Stokes-Einstein result

$$
\begin{equation*}
D=\frac{k_{B} T}{\gamma} \tag{A.15}
\end{equation*}
$$

## A.3.2 Simulation

In this section we will use what was previously described in order to simulate a Langevin Process. In particular, the simulation will be divided into several steps:

- calculate and show the trajectory of a Langiv process $X(t)$ with velocity $v(t)$ given by the Ornstein-Uhlenbeck process;
- construct a histogram showing the distribution trend for a fixed time $t$ at the arrival point;
- show that the variance of the position has a constant asymptotic trend;
- compare the variance of the position with the variance of the velocity.


## Trajectory of a Langevin process

For simulation was built a function that, from a known starting point and a known sequence of sample times, implements the Langevin process using the Ornstein-Uhlenbeck process as velocity.

```
%The function 'lang' creates and displays a Langevin
    process.
2
%INPUT: X0=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% tau=correlation time.
%OUTPUT: X=Langevin process;
% v=velocity.
function [X,v] = lang(X0,n,T,N,mu, sigma,D,tau)
dt=T/N; %Discretization
t=(0:dt:T); %Vector of sample times
%Construction of the Process:
[v,~] = oup(x0,n,T,N,mu,sigma,D,tau); %recall the
    velocity from Ornstein-Uhlenbeck
X=zeros(n,N); %initialization of the position matrix
for i=1:n
    for j=1:N
```

```
    X(i,j+1)= X(i,j)+v(i,j)*dt;
        end
end
%Plot the output paths:
plot(t,X);
xlabel('Time'), ylabel('Process State');
title('Application of the Langevin equation');
end
```



Figure A.36: Trajectories for $X 0=$ randn, $n=5, T=50, N=100, m u=0, \operatorname{sigma}=1, D=1, \tau=1$.


Figure A.37: Trajectories for $X 0=$ randn, $n=10000, T=50, N=100, m u=0, \operatorname{sigma}=1, D=1, \tau=1$.

## Distribution Trend

In this section we will study the distribution of the process considered.
In particular we use a function that constructs an histogram for a fixed instant of time $t$ at the arrival point $x$.

```
%The function 'dis_hist' explores the distribution of
        points and verifies that it behaves like a Gaussian
%INPUT: X0=starting point;
%. n=number of particles;
%. T=period of observation;
%. N=discretization step;
%. mu=mean;
%. sigma=standard deviation;
%. D=diffusion coefficient;
%. h=number of histogram columns.
%OUTPUT: x=arrival points vector.
function [x] = dis_hist(X0, n,T,N,mu, sigma,D,tau,h)
15
16 figure(1);
[X,~}]=lang(X0,n,T,N,mu, sigma,D,tau )
%[counts, centers]=hist (x,h)
%counts=counts of the number of elements in each
        column, returned as a row vector.
%centers=columns centers, returned as a numeric row
        vector.
[m,x] = hist(X(:,end),h); %Empirical histogram
    function
```

```
m_norm = (m./length (X(:, end)))./(x(2) x (1));
            %Normalize histogram function
bar(x,m_norm); %The Matlab function bar creates
    an
                    %histogram plot
hold ('on') ;
%Plot the shape of empirical distribution on the
        histogram:
plot(x,m_norm ,'r');
xlabel('x'), ylabel('Distribution') ;
title('Empirical Distribution');
figure(2);
pd = fitdist(X(:, end),'Normal'); %The Matlab function
    'fitdist', creates a probability distribution object
        by fitting the distribution specified to the data
        in column vector.
y = pdf(pd, x); %The Matlab function 'pdf', returns the
        probability distribution function of the pd,
    evaluated at the values in x.
%Plot the shape of the empirical distribution on the
%Gaussian one :
semilogy( }\textrm{x},\textrm{y},\mp@subsup{,}{}{\prime}\mp@subsup{b}{}{\prime})
hold ('on') ;
plot(x,m_norm,'r');
title('Comparison with Gaussian');
end
```



Figure A.38: Histogram with shape of the empirical distribution for $X 0=$ randn, $n=10000, T=50, N=100$, $m u=0$, sigma $=1, D=1, \tau=1, \mathrm{~h}=100$.


Figure A.39: Distribution obtained from the arrival histogram for $X 0=$ randn, $n=10000, T=50, N=100$, $m u=0$, sigma $=1, D=1, \tau=1, \mathrm{~h}=100$.

Varying the number of particles $n$ and the time $T$, we show how the empirical distribution becomes similar to the Gaussian.

For this purpose we use the following function which creates a single plot in which both the histogram, the empirical distribution and the Gaussian one appear at each execution.

```
%The function 'tot_hist' creates a single plot in %
    which both the histogram, the empirical
    distribution and the Gaussian one appear.
2
%INPUT: WO=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% h=number of histogram columns.
%OUTPUT: pd=distribution.
function [pd] = tot_hist(W0, n,T,N,mu,sigma,D,h)
[X,~]= lang(X0, n,T,N,mu,sigma,D,tau);
%[counts,centers]=hist(x,h)
%counts=counts of the number of elements in each
    column, returned as a row vector.
19 %centers=columns centers, returned as a numeric row
    vector.
[m,x] = hist (W(:,end),h); %Empirical histogram
```

function

```
m_norm = (m ./ length(W(:,end))) ./(x(2)-x(1)); %
```

Normalize histogram function
bar(x,m_norm); \%The Matlab function 'bar, creates an \%histogram plot
hold (' on') ;
\%Plot the shape of empirical distribution on the histogram:
plot(x,m_norm, 'r', 'Linewidth', 1);
hold ('on') ;
pd = fitdist(W(:, end), 'Normal'); \%The Matlab function
'fitdist, creates a probability distribution object by fitting the distribution specified to the data in column vector.
$y=p d f(p d, x) ;$ The Matlab function 'pdf' returns the probability distribution function of the pd, evaluated at the values in $x$.


Figure A.40: STEP 1:X0 $=$ randn, $n=1000, T=50, N=100, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.41: STEP 2: $X 0=$ randn, $n=1500, T=150, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.42: STEP 3: $X 0=$ randn, $n=3000, T=300, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.43: STEP 4:X0 $=$ randn $, n=6000, T=600, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.44: STEP 5:X0 $=$ randn, $n=10000, T=1000, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$


Figure A.45: STEP 6:X0 $=$ randn, $n=100000, T=1000, N=500, m u=0$, sigma $=1, D=1, \tau=1, h=100$

## Variance Trend

In this section we intend to study the scale law of variance, for this purpose we calculate the variance and build its graph to verify the trend. In particular we use the function 'var_trend' that calculates the variance numerically using the Matlab function 'var'.

We remember that we want to show that the variance of the process is such that: for t that tends to zero it behaves like $t^{2}$ while its asymptotic trend is linear like $t$.

```
% The function 'var_trend, calculates the variance
    using the Matlab function 'var'. Then the function
    makes the plot of the empirical and analytical
        variance trend.
% INPUT: X0=starting point;
% n=number of particles;
% T=period of observation;
% N=discretization step;
% mu=mean;
% sigma=standard deviation;
% D=diffusion coefficient;
% tau=correlation time.
% OUTPUT: V_p=position variance;
% V_v=velocity variance.
function[V_p,V_v] = var_trend(X0,n,T,N,mu,sigma,D,tau)
dt=T/N; %Discretization
t=(0:dt:T); %Vector of sample times
[X,v] = lang(X0,n,T,N,mu,sigma,D,tau)
```

20

```
%Plot of the empirical variance scale law:
```

$V_{-} \mathrm{p}=\mathrm{var}(\mathrm{X}, 1)$;
$\mathrm{V}_{-} \mathrm{v}=\mathrm{var}(\mathrm{v}, 1)$;
plot(t, V_p) \%position
xlabel ('Time ( t )'), ylabel ('Variance (V)');
title ('Variance Scale Law');
hold ' on'
plot(t, V_v) \%velocity
\%Plot of the analytical variance scale law:
hold ' on'
V_num=D. $* 2 . *\left(\operatorname{tau}^{\wedge} 2\right) . * \mathrm{t}$;
plot(t, V_num)
end


Figure A.46: Variance trend for $X 0=$ randn, $n=1000, T=10, N=100, m u=0$, sigma $=1, D=1$, $\tau=0.2,0.3,0.5,0.8,1$.

After having seen the trend of the empirical variance as $\tau$ varies, we consider the graph in A. 47 that compares the latter with the analytical one

$$
\left\langle X^{2}\right\rangle=2 v t
$$

where the coefficient $v$ is

$$
v=D \tau^{2}
$$

The last figure A.48, on the other hand, compares the variance of the position and the variance of the velocity.


Figure A.47: Variance trend for $X 0=r a n d n, n=8000, T=50, N=100, m u=0$, sigma $=1, D=1, \tau=1$.


Figure A.48: Variance trend for $X 0=r a n d n, n=1000, T=10, N=100, m u=0$, sigma $=1, D=1, \tau=1$.

## Appendix B

## Algorithms

In this appendix we report all the lines of code we wrote to carry out the analysis we discuss in the Chapter 3.

The codes are written for a general case in fact we write $i$ in place of the particular matrix considered. All the analysis reported here has been done in the same way for the different cases: slope $10^{\circ}$, slope $15^{\circ}$, slope $20^{\circ}$, slope $30^{\circ}$, slope $40^{\circ}$, slope $45^{\circ}$, considering the corresponding data.

## B. 1 Data cleansing

```
import numpy as np
import pandas as pd
dati = 'dati_slope.csv', # data set 1000 threads
CSVData = open(dati)
X = np.loadtxt(CSVData, delimiter=",")
X0=X[1:1001,0] # t = 0 min
X1=X[1002:2002,0] # t = 10 min
X2=X[2003:3003,0] # t = 20 min
X3=X[3004:4004,0] # t = 30 min
X4=X[4005:5005,0] # t = 40 min
X5 =X[5006:6006,0] # t = 50 min
X6=X[6007:7007,0] # t = 60 min
```

```
X7=X[7008:8008,0] # t = 70 min
X8=x[8009:9009,0] # t = 80 min
X9=x[9010:10010,0] # t = 90 min
X10=X[10011:11011,0] # t = 100 min
def no_repertir(X_curr, X_prev):
    dif_x = X_curr-X_prev
    X_curr = X_curr[dif_x > np.exp(-9)]
    return X_curr
X1 = no_repertir(X1, X0)
X2 = no_repertir(X2, X1)
X3 = no_repertir(X3, X2)
X4 = no_repertir(X4, X3)
X5 = no_repertir(X5, X4)
X6 = no_repertir(X6, X5)
X7 = no_repertir(X7, X6)
X8 = no_repertir(X8, X7)
X9 = no_repertir(X9, X8)
X10 = no_repertir(X10, X9)
```


## B. 2 Histogram representation

```
import matplotlib.pyplot as plt
import seaborn as sns
Ni,Bi,_=plt.hist(Xi, bins = 30, density=True, edgecolor='black',
    label='t=ix10 min')
plt.legend()
plt.show()
# empirical distribution
sns.distplot(Xi, bins = 30, color="blue",hist_kws=dict( density=True,
    edgecolor="black", linewidth=1, label='t=ix10 min'))
plt.legend()
plt.show()
sns.distplot(Xi, bins = 30, color="green",hist_kws=dict( density=True,
    edgecolor="black", linewidth=1,label='t=ix10 min'))
plt.legend()
plt.semilogy()
plt.show()
```

```
Ci = 0.5*(Bi[1:]+Bi[:-1])
plt.plot(Ci,Ni) ## using bin_centers
plt.show()
```

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20, 15
plt.hist(X0, bins = 5, density=True,edgecolor='black', label='t=0m')
plt.hist(X1, bins = 5, density=True,edgecolor='black', label='t=10m')
plt.hist(X2, bins = 20, density=True, edgecolor='black', label='t=20m')
plt.hist(X3, bins = 30, density=True, edgecolor='black', label='t=30m')
plt.hist(X4, bins = 50, density=True, edgecolor='black', label='t=40m')
plt.hist(X5, bins = 80, density=True, edgecolor='black', label='t=50m')
plt.hist(X6, bins = 80, density=True, edgecolor='black', label='t=60m')
plt.hist(X7, bins = 80, density=True, edgecolor='black', label='t=70m')
plt.hist(X8, bins = 80, density=True, edgecolor='black', label='t=80m')
plt.hist(X9, bins = 90, density=True, edgecolor='black', label='t=90m')
plt.hist(X10, bins = 100, density=True, edgecolor='black', label='t=100m')
plt.legend()
plt.show()
```

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20, 15
kwargs = dict(hist_kws={'alpha':.6}, kde_kws={'linewidth':2})
sns.distplot(X1, label='pdf1 - t = 10 min', **kwargs)
sns.distplot(X2, label='pdf2 - t = 20 min', **kwargs)
sns.distplot(X3, label='pdf3 - t = 30 min', **kwargs)
sns.distplot(X4, label='pdf4 - t = 40 min', **kwargs)
sns.distplot(X5, label='pdf5 - t = 50 min', **kwargs)
sns.distplot(X6, label='pdf6 - t = 60 min', **kwargs)
sns.distplot(X7, label='pdf7 - t = 70 min', **kwargs)
sns.distplot(X8, label='pdf8 - t = 80 min', **kwargs)
sns.distplot(X9, label='pdf9 - t = 90 min', **kwargs)
sns.distplot(X10, label='pdf10 - t = 100 min', **kwargs)
plt.legend()
plt.show()
```


## B. 3 Other variables trend

```
area_mean = 'areamean_slope.csv'
CSVData = open(area_mean)
A = np.loadtxt(CSVData, delimiter=",")
y=A [:,0]
x=A[:,1]
fig1 = plt.figure()
plt.plot(x,y)
fig1.suptitle('mean area')
plt.xlabel('time')
plt.ylabel('area')
plt.show()
```

from statistics import variance
\# $\operatorname{Var}(X)=E\left[(X-m u)^{\wedge} 2\right]$
$\mathrm{V} 0=\mathrm{variance}(\mathrm{XO})$
$\mathrm{V} 1=\mathrm{variance}(\mathrm{X} 1)$
$\mathrm{V} 2=\mathrm{variance}(\mathrm{X} 2)$
$\mathrm{V} 3=\mathrm{variance}(\mathrm{X} 3)$
V4=variance (X4)
$\mathrm{V} 5=\mathrm{variance}(\mathrm{X} 5)$
V6=variance (X6)
$\mathrm{V} 7=\mathrm{variance}(\mathrm{X} 7)$
$\mathrm{V} 8=\mathrm{variance}(\mathrm{X} 8)$
$\mathrm{V} 9=\mathrm{variance}(\mathrm{X} 9)$
V10=variance (X10)
$\mathrm{V}=[\mathrm{V} 0, \mathrm{~V} 1, \mathrm{~V} 2, \mathrm{~V} 3, \mathrm{~V} 4, \mathrm{~V} 5, \mathrm{~V} 6, \mathrm{~V} 7, \mathrm{~V}$, $\mathrm{V} 9, \mathrm{~V} 10]$
$\mathrm{V}=\mathrm{np} . \operatorname{array}(\mathrm{V})$
print(V)
$t=A[:, 1]$
fig2 = plt.figure()
plt.plot (t, V)
fig2.suptitle('variance trend of $A^{\prime}$ )
plt.xlabel ('time')
plt.ylabel('variance')
plt.show()

```
from scipy.stats import norm
norm_dist_Xi = norm(Xi.mean(), Xi.std())
xi = np.linspace(np.min(Xi), np.max(Xi), 1000)
pdf_Xi = [norm_dist_Xi.pdf(x) for x in xi]
pdf_Xi = np.array(pdf_Xi)
dxi=pdf_Xi[999]
maxi=np.max(pdf_Xi)
while True:
    for x in Xi:
        pdf_maxi = np.array([norm_dist_Xi.pdf(x)])
        maxxi=x
    if not abs(maxi-pdf_maxi)<np.exp(-7):
        break
```



```
    maxx10])
fig3 = plt.figure()
plt.plot(t,maxx)
fig3.suptitle('maxtrend')
plt.xlabel('time')
plt.ylabel('max')
print(X0 [1],maxx 2, maxx 3, maxx 4,maxx5)
print(maxx6,maxx 7, maxx8,maxx9,maxx10)
plt.show()
```

```
from scipy.stats import pearsonr
#function that calculates the Pearson correlation coefficient:
#output: r correlation coefficient;
# two-sided p-value.
# s fixed to the second time step
corr11=np.array(pearsonr(X[1002:2002,0],X[1002:2002,0]))
corr12=np.array (pearsonr(X[1002:2002,0],X[2003:3003,0]))
corr13=np.array (pearsonr(X[1002:2002,0], X [3004:4004,0]))
corr14=np.array (pearsonr(X[1002:2002,0],X [4005:5005,0]))
corr15=np.array(pearsonr(x[1002:2002,0],X [5006:6006,0]))
corr16=np.array(pearsonr(X[1002:2002,0],X[6007:7007,0]))
corr17=np.array (pearsonr(X[1002:2002,0],X[7008:8008,0]))
corr18=np.array (pearsonr(X[1002:2002,0],X [8009:9009,0]))
corr19=np.array (pearsonr(X[1002:2002,0],X[9010:10010,0]))
corr110=np.array(pearsonr(X[1002:2002,0],X[10011:11011,0]))
```

```
corr1=np.array([corr11[0], corr12 [0], corr13 [0], corr14 [0], corr15 [0], corr16 [0],
    corr17[0], corr18[0], corr19[0], corr110[0]])
print(corr1)
t=A[:,1]
fig6 = plt.figure()
plt.plot(t[1:11],corr1)
fig6.suptitle('correlation trend')
plt.xlabel('time')
plt.ylabel('correlation')
plt.show()
```


## B. 4 Rescale data

```
totY=np.concatenate((Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10), axis=None)
#data scaling
import numpy as np
def NormalizeData(data):
    return (data - np.min(data)) / (np.max(data) - np.min(data))
scaled_totS = NormalizeData(totY)
```


## B. 5 Parameters calculation and data fitting

## B.5.1 Beta distribution

```
from sklearn.datasets import load_diabetes
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import pandas as pd
from distfit import distfit
from scipy.stats import beta
# loc is short for "location parameter", and scale is naturally any scale
    parameters.
# Location parameters would include the mean in the normal distribution and
    the median in the Cauchy distribution.
```

```
# Scale parameters are like the standard deviation in the normal
    distribution, or either parameter of the gamma distribution.
a_y, b_y, loc_y, scale_y = beta.fit(scaled_totY)
print(a_y, b_y, loc_y, scale_y)
ax = plt.subplot(111)
ax.plot(np.linspace(0, 1, 100), beta.pdf(np.linspace(0, 1, 100), a_y, b_y,
    loc_y, scale_y))
plt.show()
a_y10, b_y10, loc_y10, scale_y10 = beta.fit(scaled_totY10)
print(a_y10, b_y10, loc_y10, scale_y10)
ax = plt.subplot(111)
ax.plot(np.linspace(0, 1, 100), beta.pdf(np.linspace(0, 1, 100), a_y10,
    b_y10, loc_y10, scale_y10))
plt.show()
l_y=[ min(CY2), min(CY3), min(CY4), min(CY5), min(CY6), min(CY7), min(CY8),
    min(CY9), min(CY10)]
g_y=[ max(CY2), max(CY3), max(CY4), max(CY5), max(CY6), max(CY7), max(CY8),
    max(CY9), max(CY10)]
mm_s=min(l_y)
MM_s=max(g_y)
fig, ax = plt.subplots(1, 1)
x=np.linspace(0,1,100)
y_y= mm_y + x* (MM_y-mm_y)
z_y=(y_y-mm_y)/(MM_y-mm_y)
ax.plot(y_y, beta.pdf(z_y, a_y, b_y, loc_y, scale_y)*[1/(MM_y-mm_y)], 'r-',
    lw=5, alpha=0.6, label='beta pdf')
```

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 15, 10
#linear
plt.plot(CY2,NY2[2:],label='pdf2 - t = 20 min')
plt.plot(CY3,NY3,label='pdf3 - t = 30 min')
plt.plot(CY4,NY4,label='pdf4 - t = 40 min')
plt.plot(CY5,NY5,label='pdf5 - t = 50 min')
plt.plot(CY6,NY6,label='pdf6 - t = 60 min')
plt.plot(CY7,NY7,label='pdf7 - t = 70 min')
plt.plot(CY8,NY8,label='pdf8 - t = 80 min')
plt.plot(CY9,NY9,label='pdf9 - t = 90 min')
plt.plot(CY10,NY10,label='pdf10 - t = 100 min')
plt.plot(y_y, (beta.pdf(z_y, a_y, b_y, loc_y, scale_y)*[1/(MM_y-mm_y)]/1.1+
```

```
    beta.pdf(z_y10, a_y10, b_y10, loc_y10, scale_y10)*[1/(MM_y10-mm_y10)]/
    10), 'r-', lw=5, alpha=0.6, label='beta pdf')
plt.title('Distribution (A-Amedia)/sqrt[var(A-Amedia)]')
plt.legend()
plt.show()
#logarithmic
plt.plot(CY2,NY2[2:],label='pdf2 - t = 20 min')
plt.plot(CY3,NY3,label='pdf3 - t = 30 min')
plt.plot(CY4,NY4,label='pdf4 - t = 40 min')
plt.plot(CY5,NY5,label='pdf5 - t = 50 min')
plt.plot(CY6,NY6,label='pdf6 - t = 60 min')
plt.plot(CY7,NY7,label='pdf7 - t = 70 min')
plt.plot(CY8,NY8,label='pdf8 - t = 80 min')
plt.plot(CY9,NY9,label='pdf9 - t = 90 min')
plt.plot(CY10,NY10,label='pdf10 - t = 100 min')
plt.plot(y_y, (beta.pdf(z_y, a_y, b_y, loc_y, scale_y)*[1/(MM_y10-mm_y10)]/
    1.3+beta.pdf(z_y10, a_y10, b_y10, loc_y10, scale_y10)*[1/(MM_y10-mm_y10)
    ]/10), 'r-', lw=5, alpha=0.6, label='beta pdf')
plt.semilogy()
plt.title('Distribution (A-Amedia)/sqrt[var(A-Amedia)]')
plt.ylim(np.exp(-4),3)
plt.xlim(-3,3)
plt.legend()
plt.show()
```


## B.5.2 Generalized logistic distribution

```
from sklearn.datasets import load_diabetes
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import pandas as pd
from distfit import distfit
from scipy.stats import genlogistic
a_y, loc_y, scale_y = genlogistic.fit(scaled_toty)
a_y10, loc_y10, scale_y10 = genlogistic.fit(scaled_totY10)
ax = plt.subplot(111)
ax.plot(np.linspace(0, 1, 100), genlogistic.pdf(np.linspace(0, 1, 100), a_y,
    loc_y, scale_y))
plt.show()
```

```
fig, ax = plt.subplots(1, 1)
x=np.linspace (0,1,100)
y_y= mm_y + x* (MM_y-mm_y)
z_y=(y_y-mm_y)/(MM_y-mm_y)
y_y10= mm_y10 + x*(MM_y10-mm_y10)
z_y10=(y_y10-mm_y10)/(MM_y10-mm_y 10)
ax.plot(y_y, genlogistic.pdf(z_y, a_y, loc_y, scale_y)*[1/(MM_y-mm_y)], 'r-'
    , lw=5, alpha=0.6, label='genlogistic pdf')
plt.show()
ax.plot(y_Y10, genlogistic.pdf(z_y, a_y, loc_y, scale_y)*[1/(MM_y-mm_y)], 'r
    -', lw=5, alpha=0.6, label='genlogistic pdf')
plt.show()
```

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 15, 10
#linear
plt.plot(CY2,NY2[2:],label='pdf2 - t = 20 min')
plt.plot(CY3,NY3,label='pdf3 - t = 30 min')
plt.plot(CY4,NY4,label='pdf4 - t = 40 min')
plt.plot(CY5,NY5,label='pdf5 - t = 50 min')
plt.plot(CY6,NY6,label='pdf6 - t = 60 min')
plt.plot(CY7,NY7,label='pdf7 - t = 70 min')
plt.plot(CY8,NY8,label='pdf8 - t = 80 min')
plt.plot(CY9,NY9,label='pdf9 - t = 90 min')
plt.plot(CY10,NY10,label='pdf10 - t = 100 min')
plt.plot(y_y, (genlogistic.pdf(z_y, a_y, loc_y, scale_y)*[1/(MM_y-mm_y)]/
    1.11+genlogistic.pdf(z_y10, a_y10, loc_y10, scale_y10)* [1/(MM_y10-mm_y10
    )]/15), 'r-', lw=5, alpha=0.6, label='genlogistic pdf')
plt.title('Distribution (A-Amedia)/sqrt[var(A-Amedia)]')
plt.legend()
plt.show()
#logarithmic
plt.plot(CY2,NY2[2:],label='pdf2 - t = 20 min')
plt.plot(CY3,NY3,label='pdf3 - t = 30 min')
plt.plot(CY4,NY4,label='pdf4 - t = 40 min')
plt.plot(CY5,NY5,label='pdf5 - t = 50 min')
plt.plot(CY6,NY6,label='pdf6 - t = 60 min')
plt.plot(CY7,NY7,label='pdf7 - t = 70 min')
```

```
plt.plot(CY8,NY8,label='pdf8 - t = 80 min')
plt.plot(CY9,NY9,label='pdf9 - t = 90 min')
plt.plot(CY10,NY10,label='pdf10 - t = 100 min')
plt.plot(y_y, (genlogistic.pdf(z_y, a_y, loc_y, scale_y)*[1/(MM_y-mm_y)]/1.2
    +genlogistic.pdf(z_y10, a_y10, loc_y10, scale_y10)*[1/(MM_y10-mm_y10)]/
    15), 'r-', lw=5, alpha=0.6, label='genlogistic pdf')
plt.semilogy()
plt.title('Distribution (A-Amedia)/sqrt[var(A-Amedia)]')
plt.ylim(np.exp(-4),3)
plt.xlim(-2.8,3)
plt.legend()
plt.show()
```


## B. 6 Analysis of moments of various order

```
scaled_totY2 = NormalizeData(Y2)
scaled_totY3 = NormalizeData(Y3)
scaled_totY4 = NormalizeData(Y4)
scaled_totY5 = NormalizeData(Y5)
scaled_totY6 = NormalizeData(Y6)
scaled_totY7 = NormalizeData(Y7)
scaled_totY8 = NormalizeData(Y8)
scaled_totY9 = NormalizeData(Y9)
scaled_totY10= NormalizeData(Y10)
a_y2, b_y2, loc_y2, scale_y2 = beta.fit(scaled_totY2)
a_y3, b_y3, loc_y3, scale_y3 = beta.fit(scaled_totY3)
a_y4, b_y4, loc_y4, scale_y4 = beta.fit(scaled_totY4)
a_y5, b_y5, loc_y5, scale_y5 = beta.fit(scaled_totY5)
a_y6, b_y6, loc_y6, scale_y6 = beta.fit(scaled_totY6)
a_y7, b_y7, loc_y7, scale_y7 = beta.fit(scaled_totY7)
a_y8, b_y8, loc_y8, scale_y8 = beta.fit(scaled_totY8)
a_y9, b_y9, loc_y9, scale_y9 = beta.fit(scaled_totY9)
a_y10, b_y10, loc_y10, scale_y10 = beta.fit(scaled_totY10)
mY2, vY2, sY2,kY2 = beta.stats(a_y2, b_y2, loc_y2, scale_y2, moments='mvsk')
mY3,vY3,sY3,kY3 = beta.stats(a_y3, b_y3, loc_y3, scale_y3, moments='mvsk')
mY4, vY4, sY4,kY4 = beta.stats(a_y4, b_y4, loc_y4, scale_y4, moments='mvsk')
mY5,vY5, sY5,kY5 = beta.stats(a_y5, b_y5, loc_y5, scale_y5, moments='mvsk')
mY6, vY6, sY6,kY6 = beta.stats(a_y6, b_y6, loc_y6, scale_y6, moments='mvsk')
mY7, vY7, sY7,kY7 = beta.stats(a_y7, b_y7, loc_y7, scale_y7, moments='mvsk')
mY8, vY8,sY8,kY8 = beta.stats(a_y8, b_y8, loc_y8, scale_y8, moments='mvsk')
mY9,vY9,sY9,kY9 = beta.stats(a_y9, b_y9, loc_y9, scale_y9, moments='mvsk')
```

```
mY10,vY10,sY10,kY10 = beta.stats(a_y10, b_y10, loc_y10, scale_y10, moments='
    mvsk')
```

```
MY2=np.mean(scaled_totY2)
MY3=np.mean(scaled_totY3)
MY4=np.mean(scaled_totY4)
MY5=np.mean(scaled_totY5)
MY6=np.mean(scaled_totY6)
MY7=np.mean(scaled_totY7)
MY8=np.mean(scaled_totY8)
MY9=np.mean(scaled_totY9)
MY10=np.mean(scaled_totY10)
mY=[mY2,mY3,mY4,mY5,mY6,mY7,mY8,mY9,mY10]
mY=np.array(mY)
print(mY)
MY=[MY2, MY3, MY4, MY5, MY6, MY7, MY8, MY9, MY10]
MY = np.array(MY)
print(MY)
t=A[:,1]
plt.plot(t[2:],mY, 'r-', label='distribution mean')
plt.plot(t[2:],MY, 'g-', label='data mean')
plt.xlabel('time')
plt.ylabel('mean')
plt.title('Beta Mean')
plt.legend()
plt.show()
```

```
from statistics import variance
VY2=np.var(scaled_totY2)
VY3=np.var(scaled_totY3)
VY4=np.var(scaled_totY4)
VY5=np.var(scaled_totY5)
VY6=np.var(scaled_totY6)
VY7=np.var(scaled_totY7)
VY8=np.var(scaled_totY8)
VY9=np.var(scaled_totY9)
VY10=np.var(scaled_totY10)
vY=[vY2,vY3,vY4,vY5,vY6,vY7,vY8,vY9,vY10]
vY=np.array(vY)
print(vY)
VY=[VY2, VY3, VY4, VY5, VY6, VY7, VY8, VY9, VY10]
```

```
VY = np.array(VY)
print(VY)
t=A[:,1]
plt.plot(t[2:],vY, 'r-', label='distribution variance')
plt.plot(t[2:],VY, 'g-', label='data variance')
plt.xlabel('time')
plt.ylabel('variance')
plt.title('Beta Variance')
plt.legend()
plt.show()
```

```
from scipy.stats import skew
SY2=skew(scaled_totY2)
SY3=skew(scaled_totY3)
SY4=skew(scaled_totY4)
SY5=skew(scaled_totY5)
SY6=skew(scaled_totY6)
SY7=skew(scaled_totY7)
SY8=skew(scaled_totY8)
SY9=skew(scaled_totY9)
SY10=skew(scaled_totY10)
SY=[SY2, SY3, SY4, SY5, SY6, SY7, SY8, SY9, SY10]
SY = np.array(SY)
print(SY)
sY=[sY2,sY3,sY4,sY5,sY6,sY7,sY8,sY9,sY10]
sY=np.array(sY)
print(sY)
t=A[:,1]
plt.plot(t[2:],sY, 'r-', label='distribution skew')
plt.plot(t[2:],SY, 'g-', label='data skew')
plt.xlabel('time')
plt.ylabel('skew')
plt.title('Beta Skewness')
plt.legend()
plt.show()
```

```
from scipy.stats import kurtosis
KY2=kurtosis(scaled_totY2)
KY3=kurtosis(scaled_totY3)
KY4=kurtosis(scaled_totY4)
KY5=kurtosis(scaled_totY5)
```

```
KY6=kurtosis(scaled_totY6)
KY7=kurtosis(scaled_totY7)
KY8=kurtosis(scaled_totY8)
KY9=kurtosis(scaled_totY9)
KY10=kurtosis(scaled_totY10)
KY=[KY2, KY3, KY4, KY5, KY6, KY7, KY8, KY9, KY10]
KY = np.array(KY)
print(KY)
kY=[kY2,kY3,kY4,kY5,kY6,kY7,kY8,kY9,kY10]
kY=np.array (kY)
print(kY)
t=A[:,1]
plt.plot(t[2:],kY, 'r-', label='distribution kurtosis')
plt.plot(t[2:],KY, 'g-', label='data kurtosis')
plt.xlabel('time')
plt.ylabel('kurtosis')
plt.title('Beta Kurtosis Efficiency')
plt.legend()
plt.show()
```


## Appendix C

## Software

In this Appendix, you can find some of the functions from Python library used for the analysis. This function are used in the codes reported in Appendix B. The additional software used is also listed.

## C. 1 Functions

- plt.hist: uses numpy.histogram to bin the data in $x$ and count the number of values in each bin, then draws the distribution either as a BarContainer or Polygon.


## Parameters:

- $x:(n$,$) array or sequence of (n$,$) arrays$

Input values, this takes either a single array or a sequence of arrays which are not required to be of the same length.

- bins: int or sequence

If bins is an integer, it defines the number of equal-width bins in the range. If bins is a sequence, it defines the bin edges, including the left edge of the first bin and the right edge of the last bin; in this case, bins may be unequally spaced. All but the last (righthand-most) bin is half-open.

- range: tuple or None, default: None

The lower and upper range of the bins. Lower and upper outliers are ignored. If not provided, range is $(x \cdot \min (), x \cdot \max ())$. Range has no effect if bins is a sequence. If bins is a sequence or range is specified, autoscaling is based on the specified bin range instead of the range of x.

- density: bool, default: False

If True, draw and return a probability density: each bin will display the bin's raw count divided by the total number of counts and the bin width $($ density $=$ counts $/(\operatorname{sum}($ counts $) * n p . \operatorname{diff}($ bins $)))$, so that the area under the histogram integrates to 1 (np.sum(density $* n p . \operatorname{diff}($ bins $))==$ 1). If stacked is also True, the sum of the histograms is normalized to 1.

- color: color or array-like of colors or None, default: None Color or sequence of colors, one per dataset. Default (None) uses the standard line color sequence.
- label: str or None, default: None

String, or sequence of strings to match multiple datasets. Bar charts yield multiple patches per dataset, but only the first gets the label, so that legend will work as expected.

## Returns:

- n: array or list of arrays

The values of the histogram bins. See density and weights for a description of the possible semantics. If input $x$ is an array, then this is an array of length nbins. If input is a sequence of arrays [data1, data $2, \ldots$ ], then this is a list of arrays with the values of the histograms for each of the arrays in the same order. The dtype of the array n (or of its element arrays) will always be float even if no weighting or normalization is used.

- bins: array

The edges of the bins. Length nbins +1 (nbins left edges and right edge of last bin). Always a single array even when multiple data sets are passed in.

- patches:BarContainer or list of a single Polygon or list such objects Container of individual artists used to create the histogram or list of such containers if there are multiple input datasets.
- sns.distplot: a Distplot or distribution plot, depicts the variation in the data distribution. Seaborn Distplot represents the overall distribution of continuous data variables. The Seaborn module along with the Matplotlib module is used to depict the distplot with different variations in it. The Distplot depicts the data by a histogram and a line in combination to it.


## Parameters:

- data: pandas.DataFrame, numpy.ndarray, mapping, or sequence Input data structure. Either a long-form collection of vectors that can be assigned to named variables or a wide-form dataset that will be internally reshaped.
- label: str or None, default: None

String, or sequence of strings to match multiple datasets. Bar charts yield multiple patches per dataset, but only the first gets the label, so that legend will work as expected.

- kwargs

Other keyword arguments are documented with the relevant axes-level function.

## Returns:

- FaceGrid: an object managing one or more subplots that correspond to conditional data subsets with convenient methods for batch-setting of axes attributes.
- nunpy.mean: compute the arithmetic mean along the specified axis. Returns the average of the array elements. The average is taken over the flattened array by default, otherwise over the specified axis.


## Parameters:

- a: array _like

Array containing numbers whose mean is desired. If a is not an array, a conversion is attempted.

- axis: None or int or tuple of ints, optional

Axis or axes along which the means are computed. The default is to compute the mean of the flattened array.

## Returns:

- m: ndarray.

If out=None, returns a new array containing the mean values, otherwise a reference to the output array is returned.

- variance: returns the variance of the array elements, a measure of the spread of a distribution. The variance is computed for the flattened array by default, otherwise over the specified axis.


## Parameters:

- a: array_like

Array containing numbers whose mean is desired. If a is not an array, a conversion is attempted.

- axis: None or int or tuple of ints, optional

Axis or axes along which the means are computed. The default is to compute the mean of the flattened array.

## Returns:

- m: ndarray, see dtype parameter above.

If out=None, returns a new array containing the mean values, otherwise a reference to the output array is returned.

- correlation: return Pearson product-moment correlation coefficients.The relationship between the correlation coefficient matrix, $R$, and the covariance matrix, $C$, is

$$
R_{i j}=\frac{C_{i j}}{\sqrt{C_{i i} C_{j j}}}
$$

## Parameters:

- x: array_like

A 1-D or 2-D array containing multiple variables and observations. Each row of $x$ represents a variable, and each column a single observation of all those variables.

- y: array_like

An additional set of variables and observations. y has the same shape as x .

## Returns:

- R: ndarray.

The correlation coefficient matrix of the variables.

- beta fit: given the input data, this function gives as outputs the parameters for the distribution to be a beta one. Therefore, it is possible to say that the distribution was calculated empirically.

Parameters: data: array like

## Returns:

- a, b: array like

Shape parameters

- loc: array_like

Location parameter

- scale: array_like

Location parameter

- beta.stats: is a beta continuous random variable that is defined with a standard format and some shape parameters to complete its specification.

Parameters: a, b, loc, scale

Returns: mean, variance, skweness, kurtosis.

- skewness: for normally distributed data, the skewness should be about zero. For unimodal continuous distributions, a skewness value greater than zero means that there is more weight in the right tail of the distribution. The function skewtest can be used to determine if the skewness value is close enough to zero, statistically speaking.


## Parameters:

- a: ndarray

Input array

- axis: int or None, default: 0

If an int, the axis of the input along which to compute the statistic. The statistic of each axis-slice (e.g. row) of the input will appear in a corresponding element of the output. If None, the input will be raveled before computing the statistic.

## Returns:

- skewness: ndarray.

The skewness of values along an axis, returning NaN where all values are equal.

- kurtosis: compute the kurtosis (Fisher or Pearson) of a dataset. Kurtosis is the fourth central moment divided by the square of the variance. If Fisher's definition is used, then 3.0 is subtracted from the result to give 0.0 for a normal distribution. If bias is False then the kurtosis is calculated using k statistics to eliminate bias coming from biased moment estimators.


## Parameters:

- a: ndarray

Data for which the kurtosis is calculated.

- axisint or None, default: 0

If an int, the axis of the input along which to compute the statistic. The statistic of each axis-slice (e.g. row) of the input will appear in a corresponding element of the output. If None, the input will be raveled before computing the statistic.

## Returns:

- kurtosis: array.

The kurtosis of values along an axis, returning NaN where all values are equal.

## C. 2 Hypatia

Hypatia is the Cloud infrastructure that has been developed to support the computational needs of the ELIXIR-GR community, but also the broader community
of life scientists in Greece and abroad. It currently hosts important ELIXIR-GR services and resources (e.g., the national COVID19 Data Portal of Greece), while it undertakes computational tasks in the context of various projects of ELIXIRGR members. The infrastructure is named after Hypatia, a Greek philosopher, astronomer, and mathematician, who lived in Alexandria, Egypt.

Under the hood, Hypatia consists of a powerful computational cluster of heterogeneous physical machines. Currently, its cluster is comprised of: 32 basic nodes: ( 2 CPUs, 14 cores/CPU, 512GB DDR4 RAM), 2 hefty nodes: (2 CPUs, 24 cores/CPU, 1TB DDR4 RAM), 3 GPU nodes: ( 2 CPUs, 14 cores/CPU, 768 GB DDR4 RAM, 2 GPUs), 8 I/O nodes: (2 CPUs, 14 cores/CPU, 512GB DDR4 RAM, $2 \times 2$ TB SSD 6G), 9 infrastructure nodes:(2 CPUs, 14 cores/CPU, 192GB DDR4 RAM).

Hypatia's computational resources are allocated for predetermined time periods to particular user-created projects.
We had the opportunity to use this computational resource because it was provided by BCAM.

## C. 3 Minitab

Minitab is a statistics package developed at the Pennsylvania State University by researchers Barbara F. Ryan, Thomas A. Ryan, Jr., and Brian L. Joiner in conjunction with Triola Statistics Company in 1972. It began as a light version of OMNITAB 80, a statistical analysis program by National Institute of Standards and Technology.

Minitab statistical software helps you understand your current and past data, to find trends and predict patterns, discover hidden relationships between variables, visualise interactions and identify important factors to answer even the most difficult questions and problems.

Minitab has all the tools you need to analyse your data effectively and is able to suggest the right analyses, giving you clear and comprehensive results.

Minitab helps you find meaningful solutions to the most complex business problems. It is used in various sectors of industry, for statistical analysis, cost reduction, increased efficiency, defect reduction and variation control.

Using Minitab, it was possible to verify that the distribution sought was indeed a distribution with a restricted domain. In fact, performing an analysis using Minitab, none of the distributions evaluated by the software were compatible with the data.

## Bibliography

[1] A Alexandridis et al. "Wildland fire spread modelling using cellular automata: evolution in large-scale spatially heterogeneous environments under fire suppression tactics". In: International Journal of Wildland Fire 20.5 (2011), pp. 633-647.
[2] Alex Alexandridis et al. "A cellular automata model for forest fire spread prediction: The case of the wildfire that swept through Spetses Island in 1990". In: Applied Mathematics and Computation 204.1 (2008), pp. 191201.
[3] EA Alhassan, L Albert, and O Louis. "On some Applications of Beta Function in some Statistical Distributions". In: ().
[4] Frédéric Allaire, Jean-Baptiste Filippi, and Vivien Mallet. "Generation and evaluation of an ensemble of wildland fire simulations". In: International journal of wildland fire 29.2 (2020), pp. 160-173.
[5] Frédéric Allaire, Vivien Mallet, and Jean-Baptiste Filippi. "Novel method for a posteriori uncertainty quantification in wildland fire spread simulation". In: Applied Mathematical Modelling 90 (2021), pp. 527-546.
[6] Paolo Baldi. "Stochastic calculus". In: Stochastic Calculus. Springer, 2017, pp. 215-254.
[7] B.C. Carlson. Special Functions of Applied Mathematics. Academic Press, 1977. ISBN: 9780121601508 . URL: https://books.google.it/books? id=WybvAAAAMAAJ.
[8] A Collin, D Bernardin, and O Sero-Guillaume. "A physical-based cellular automaton model for forest-fire propagation". In: Combustion science and technology 183.4 (2011), pp. 347-369.
[9] Vera N Egorova, Andrea Trucchia, and Gianni Pagnini. "Fire-spotting generated fires. Part I: The role of atmospheric stability". In: Applied Mathematical Modelling 84 (2020), pp. 590-609.
[10] Vera N Egorova, Andrea Trucchia, and Gianni Pagnini. "Fire-spotting generated fires. Part II: The role of flame geometry and slope". In: Applied Mathematical Modelling 104 (2022), pp. 1-20.
[11] Tiziano Ghisu et al. "An optimal Cellular Automata algorithm for simulating wildfire spread". In: Environmental Modelling \& Software 71 (2015), pp. 1-14.
[12] Desmond J Higham. "An algorithmic introduction to numerical simulation of stochastic differential equations". In: SIAM review 43.3 (2001), pp. 525546.
[13] Bernardo Lagos-Álvarez et al. "A Type I Generalized Logistic Distribution: Solving Its Estimation Problems with a Bayesian Approach and Numerical Applications Based on Simulated and Engineering Data". In: Symmetry 14.4 (2022), p. 655.
[14] N.N. Lebedev and R.A. Silverman. Special Functions and Their Applications. Dover Books on Mathematics. Dover Publications, 1972. ISBN: 9780486606248. URL: https://books .google.it/books?id=po6Yxz851MC.
[15] Marcos López-De-Castro et al. "Physical and Non-Physical Fire-Spotting Models: A Comparison Study by a Wildfire Simulator Based on a Cellular Automata Approach". In: Environmental Sciences Proceedings 17.1 (2022), p. 27.
[16] Saralees Nadarajah and Samuel Kotz. "Multitude of beta distributions with applications". In: Statistics 41.2 (2007), pp. 153-179.
[17] Grigorios A Pavliotis. Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations. Vol. 60. Springer, 2014.
[18] Holly A Perryman et al. "A cellular automata model to link surface fires to firebrand lift-off and dispersal". In: International journal of wildland fire 22.4 (2012), pp. 428-439.
[19] DVS Sastry and Deepesh Bhati. "A new skew logistic distribution: Properties and applications". In: Brazilian Journal of probability and statistics 30.2 (2016), pp. 248-271.
[20] Ian N Sneddon and E Richard Cohen. "Special functions of mathematical physics and chemistry". In: Physics Today 9.11 (1956), p. 46.
[21] Andrea Trucchia et al. "RandomFront 2.3: a physical parameterisation of fire spotting for operational fire spread models-implementation in WRFSFIRE and response analysis with LSFire+". In: Geoscientific Model Development 12.1 (2019), pp. 69-87.
[22] Andrea Trucchia et al. "PROPAGATOR: an operational cellular-automata based wildfire simulator". In: Fire 3.3 (2020), p. 26.
[23] Domingos Xavier Viegas. "Forest fire propagation". In: Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 356.1748 (1998), pp. 2907-2928.


[^0]:    1 "to perform a betafit" means to fit the data using Python's function beta.pdf( $\mathrm{x}, \mathrm{a}, \mathrm{b}, \mathrm{loc}$, scale) whose inputs were calculated during the analysis from the data itself (B.5.1).
    ${ }^{2}$ similarly "to perform a genlogisticfit" means to fit the data using Python's function genlogistic.pdf( $\mathrm{x}, \mathrm{a}$, loc, scale) whose inputs were calculated during the analysis from the data itself (B.5.2).

[^1]:    ${ }^{3}$ per "eseguire un betafit" si intende fittare i dati utilizzando la function beta.pdf( $x$, a, b, loc, scale) di Python i cui input sono stati calcolati durante l'analisi a partire dai dati stessi (B.5.1).
    ${ }^{4}$ analogamente per "eseguire un genlogisticfit" si intende fittare i dati utilizzando la function genlogistic.pdf(x, a, loc, scale) di Python i cui input sono stati calcolati durante l'analisi a partire dai dati stessi (B.5.2).

