A CONTINUUM EQUATION DISPLAYING TRACY-WIDOM DISTRIBUTION IN THE SPATIAL AND TEMPORAL FLUCTUATIONS OF GROWING INTERFACES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

PHYSICS

By

Xhulian Liçkollari

September 2022

A CONTINUUM EQUATION DISPLAYING TRACY-WIDOM DISTRIBUTION IN THE SPATIAL AND TEMPORAL FLUCTUATIONS OF GROWING INTERFACES

By Xhulian Liçkollari

September 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Fatih Ömer İlday(Advisor)

Oğuz Gülseren

Mehmet Emre Taşgın

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

A CONTINUUM EQUATION DISPLAYING TRACY-WIDOM DISTRIBUTION IN THE SPATIAL AND TEMPORAL FLUCTUATIONS OF GROWING INTERFACES

Xhulian Liçkollari
M.S. in Physics
Advisor: Fatih Ömer İlday
September 2022

A wide variety of surface growth phenomena involves random processes that result in correlated stochastic dynamics. Such dynamics is most succinctly described by a nonlinear equation known as the Kardar-Parisi-Zhang (KPZ) equation. It is of particular interest that the random fluctuations observed along a growing interface described by the KPZ equation turn out to be correlated, with statistics that match the so-called Tracy-Widom distribution. The correlated fluctuations pertain only to the space dimension, namely, along the growing interface. The fluctuations of any given point along the interface over time remain uncorrelated, thus exhibiting Gaussian fluctuations. This is to be expected since the KPZ equation and the experimental systems where the Tracy-Widom statistics have been observed lack mechanisms to induce temporal correlations. Recently, a new mechanism of dissipative self-assembly has been reported, where the self-assembly process is driven by an intrinsic feedback mechanism that is expected to induce temporal correlations. Indeed, such correlations have been experimentally observed with statistics that match the Tracy-Widom probability distribution. Here, we explore the theory of the emergence of correlated temporal fluctuations in such a system when a simplified feedback mechanism is introduced. We develop a highly simplified model, which formally constitutes a modified KPZ equation. We, then, show that this modified equation exhibits temporal fluctuations that are well described by Tracy-Widom fluctuations, up to at least the eight moment, in excellent agreement with the experimental results.

Keywords: Universality, KPZ equation, Tracy-Widom distribution, temporal fluctuations, surface growth phenomena.

ÖZET

BÜYÜYEN ARAYÜZLERİN UZAYŞAL VE ZAMANSAL SALINIMLARINDA TRACY-WIDOM DAĞILIMI SERGİLEYEN BİR SÜREKLİ ORTAM DENKLEMİ

Xhulian Liçkollari
Fizik, Yüksek Lisans
Tez Danışmanı: Fatih Ömer İlday
Eylül 2022

Yüzey büyüme içeren çok çeşitli fenomen korelasyonlar içeren stokastik dinamiklerle sonuçlanan rastgele süreçleri içerir. Bu tür dinamikler en kısa ve öz olarak Kardar-Parisi-Zhang (KPZ) denklemi olarak bilinen doğrusal olmayan bir denklemle betimlenir. KPZ denklemine uygun olarak büyüyen bir arayüz boyunca gözlemlenen rastgele salınımların Tracy-Widom olarak bilinen bir tür dağılım istatistiği sergilemesi özellikle ilgi çekicidir. Bu salınımlardaki korelasyonlar sadece uzay boyutunda, yani büyüyen arayüz boyunca, elde edilmektedir. Arayüz boyunca herhangi bir noktanın zaman içindeki salınımları ise korelasyon içermez, bu nedenle Gauss salınımları sergiler. KPZ denklemi ve Tracy-Widom istatistiklerinin gözlemlendiği deneysel sistemler zamansal korelasyonları indükleyecek herhangi bir mekanizmaya sahip olmadığından bu beklenen bir durumdur. Yakın geçmişte, enerji kayıplı dinamikleri olan yeni bir özbirleşme mekanizması duyurulmuştur. Bu mekanizmada özbirleşim bir takım geribesleme mekanizmaları ile gerçekleşmektedir. Bu geribesleme mekanimalarının zamanda korelasyonları indüklemesi beklenir. Gerçekten de, bu tür zamansal korelasyonlar deneysel olarak gözlemlenmiştir ve Tracy-Widom olasılık dağılımına uyduğu saptanmıştır. Burada, biz basitleştirilmiş bir geri besleme mekanizmasını katarak böyle bir sistemde zamansal salınımlarda korelasyonun ortaya çıkışının teorisini araştırıyoruz. Bizim modelimiz matematiksel yapısı olarak değiştirilmiş bir KPZ denklemine denk gelmektedir. Bu değiştirilmiş KPZ denkleminin, deneysel sonuçlarla mükemmel bir uyum içinde, en az sekiz momente kadar Tracy-Widom salınımlarına sahip olduğunu gösteriyoruz.

Anahtar sözcükler: Evrensellik, KPZ denklemi, Tracy-Widom dağılımı, zamansal salınımlar, yüzey büyüme fenomenleri.

Acknowledgement

First and foremost, I want to express my deepest gratitude to my supervisors, Dr. Serim Ilday and Dr. F. Ömer Ilday for their invaluable advice, continuous support, and patience during my MSc study. I have to thank very much Dr. Ghaith Makey for his invaluable support for my work as well as all other members of the Ilday groups. I owe a very important debt to Batoul Hashemi and Selçuk Parlak for their assistance and great friendship. In addition, I would like to express my heartfelt appreciation to all my dear friends, especially my home mates Ridell Logli and Fjoralb Ismolli. Lastly, I would be remiss in not mentioning my family. Their unconditional support and encouragement is the most precious thing I had.

To my beloved family . . .

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The work in this thesis is, in a way, all about surfaces; however, if it seems so, this research topic is not superficial at all. We may often underestimate the importance of the surfaces, but they are the ones with which we are in contact every moment, and most of our interactions are done through them.

Having all the complex conditions necessary for life only on the surface of a sphere, whose inside is molten and so hostile to life, is just an example of surfaces' importance. Similarly, on a much smaller scale, the functionalities of cell membranes are vital for living organisms.

Thus, studying surfaces is a must, and the next question is, how do we do it while considering the huge variety of them? Surfaces can differ in many aspects, such as their size, composition, structure, shape, roughness, and dynamics. And at the same time, they may be similar to each other or themselves (as in fractals). Moreover, some of their properties are relative to the scale of observing them. For instance, the same surface can be thought smooth or rough depending on the distance from which we are viewing it.

This considerable variety has brought up the necessity of classification and the usage of many other various scientific methods and models to approach this field of research.

Here, we study the kinetic rough surfaces, especially the process of how the interfaces grow with time, which definitely is one of the fascinating branches of the field and with much practical importance.

Examples of rough surface growths are very diverse and in a wide range, too. We would roughly say that they are observed anywhere there is a boundary moving as the consequence of the growth or shrink in size of the system it is surrounding. That boundary can be the rough, upward-growing interface of snow mainly noticed when deposited near the windows during some snowfall, the boundary of a growing bacterial colony or group of cancer cells, the surface of rocks in caves - stalagmites - being formed by deposition, the surface of caves themselves shaped under the process of erosion, the boundary of a liquid propagating in some cloth or similar porous media, the flame fronts propagating in a piece of paper or a dense forest, the boundary of thin films growing by molecular beam epitaxy, and the like.

We can notice many different characteristics, even in the small list of the above examples. For example, the time of the growing process of a bacterial colony may vary between the types of bacteria, but it is way shorter than the time it takes for a stalagmite to form, which is also differing very much depending on the type of their composing materials. Also, the system of water flowing in a piece of paper has substantial size differences from that of the oil flow in vast areas of oil-bearing rocks. However, it seems that all these diverse phenomena -being in some strange way so similar- can be studied under this topic of stochastic surface growths with the help of methods such as the scaling concepts.

Scaling relations, bringing closer together systems that differ so much in space-time scales, and many other aspects, are one of the main bricks in constructing the universality classes, those collections of systems showing similar behaviours independent of the differences in details as the ones aforementioned. They are frequently used as methods or tools to analyze the stochastic growth phenomena, accompanied by, of course, experiments and also other mathematical tools such as the many discrete models and continuum equations [1].

In this thesis, we deal with continuum equations, being very useful, practical and instructive tools for describing surface growths, and, in particular, we work with one of the most essential continuum equations in this area of research, that of KPZ equation [2]. We start by giving some background on this remarkable equation, explaining its usage and where its universality extends. Then, in this way, we have to discuss the universality, how it helps us, and go on with the Tracy-Widom distribution [3], as one of the main universal statistical properties related to the KPZ equation.

In giving this background at the start, we cannot end it without talking about a significant, recent experimental discovery in the specific field, which is the primary motivation of this work. Thus, introducing and roughly explaining the dissipative self-assembly (DSA) experiments [4] and their innovative results ends the second chapter of the thesis.

In the third chapter, we discuss our work regarding the simulation of the KPZ equation itself, producing the theoretical results and analyzing its main components and characteristics. Firstly, the method used is explained, including all details of the simulation, and then the analysis results regarding interface width and spatial and temporal height fluctuations are shown. This chapter, together with the background, is crucial to better understanding the other part of the work since it is built upon what is explained here.

Next, the fourth chapter deals with the most significant contribution of the work, that of modifying the KPZ equation and showing the novel results of this change. After going over how we added two other terms to the KPZ equation, we again explain the method of simulating and analyzing this new equation, making many references and comparisons to the details of the third chapter. Then we reveal what this new equation produces, what is different from the results in the previous form, and how these new authentic results have a real significance in the field.

Finally, we summarise the work's main contributions and what new roads are opened up from the new results shown in this thesis.

Chapter 2

Background and Motivation

2.1 KPZ Universality Class

Since 1986, when Kardar, Parisi and Zhang [2] first published their work, the KPZ equation has been at the centre of the studies around stochastic kinetic surfaces. One reason for this was that KPZ's work became one of the main gates toward a greater understanding of complex systems living in far-from-equilibrium circumstances.

It came as a solution to two of the significant challenges in the field, that of being able to describe the rough kinetic interfaces quantitatively and giving some fundamental means to classify them, which are both very important in the scientific endeavour.

By that time, numerical models illustrating the deposition of particles by only some simple rules could produce results closer to the growth of complex interfaces observed in nature [5, 6]. This was the case in the Ballistic [7] and Eden [8] deposition models, to name but just a few.

Having a single equation that would describe the not-so-straightforward behaviours seen on these models was not easy.

For the case of the totally random deposition model, the most straightforward equation describing the interface growths worked perfectly, but it was not enough to describe more complex observations or models.

This equation describes the evolution of the interface during the deposition process using the function $h(\mathbf{x}, t)$ to show the surface height over the space coordinate x at each instant.

Its particular form was

$$\frac{\partial h}{\partial t} = v + \sqrt{D}\,\xi(\mathbf{x}, t) , \tag{2.1}$$

where the time derivative in the left is calculating the evolution in time of that rough edge's height $h(\mathbf{x}, t)$, as its growth is being driven by randomly "depositing particles" expressed on the right hand of the equation by v as the constant growth velocity (deposition rate), and $\sqrt{D}\,\xi(\mathbf{x}, t)$ as the noise which provides the roughness on the surface and its fluctuations in time.

The random term is usually a white noise in space-time domains, which, as in the case of KPZ, has a zero mean, a finite standard deviation and is uncorrelated:

$$\langle \xi(\mathbf{x}, t) \rangle = 0 \qquad \langle \xi(\mathbf{x}, t)\xi(\mathbf{x}', t') \rangle = \delta^d(\mathbf{x} - \mathbf{x}')\,\delta(t - t') \quad [5].$$

However, as in the random deposition model itself, the lack of some lateral correlation is causing the modelled interfaces to be so rough that they do not seem realistic at all, at least compared to most of the natural phenomena of surface growth.

Introducing some surface relaxation as an additional property to the random deposition model made results closer to the natural phenomena. This meant that equation (2.1) had to be changed so that it described the behaviours of this new model, namely the random deposition with surface relaxation (RDSR). Edwards and Wilkinson carefully observed the differences and thought to include some kind of spatial diffusion in the equation so that interfaces get smoother, thus, more realistic.

The modification to the equation (2.1), coming up at 1982 [9], consisted of adding one more term to the equation and converting it to a version of the widely used Langevin equation:

$$\frac{\partial h}{\partial t} = v + \nu \nabla^2 h + \sqrt{D}\, \xi(\mathbf{x}, t) \; . \tag{2.2}$$

The term consisting of a Laplacian and its coefficient described the diffusion on the surface very well. In the field of surface growth phenomena, equation (2.2) was called Edwards-Wilkinson (E-W) equation since they introduced the new term and also, later on, it was discovered that this would be the default equation for the particular universality class named on their name too.

However, many of the natural surface growth processes or numerical deposition models were still not included in this universality class [5]. Only four years later, when KPZ's work [2] came about, it was shown that most of the interesting surface growth phenomena could not adequately be described by the E-W equation and that a nonlinear version of it would work instead. The E-W linear equation could not account for the more complex behaviours seen in the processes outside of the E-W universality class.

Kardar et al. [2] proposed a new nonlinear term, responsible for lateral growth or similar behaviours present in many different cases, to be added into the E-W equation, making thus, what we call the KPZ equation:

$$\frac{\partial h}{\partial t} = v + \frac{\lambda}{2} (\nabla h)^2 + \nu \nabla^2 h + \sqrt{D}\, \xi(x, t) \; . \tag{2.3}$$

This would increase the variety of surface growth phenomena being able to be described quantitatively via the above stochastic differential equations. Tackling the most complex and interesting cases, the KPZ equation would become a more satisfactory basis for the people working in this field.

Moreover, these three mentioned equations define each of their universality classes, and together with some other equations which also describe interface growth in the hydrodynamic limit, are thus, classifying the growth processes, which helps a lot in digesting easier all the different results from these phenomena since most of them are belonging to one of the universality classes.

Various approaches may be taken to determine the universality class of some model or phenomena, but the most used one is to classify them according to the scaling exponent β [1]. This exponent is also the only independent one for the KPZ dynamics in 1D systems [5].

Nevertheless, what is the exponent β? The width of the rough interface between the growing medium and the outside increases with time; thus, the spatial fluctuations get bigger as time passes. This dependence of roughness' growth with time is a power law. Briefly, it is proportional with time t to power some certain exponent β, and that is where it comes from.

Usually, the $\beta = 1/4$ case signifies the E-W universality class [10]. Whereas the scaling exponent β being equal to 1/3 is one of the special cases and corresponds to many interesting nonlinear growth processes, it is that exponent which was fascinatingly captured by the KPZ's work. In the 1+1 dimensional case, it is analytically proven that it must be the scaling exponent for the KPZ equation [2], which is also supported by a significant number of various simulation results [11, 12].

Now, let us see the KPZ equation and its universality class closely. Since the nonlinear term is causing height growth, we can discard the constant velocity term, which was necessary for the previous equations for growth production; thus, we are left with three terms on the right hand of the equation.

The randomness comes again from the only stochastic term, the $\sqrt{D}\xi(\mathbf{x},t)$, which has not changed and is the same as in the first equation. It is the uncorrelated normally distributed noise with:

$$\langle \xi(\mathbf{x},t) \rangle = 0 \quad \text{and correlation function} \quad \langle \xi(\mathbf{x},t)\xi(\mathbf{x}',t') \rangle = \delta^d(\mathbf{x}-\mathbf{x}')\,\delta(t-t')\ ,$$

and the value of the parameter D in front of it directly varies the noise's standard deviation.

The surface diffusion is in the KPZ equation with the same justification Edwards and Wilkinson had to include in their equation. It is "relaxing" the surface over space, and how much is going to affect the dynamics is decided by the value of its coefficient ν, which, as all other parameters in the equation, is a physical constant varying accordingly to the specific system and is thought to represent the surface tension in many cases.

The nonlinear term, simply the square of the first space derivative of height, changes the dynamics enough so that the KPZ equation defines another universality class. There may be several different origins of this term, depending on the physical system or model, but Kardar et al., in their groundbreaking paper, have justified the form of this nonlinear term mainly on the presence of the lateral growth [2]. Its coefficient $\lambda/2$ also comes from the approximation they do in justifying the term, and λ's value varies the amount of nonlinearity in the equation and depends on the phenomenological properties.

However, more new fascinating properties of the KPZ equation had to come out later, making it one of the essential equations for the theoretical physics of the past decades.

It was early observed that if we make

$$h(x,t) \equiv \frac{2\nu}{\lambda} \ln Z(x,t)\,,$$

the so-called Hopf-Cole transformation, it would map the KPZ equation onto the area of the directed polymers in random media [13], which, by contrast, is part of the traditional equilibrium statistical physics.

Moreover, if we replace ∇h by -let us say- u, we obtain the stochastic Burgers equation [14]. Apart from many other advantages, in this way, the physics of KPZ and directed polymers in random media is related to the well-worked area of driven lattice gases [13], expanding so, much more, the area of KPZ equation's significance.

Furthermore, dealing with this special stochastic partial differential equation (even though mostly in 1+1 dimensions) required work (as reported in many reviews [5, 15, 16]) to be done on each front, the analytical and numerical front, and of course, the experimental one, which was the most subtle one.

Soon, the focus of analysis of results became mainly the scaling exponents (β and many others not mentioned above), correlation functions and universal amplitude ratios, and many probability distributions, with much interest being on the PDF of the height fluctuations.

The analysis of the probability distribution of height fluctuations made it possible to show the inspiring link between the KPZ growth dynamics and an enormous and significant universality class of Gaussian random matrices.

The unexpected impressive discovery came by Johansson [17], who rigorously showed that the height fluctuations of a specific 1+1 dimensional growth model from KPZ universality class were in the Tracy-Widom (T-W) largest eigenvalue distribution for the Gaussian Unitary Ensemble (GUE) of random matrices [3, 18].

Soon after, a similar stunning result came from Prähofer and Spohn [19], who studied another growth model known to be in the KPZ class, particularly the polynuclear growth (PNG). They indicated that PNG could be mapped onto the Ulams's problem about random permutations governed by T-W GUE, too. It only needs to think of height in the growth model as the length of the largest increasing subsequence of random permutations in Ulam's problem, and it can then easily be shown that it behaves like the largest eigenvalue of GUE of random matrices.

Moreover, using these mappings, Prähofer and Spohn were able to show that different T-W distributions come up according to the different initial conditions used on the growth model, and, for example, they specifically showed that T-W GOE (Gaussian Orthogonal Ensemble) is the distribution to be expected on the 1+1 dimensional KPZ interface growths starting from a flat substrate.

Yet, these being only the first observations, T-W continued to pop up in many other related models [20, 21], and this was then becoming one of the universal properties signifying the KPZ universality class.

Then, many independent research teams reported exact solutions to the issue a decade after the first observation. The seemingly very different but closely related 1+1 dimensional systems of the KPZ equation and the directed polymers in random media were analytically and numerically studied. Those teams [22, 23, 24, 25, 26] rigorously deduced the first exact expressions solving each problem, and also obtained the exact formula for the universal probability functions of respectively height and free-energy for any time on their evolution towards respective asymptotic Tracy-Widom forms.

Unfortunately, things are not so clear when it comes to experimental work. Apart from the very low number of good experiments, the difficulty in their interpretation is another problem. This also shows how delicate and complicated the work is in this complex nonlinear world.

It was only up to nearly the same time, in 2010, when some good reliable results came for the first time from experiments in KPZ growing systems showing the T-W probability distributions of height fluctuations. It was the work done by Takeuchi et al. [27] which also experimentally showed that T-W distributions from different types of ensembles (GUE, GOE or GSE) come up for different geometries of the substrate over which the growing is done, same as what was seen on the theoretical models a decade before.

Within this low number of experimental observations of Tracy-Widom statistics in the surface growing systems, some notable results came in 2020 from experiments in dissipative self-assemblies [4]. Apart from many other rich and complex behaviours that those systems revealed, the temporal fluctuations of the growing assembly's surface turned out to have Tracy-Widom distribution.

But what is special about the Tracy-Widom distribution and this remarkable experimental result? Let us discuss more about them in the following sections.

2.2 Tracy-Widom Distribution

Talking about the Tracy-Widom distribution is never complete without mentioning what universality is and why it is important. It is so because the property of universality is what makes Tracy-Widom very special in the first place.

We think of universality when we observe the same specific behaviour in many seemingly unrelated systems. This property's independence from the detailed mechanisms of diverse systems' features and constitutive elements is usually referred to as universality.

Even though there might be many theories and hypotheses about the origin of this phenomenon, it remains an intriguing complete mystery, which is also lying behind most of the success of physics about complex systems. It is fascinating how universality helps us profoundly understand and accurately model complicated systems without requiring much knowledge of the inner details.

Being so crucial, universality has become a focus of attention for many researchers. In the past decades, many universal surface growth phenomena behaviours have been central in studies of stochastic non-equilibrium systems. Over about the same period, a very puzzling phenomenon, usually called Random Matrix universality [28], took great interest, too. A bizarre universal pattern connects many diverse systems from mathematics, physics and biology. It is so hard to believe that the same pattern of the random matrices is emerging in the chicken eyes [29, 30], atom nuclei [31, 32, 33], a special bus system [34], and zeroes of Riemann zeta function [35].

Surprised by this behaviour, many mathematicians and physicists are working on modelling complex correlated systems via random matrices, those simple enough mathematical objects which seem to give rise to a deeper understanding of the world.

And, from work on random matrices, too, Craig Tracy and Harold Widom [3] discovered the peculiar probability distribution named in their honour, which soon was to reveal another universality in the realm of Gaussian random matrices. Tracy-Widom distribution immediately explained a critical value in a simple mathematical model devised from R.May [36] for the populations of different interacting species in a complicated ecological system. Soon, many manifestations of T-W were showing up all over mathematics, physics and beyond.

The Tracy-Widom distribution is the probability distribution of the largest eigenvalue of Gaussian random matrices [3, 18]. It describes the variations (fluctuations) that the largest eigenvalue has on a large ensemble. Tracy and Widom define the three distribution functions F_β with $\beta = 1, 2$, and 4, according to the symmetries of the matrix ensembles [3, 18]. Thus, at first look, they seem like a specific property of the Gaussian random matrices; however, their universality was to become obvious in a short time.

In 1999, it was shown that this distribution describes other variations in another unrelated mathematical model, such as random permutations. There, one only "shuffles" some integers and, in one case, counts the longest increasing subsequence in them, which turns out to be a number fluctuating according to the T-W distribution [37]. Another case can be counting the longest common subsequences, and again we get the same statistical curve [38].

One year later, it appeared in the realm of stochastic growths [17], and in a decade, as mentioned in the previous section, it was even analytically related to the 1+1 dimensional cases of the KPZ equation and directed polymers in random media [22, 23, 24, 25, 26]. T-W statistics are seen in most of the "members" of the KPZ universality class, such as the polynuclear growth (PNG) [17], deposition models involving some nonlinearity as the Ballistic [38], Eden [39, 40] and Restricted solid-on-solid [41] models, the ballistic decomposition model [20], the asymmetric simple exclusion process [17, 42, 43], and also in some of the particular models within directed polymers in random media [21, 26, 44, 24].

The components can differ very much, be it integers, particles, polymers or living species, but their common statistical behaviour remains the same, universal. Moreover, T-W is not limited to mathematics and physics, but it is also seen in other diverse fields as in financial performance [45, 46] or wireless communication [47, 48]. Most systems making up the T-W universality class seem to be within the class of complex correlated systems. Also, the T-W distribution's universality is believed to be quite related to the universality of the phase transitions, and, in a way, it seems to be the analogue of the Gaussian distribution for the complex correlated systems since the Gaussian bell-curve is universal for the systems of independent random variables.

As stated in the previous section, the first experimental appearance of T-W distribution in the random growth systems was in turbulent liquid crystal experiments by Takeuchi et al. [27] in 2010. And lately, it appeared in the DSA experiments [4], where the systems are complex and correlated, as most of the other members of the same universality class, and also they are showing many other universal behaviours within their own.

2.3 Dissipative Self-Assembly Experiments

They were these remarkable experiments which motivated us mostly to work on this topic. The evidence of universal T-W temporal fluctuations within these growing aggregates was very crucial for the primary hypothesis suggested by the supervisor at the beginning of our research.

These experiments and their results are explained below in a bit more detail, and it should be mentioned here that all the information that follows in this section is paraphrased according to the two seminal papers [49, 4] from Ilday et al. published in the *Nature* journal so that to not repeatedly refer to them in each sentence.

The simplest system, which interests us more, consists of tens of thousands of colloidal nanoparticles living in an almost 2D system filled with water and "performing" many complex behaviours similar to the ones seen in living organisms. The physical system is mainly composed of an ultrafast laser beam shining over two glass plates sandwiching the water with "swimming" colloidal particles. Since the distance between the glass plates is comparable to the diameter of the particles, the system is considered a quasi-two-dimensional system; however, the system is wide enough for particles to move freely. The particles' motion is influenced mostly by water flow (which in turn is influenced by the laser beam) and the Brownian motion.

This simple system of particles self-aggregates when the laser is on, and the aggregate manifests complex behaviours such as self-sustaining, self-regulation, self-replication, self-healing and mobility.

Moreover, the authors later show that those main behaviours do not depend on the system's inner details and turn out to be universal. Notably, systems of different "particles" of various sizes suspended in different liquids can perform the dissipative self-assembly methodology. The experimental evidence shows this on $\sim$ 3-nm CdTe quantum dots, 500-nm polystyrene colloids, $\sim$ 0.7-μm soft spheres of non-motile bacterial cells, $\sim$ 1-μm$\times$2-μm rod-like motile bacterial cells, $\sim$ 5-μm elliptical yeast cells and $\sim$ 15-μm human cells, and so we see that it is not only the size changing but also the complexity of these "particles" swimming in some liquid. Showing such universal behaviours independent of such significant differences in the components of the system, and not showing T-W universality would be a surprise.

Furthermore, a particular property as the hyperuniformity was detected in the same system of the colloidal particles [50]. The same property is observed in the bird's eye receptors, whose distribution is also related to the random matrices [29, 30].

The aggregate of the colloidal polystyrene nanoparticles can take many shapes according to the situation, but usually, it grows from a single seed into large circular aggregates as the particles are continuously being "deposited" to it from every direction. These aggregates form a perfect system for studying surface growth phenomena since they, being in a quasi-two-dimensional system, can produce a rough 1D interface growing in time. The T-W universality pops up in this system in the temporal fluctuations of the radius of the aggregate (height of the interface).

The temporal fluctuations have the T-W GUE distribution with odd moments being negative, which makes a ”mirrored” T-W distribution, that is nothing different in the universal statistical properties apart from the sign of these fluctuations. Moreover, the sign of the temporal fluctuations directly depends on the particular choice of defining them.

With all of these in mind, we started our research about finding this Tracy-Widom universality in the temporal fluctuations described by some continuum equation that had to be discovered. Our first hypothesis was that a modification in the time domain of the KPZ equation would work.

Chapter 3

Numerical Analysis of KPZ Equation

To understand the universal (and other) properties of the KPZ dynamics, many people have been headed to numerical simulations as the main method of approaching the problem. The simulations done are various and in an extensive range, including also the simulations of discrete models, such as the deposition growth models, PNG model, ASEP model etc., whose common thing is belonging to the KPZ universality class.

Here, we do only a direct simulation of the KPZ equation in 1+1 dimensions with a flat substrate using the Runge-Kutta (4th order) method.

$$\frac{\partial h(x,t)}{\partial t} = \frac{\lambda}{2}\left(\frac{\partial h(x,t)}{\partial x}\right)^2 + \nu\frac{\partial^2 h(x,t)}{\partial x^2} + \sqrt{D}\xi(x,t) \tag{3.1}$$

We were able to get the $\beta = 1/3$ scaling exponent for the roughness growth of the system, which signifies the KPZ universality class. Moreover, we confirmed once more that the height fluctuations over space are in the T-W distribution and also looked at the distribution of what we define as the temporal fluctuations of height, which in the case of the KPZ equation is Gaussian.

3.1 Method

However, let us first explain this simulation's details and key points.

1. *Integration Elements*

The integration is done over a grid discretized by $dx = 1$ as the space element and $dt = 0.01$ as the time-step.

2. *Initial Conditions*

Taking $h_0 = 0$ (over all x) as the initial condition is a perfect choice for simulating interfaces that grow over a flat substrate.

We began by simulating the equation using the Euler method for the integration since it is the simplest and perfect for a first exploration which would help us to decide on other technical things such as the boundary conditions and parameter ranges.

3. *System Size and Boundary Conditions*

Deciding over the system size and boundary conditions became then easier as we were trying each step we were taking. First of all, the boundary in the space domain had to be periodic because the numerical integration was failing on the boundaries. It produced divergences on the left and right boundaries very early, and they were increasing with time towards the inside of the system.

Secondly, after putting the constriction of having periodic BC, we had to increase the length of the substrate significantly so that it would locally "feel" like a 1-dimensional system in space. Thus, the equation is actually simulated over a grid surrounding a cylinder with a perimeter of base equal to L.

And then, since for a large L, a great amount of time is needed, they were decided to be exactly 2^{13} units for L and 5000 seconds for the total time. Also, we will investigate the KPZ equation's long-time behaviour; thus, having a large total time is significant.

4. *Adjusting the Nonlinear Term*

Yet, the divergences were not over, and this time there was no problem with the technicalities we were using. Even shrinking the time-step interval was not helping. No matter how we changed the parameters, the interface continued to be unstable after a while during the integration process. Some rapid growths were occurring on some spots of the interface; that is, in those local regions, the height growth velocity was much higher than the corresponding velocity of the interface. This would cause divergences within the time of our process or alter the statistical results so that they were way different from the respective theoretical ones. Then, the literature on the simulations of the KPZ equation came to our help, and we found the exceptional papers by Dasgupta et al. [51, 52]. They explain in detail that the instabilities we encountered are intrinsic to the discretized version of the equation. This instability is generic, so it is a property of a large class of discretized nonlinear equations with or without noise. They occur when isolated pillars or grooves of the interface experience rapid growth in height/depth when the magnitude of their height/depth surpasses a critical limit, and this seems to happen because the algebra of derivatives is not always the same in the discretized and continuum versions.

However, in there was well-explained the solution to the problem, too. We could control these instabilities in many methods, such as introducing the other missing higher-order nonlinear terms with their appropriate coefficients, or applying any meaningful restriction to the growth of isolated pillars or grooves of the interface. Dasgupta et al. also proposed replacing the nonlinear term $(|\frac{\partial h}{\partial x}|^2)$ with the sigmoidal type function $f(x) = (1 - e^{-cx})/c$, x being the NL term. This does the trick because it introduces an infinite number of higher-order NL terms with their coefficients depending on the adjustable constant c.

So, we went on with that and soon saw that our simulations' results were not far from the respective results of theoretical works. It was so because this function's job is roughly preventing the instabilities from happening when "unexpected" extreme noise points try to make the discrete form of the nonlinear term diverge. Whereas on the other parts of the interface, the effects of the higher-order NL terms are very low, supporting the arguments of neglecting them while deriving the equation with its famous form.

The results of our numerical integration were now matching what is expected from the continuous KPZ equation, as they should when no instability is present. All the problems regarding divergences were solved, but it cost us adding one more constant to our simulation.

5. *Parameters*

Thus, there was one more parameter whose optimal value was to be decided. We played a lot with all the parameters of the equation, but, in the end, we decided to continue with the most straightforward configuration we were to get fast and stable results.

The Taylor expansion about $x = 0$ of the function used to replace the NL term is

$$f(x) = \frac{1}{c}(1 - e^{-cx}) = x - \frac{cx^2}{2} + \frac{c^2x^3}{6} - \frac{c^3x^4}{24} + \dots ,$$

and we can easily see that to make sure the high-order terms get smaller and smaller, the constant c must be between 0 and 1. The closer it gets to 0, the closer the function gets to x itself, that is, the square of the first space derivative. However, this also means that the probability of the instabilities to happen is increased in this way. As c goes to 1, the instabilities almost always become extinct; however, the high-order NL terms' coefficients would also become more significant. Moreover, the optimal value of c depends on the value of λ since it directly varies the strength of the nonlinearity and thus also affects the probability of instabilities occurring. We first figured out that changing the value of c around the midpoint of [0 1] segment did not change much the form of the interface even though it affected a bit the growth velocity of it. Since we wanted just to get rid of the instabilities and keep the effects of the high-order terms as low as possible, after some tries with some of the optimal values of λ, we chose the value of c to be 0.3.

In the paper [11], the effective coupling parameter $g = \lambda D/2\nu^3$ was shown to be around 4π by simulating the equation for different values of g, so that the primary behaviour of KPZ equation, roughness growing proportionally with the $\beta = 1/3$ exponent, was taking place neither too "early" nor too "late". However, we were looking for the Tracy-Widom spatial fluctuations too, which show themselves in the long-time period. Therefore, we decided to continue not with the $g = 12.56 \approx 4\pi$ but with $g = 24$, so we would be sure that no saturation would occur in the long growth processes we were going to try.

In the same paper, it is explicitly stated that if λ and ν are varied, the quality of scaling is not bound to be the same even though the parameter g is kept constant. Thus, the authors also warn us not to misinterpret their result of having an optimal value of g as the signature of a fixed point.

For us, this also meant that there was simply no best choice for these parameters. With $g = \lambda D/2\nu^3 = 24$, and by taking $\nu = D = 1$, we would get $\lambda = 4\sqrt{3} \approx 7$, which would make a simple set of parameters, and hopefully be efficient since having a high NL coefficient is advantageous for the asymmetric T-W distribution of the fluctuations to pop up early in the process. After many tries, we confirmed our hypothesis about this "optimal" parameter configuration. The system we were simulating was definitely in the KPZ universality class.

We saw that the scaling exponent was quite robust when the parameters were changed, whereas the moments describing the distribution of the height fluctuations were more "elastic" compared to it.

On the way, we had to change them and try new values again; however, we got the final configuration to be as below:

$$\lambda = 7 \qquad \nu = D = 1 \qquad c = 0.3\ ,$$

which seems to be pretty practical and efficient.

6. *Noise*

The theoretical noise $\xi(x,t)$ is Gaussian white noise with 0 mean and unit standard deviation. To produce this stochastic variable changing in space and time, we used the *randn(1,L)*, a built-in MATLAB function, to produce a vector of L normally distributed random numbers on each time-step. Then, to include the dimensions properly, we have to divide this vector by the square root of the space and time elements. Thus, together with the D coefficient of it, the complete noise term is as below:

$$\sqrt{D}\xi(x,t) = \sqrt{\frac{D}{dx\,dt}}R\ ,$$

where R represents the vector of random numbers with 0 mean and unit standard deviation.

7. *Runge-Kutta Method*

The KPZ equation is written as

$$\frac{\partial h(x,t)}{\partial t} \ = \ f(t,h)\ ,$$

where we have defined the function to be used in Runge-Kutta (RK4) method as below:

$$f(t,h) \ \equiv \ \frac{\lambda}{2}\left(\frac{\partial h(x,t)}{\partial x}\right)^2 + \ \nu\ \frac{\partial^2 h(x,t)}{\partial x^2} \ + \ \sqrt{D}\,\xi(x,t)\,.$$

We are going to integrate over time starting from the initial condition $h(\mathbf{x},0) = 0$, and finding the height values the system takes step-by-step up to a total time of 5000 seconds. We do it by calculating $h(\mathbf{x},t)$ on each time-step, which, by the way, is preferred to be small enough so that we have a good approximation. Using the initial condition, we can easily find our "new" height, that is, the height on the first time-step. In the RK4 method, the idea for having a better approximation of estimating the slope of the line ($\dot{h}$), specified by the function $f(t, h)$, is to average the slopes of the line calculated at each one-fourth of the time-step and giving greater weight to the slopes at the midpoint of the time-step. Particularly, it is done by finding the height at the next time-step ($t_{new} = t_{old} + dt$) as:

$$h_{new} = h_{old} + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) dt \ ,$$

where

$$k_1 = f(t_{old},\, h_{old})$$

$$k_2 = f(t_{old} + \frac{dt}{2}, h_{old} + dt\,\frac{k_1}{2})$$

$$k_3 = f(t_{old} + \frac{dt}{2}, h_{old} + dt\,\frac{k_2}{2})$$

$$k_4 = f(t_{old} + dt, h_{old} + dt\, k_3) \ .$$

In this way, we are ensuring a pretty good simulation of the equation by using one of the best methods for temporal discretisation, such as the widely used RK4.

8. *Number of Runs*

However, we should not forget that we are simulating a stochastic partial differential equation rather than a usual PDE. And, since we are going to analyse the height fluctuations, which are a kind of random variable, we definitely need a significant number of samples so that our probabilistic analysis gives stable and reliable results. For this reason, we repeat the same simulation N times and collect all their data in a single set, which will be our ensemble. Naturally, the greater the ensemble, the more reliable the analysis; that is why we run the simulation for at least 100 times.

3.2 Results and Discussion

Now, first, let us visually check what this simulation produces. What are the interfaces under KPZ dynamics looking like?

This is shown in figure 3.1 where we can see the interface in three different moments during the process and observe how it changes over time.

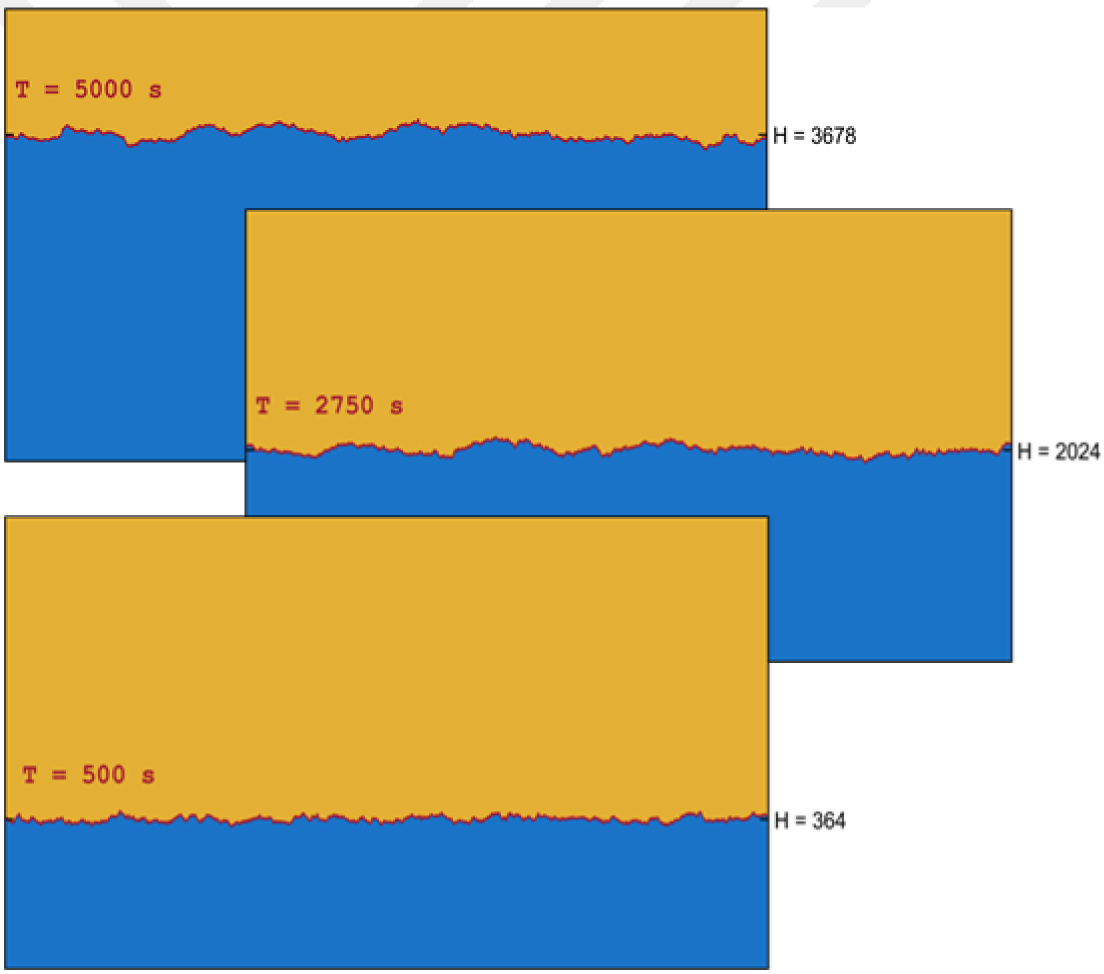


Figure 3.1: Interface moving up under KPZ dynamics.

We can easily see from here that the roughness of the interface increases as time goes on. But, is it growing proportional to time itself?

3.2.1 Roughness' Width

The answer to it is given in the following figure, showing the growth of the interface's width over time.

The best way to quantify the width of the spatial roughness of the interface is to calculate the standard deviation of all the height values over space at a fixed time. We define thus the overall width of the interface as

$$width \equiv std(h) = \sqrt{\frac{1}{L-1}\sum_{i=1}^{L} |h_i - \mu|^2} \ ,$$

where μ is the mean of all the height values:

$$\mu = \frac{1}{L}\sum_{i=1}^{L} h_i \ .$$

1. *Roughness' Width over Time*

Since roughness' growth is a power law, to best show the exact value of the exponent, we plot the width values as they change in time in a so-called log-log graph, where each of the axes is in the logarithmic scale. Thus, the power law in the log scale produces a linear graph with slope equal to the exponent.

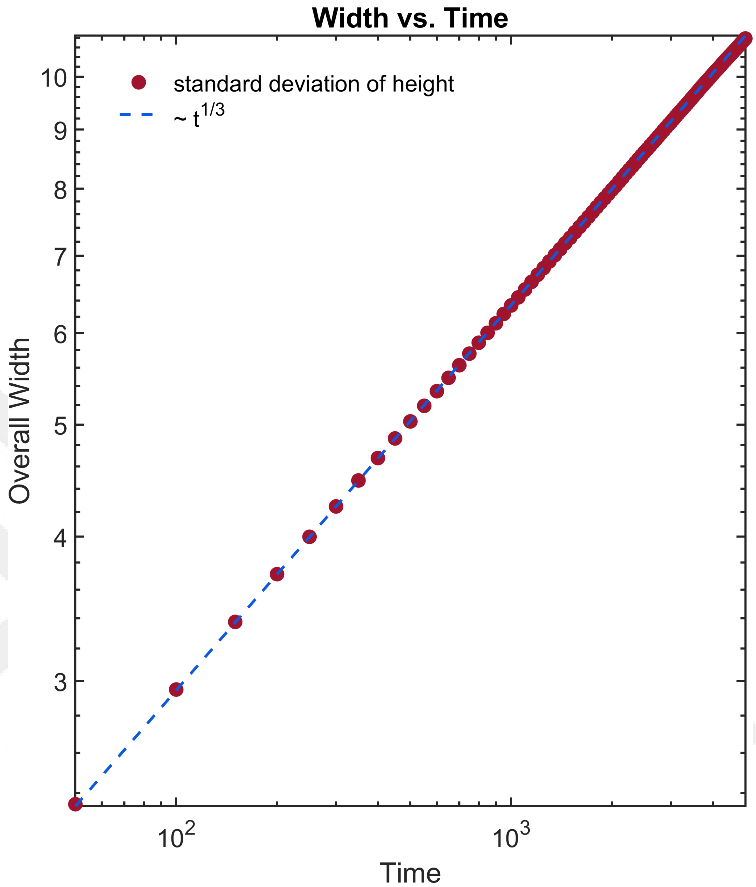


Figure 3.2: Interface width growth versus time.
Data taken from 10 000 runs, with $L = 2^{13}$ and Time = 5000 s.

We see that width is growing proportionally to $t^{1/3}$, as expected for the KPZ dynamics. This directly indicates that the interface simulated is in the KPZ universality class.

Now, we look more closely at the fluctuations in the interface. We can detect two types of height fluctuations, namely, spatial and temporal fluctuations.

3.2.2 Spatial Fluctuations

The spatial fluctuations (SF) are the variations that the height of the interface has over space at fixed times; that is, they are the quantities describing the roughness of the interface when frozen at a moment of time.

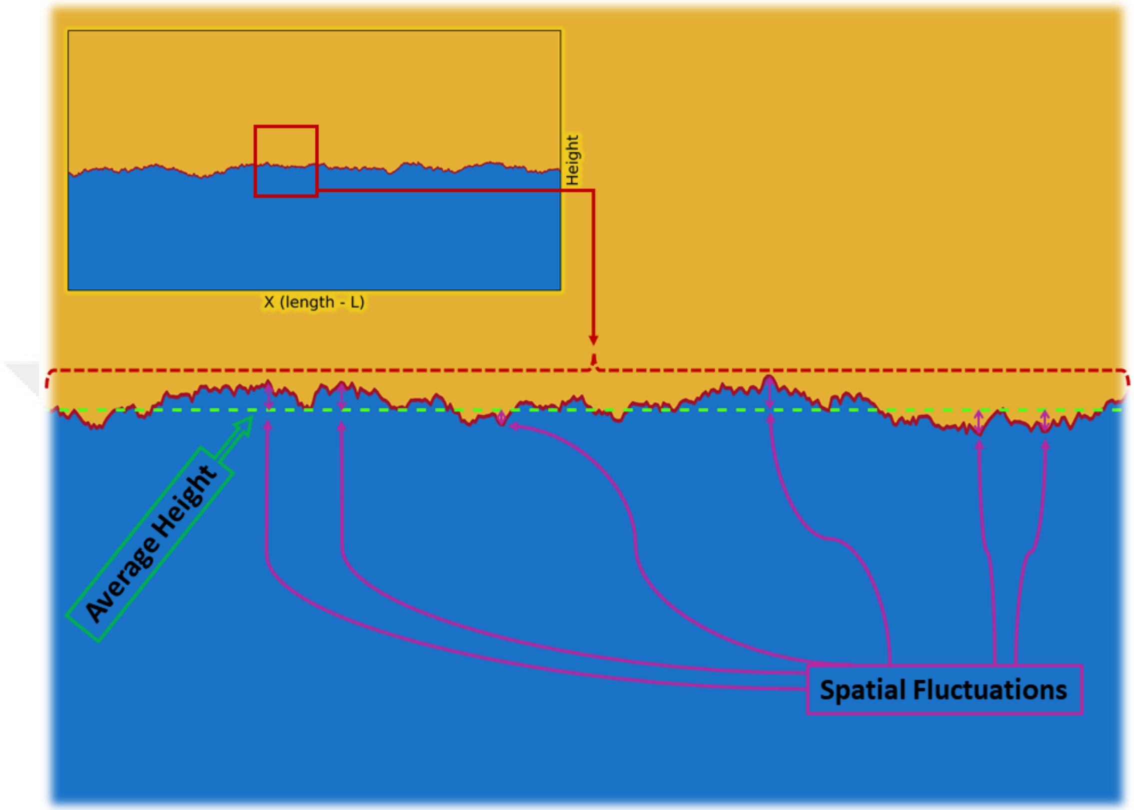


Figure 3.3: Spatial fluctuations in KPZ dynamics.

Those fluctuations are defined as shown below

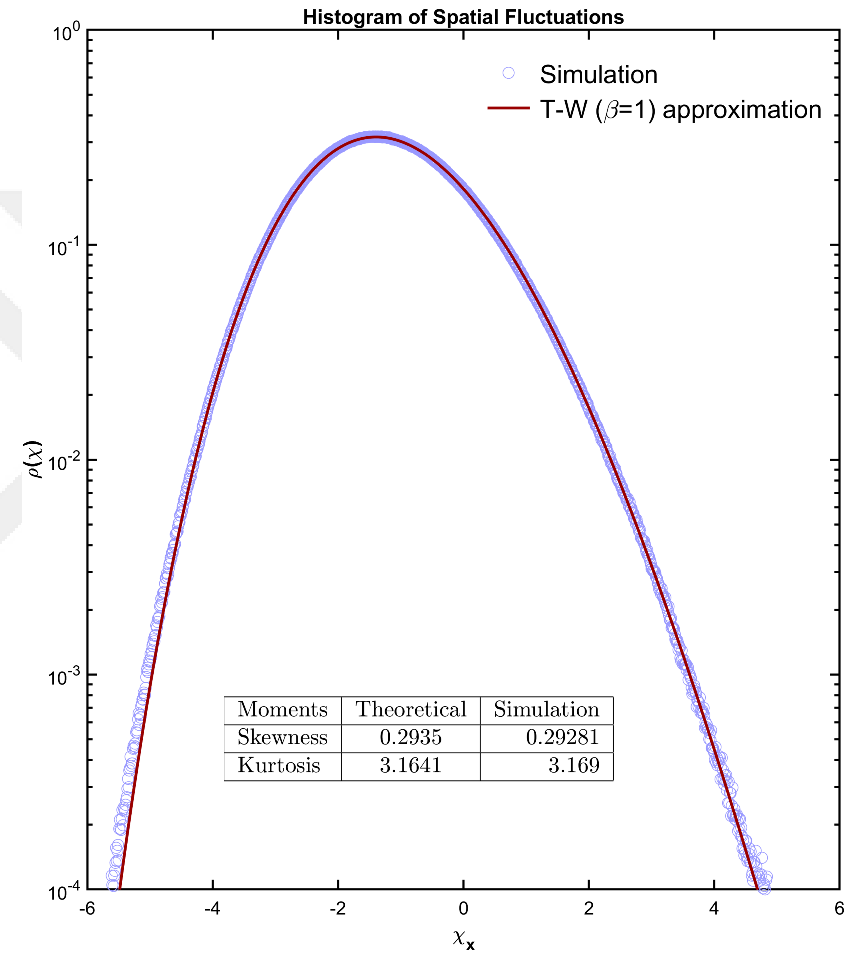
$$\chi_x(t) \equiv \frac{h(t) - \langle h(t) \rangle}{\sqrt{\langle (h(t) - \langle h(t) \rangle)^2 \rangle}} \; ,$$

where the angle brackets are denoting the average "operator". Thus, at a fixed time t, we find the interface's SF by subtracting the interface's mean height from each interface point's height and dividing this by the standard deviation of these variations.

Then, the statistical properties of these height fluctuations are analyzed so that we obtain some other quantitative properties to describe the roughness of the interface.

1. *Probability Distribution of SF*

The distribution of the spatial fluctuations is as follows:



Moments	Theoretical	Simulation
Skewness	0.2935	0.29281
Kurtosis	3.1641	3.169

Figure 3.4: The distribution of spatial fluctuations.
Data taken from 10 000 runs, with $L = 2^{13}$ and Time = 5000 s.

Our simulation evidently produces spatial fluctuations on the interface with the Tracy-Widom ($\beta = 1$) distribution. Again, this is expected for the 1-D flat systems of the KPZ universality class.

To see how this distribution changes in time, we calculate the first eight moments of the distribution as it changes throughout the growing process of the interface.

2. *Skewness and Kurtosis of SF Distribution*

But firstly, let us see how only the third and fourth moment does change with time and how close to T-W they are during the process:

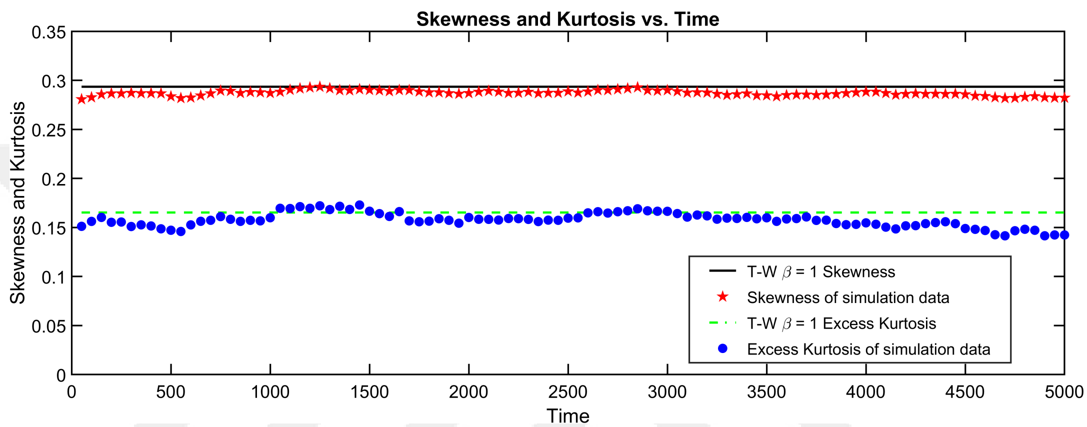


Figure 3.5: Change of skewness and kurtosis of the distribution of SF over time. Data taken from 10 000 runs, with $L = 2^{13}$ and Time $= 5000$ s

From the skewness and kurtosis graph, we understand that the distribution of the spatial fluctuations is T-W for most of the time of the process.

3. *Moments of SF Distribution*

Then, let us see how close to the theoretical values of the T-W ($\beta = 1$) distribution are all the first eight moments:

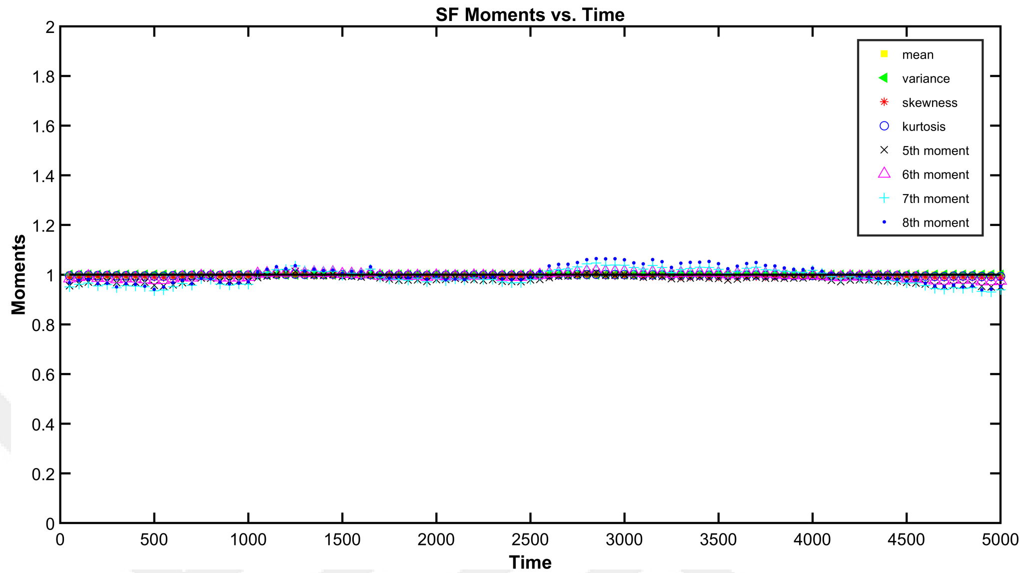


Figure 3.6: Closeness of moments of the distribution of SF to T-W ($\beta = 1$). Exact moment values are normalized using their respective T-W values. Data taken from 10 000 runs, with $L = 2^{13}$ and Time = 5000 s.

Clearly, the distribution is not changing much in time - except in the very initial period - during these 5000 seconds.

And on a specific time, their exact values are as below:

Moments	T-W ($\beta = 1$) dist	Simulation
1st (mean)	0	0
2nd (variance)	1	1
3rd (skewness)	0.293465	0.2928
4th (kurtosis)	3.16524	3.1690
5th moment	3.03709	3.0657
6th moment	18.3825	18.6575
7th moment	34.6221	36.2572
8th moment	167.491	178.3021

Table 3.1: The exact moment values for the spatial fluctuations under KPZ.

All of them are so close to the respective theoretical values for T-W ($\beta = 1$) distribution. This good agreement up to the eighth moment confirms one more time the simulation and the T-W universality of the KPZ dynamics.

3.2.3 Temporal Fluctuations

Other height fluctuations are variations of a single interface point's height over time. That is, the temporal fluctuations (TF) are the slight alterations that produce the "roughness" over the smooth growth of height.

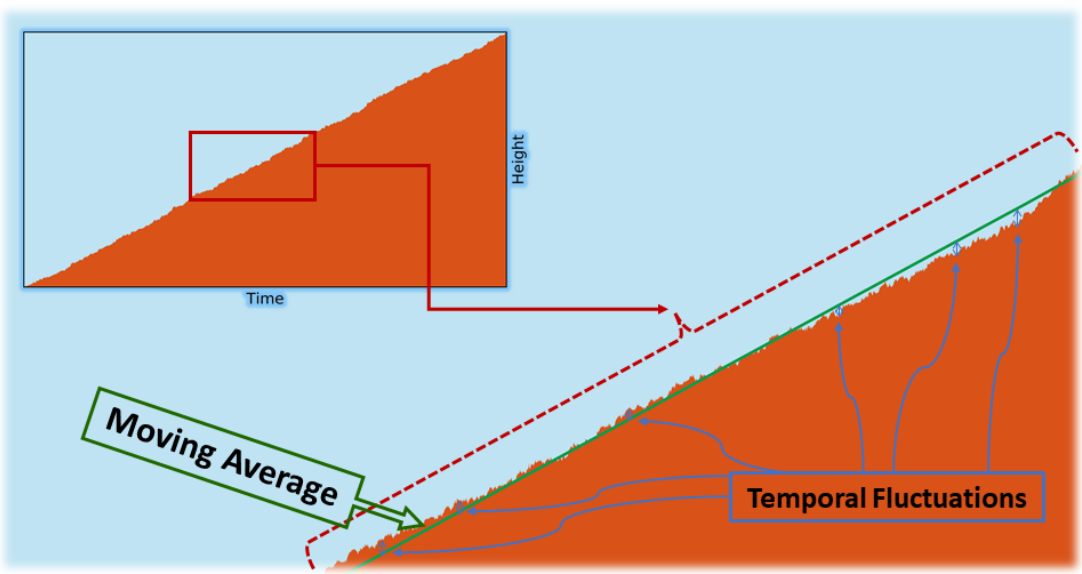


Figure 3.7: Temporal fluctuations in KPZ dynamics.

To capture these variations in time, the TF are simply defined as what is left when the overall growth of the height of the interface is filtered out. To filter out the main height growth, we subtract the smoothed version of the height function, that is, the moving average of the growing height of this upward moving interface. The TF are

$$\chi_t(x) \equiv h(x) - \overline{h(x)} \; ,$$

where the overbar ($^{-}$) represents the moving average, which is shown as the green dotted line in the figure 3.7.

Each element of the moving average vector is calculated as the average of an array of adjacent elements of $h(x)$ centred at the position of the particular element of the moving average. Thus, e.g. to find the 10th element of the moving average using arrays of length (span) of 5, we find the average of the 5 adjacent elements of $h(x)$ centred at the 10th element of $h(x)$, that is the array from 8th to 12th element of $h(x)$.

However, how do we decide on the optimal length (span) of the array used to find the moving average? Depending on it, the moving average changes, and so do the temporal fluctuations. If the span value is minimal, then the moving average will also fluctuate resembling the height growth itself, so we cannot properly extract the TF from it. Thus, the span value we have to choose cannot be small. But how much large should it be? Of course, there is a limit for the largest span value too. As we increase the span, the moving average tends to move (grow) perfectly linearly. However, that is not the case for the actual height. We cannot say that we have a perfect linear growth on average. At least, in the start, we easily see a slow growth whose slope changes much from the later part.

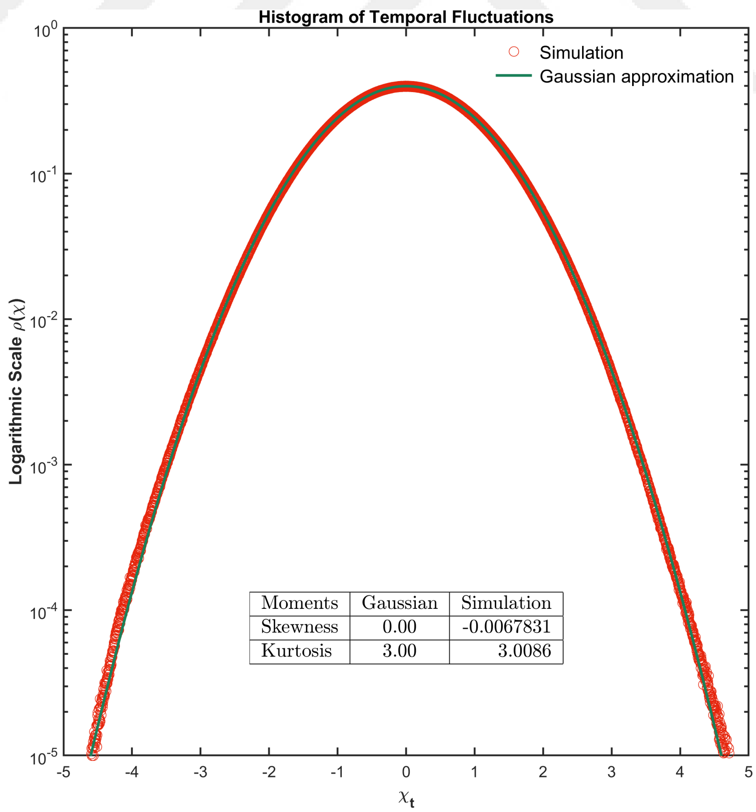
The optimal moving average we want for extracting the TF in the best way is a moving average with no fluctuations but describes the average height growth very well. Since, in our case, the growth is very close to perfect linear growth, we understand that the span value should be very large.

In the end, we do not know for sure the exact span value we have to take to find the moving average. And, since this is a kind of approximation to the best theoretical moving average, we want our results to be stable in a range of large span values (not changing very much depending on the value of the span).

For the process of 5000 seconds, we check how much the temporal fluctuations change for span values between 50 to 4000. All this wide range is enough to look for the best approximation of filtering the fluctuations from this growing height "jiggling" in time.

1. *Probability Distribution of TF*

Let us now see the distribution of temporal fluctuations for some particular good approximation of the moving average.



Moments	Gaussian	Simulation
Skewness	0.00	-0.0067831
Kurtosis	3.00	3.0086

Figure 3.8: The distribution of temporal fluctuations.
Data taken from 100 runs, with $L = 2^{13}$, Time $= 5000\,s$ and span$= 2000\,s$.

Their distribution is obviously the well-known bell-shaped Gaussian distribution. This was also expected for the KPZ equation, even though we could not find any direct mention or evidence regarding the TF in the literature. We expected the symmetrical Gaussian distribution since there is nothing nonlinear in the time domain, and from the Central Limit Theorem, we know that only a normal distribution can be produced linearly.

Next, let us make the necessary checks to see how stable this distribution is. We again calculate the first eight moments of the distribution and see how they change with the span, that is, the window size over which we calculate the moving average needed to high-pass filter the temporal fluctuations.

2. *Skewness and Kurtosis of TF Distribution*

First, let us see the behaviour of only the third and fourth moments, then compare all 8 of them to the respective theoretical values of Gaussian distribution.

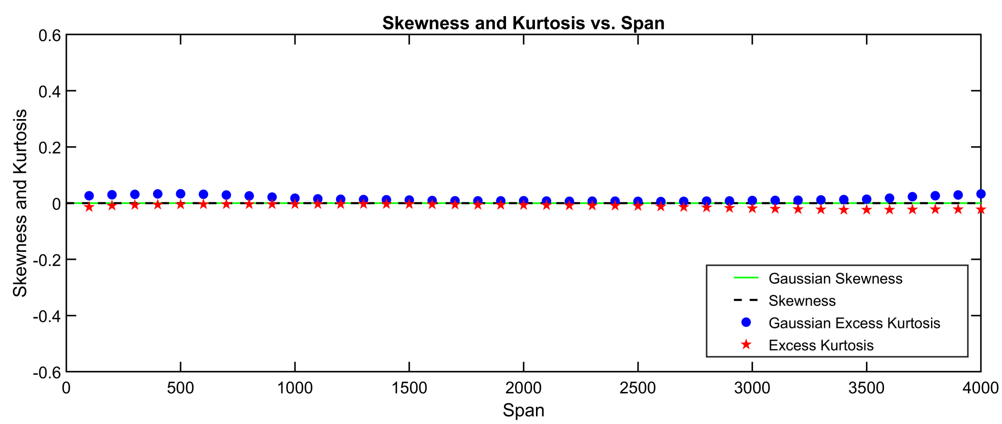


Figure 3.9: Skewness and kurtosis of the distribution of TF.
Data taken from 100 runs, with $L = 2^{13}$ and Time= $5000\,s$.

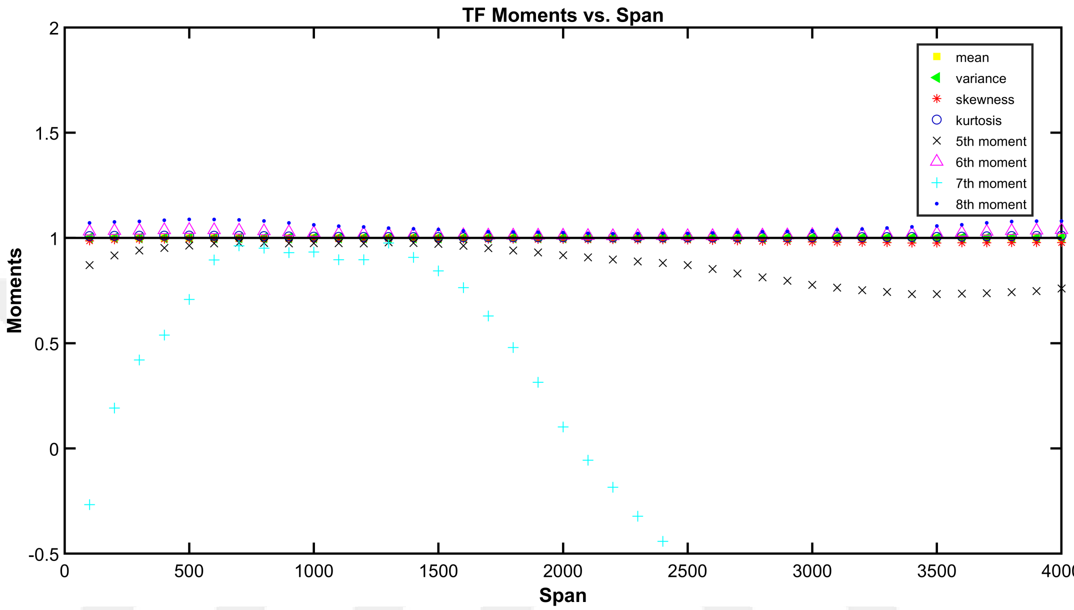


Figure 3.10: Closeness of moments of the distribution of TF to Gaussian distribution. Exact moment values are normalized using their respective Gaussian values. Data taken from 100 runs, with $L = 2^{13}$ and Time$= 5000\,s$.

It seems that changing the span of data over which the moving average is calculated cannot change the fact that the distribution of temporal fluctuations is Gaussian. Nevertheless, we had to keep the span values relatively high (as shown in the graphs) to have the proper high-pass filtering.

And, for some particular span value, we have the following exact values of the first eight moments, whose closeness to their respective theoretical values of Gaussian distribution is obvious:

Moments	Gaussian dist	Simulation
1st (mean)	0	0
2nd (variance)	1	1
3rd (skewness)	0	-0.0068
4th (kurtosis)	3	3.0086
5th moment	0	-0.0818
6th moment	15	15.1671
7th moment	0	-0.8980
8th moment	105	107.6323

Table 3.2: The exact moment values for the temporal fluctuations under the KPZ equation.

Thus, up to now, we saw that our KPZ equation simulations are consistent with the theory, producing growing interfaces with the $\beta = 1/3$ scaling exponent, roughness in Tracy-Widom distribution and Gaussian temporal fluctuations. Being sure of the simulation method and details for the KPZ equation, now we can start modifying it and check the differences using the same methods.

Chapter 4

Numerical Analysis of the Proposed Modified KPZ Equation

As explained in chapter 2, we had strong motivation to try and make some alterations to the KPZ equation, or come out with a new similar equation, so that the function $h(x,t)$ would have some fluctuations in the time domain not normally distributed, but with at least some skewed distribution, such as the Tracy-Widom.

The results from the DSA experiments were clear. Our job was to mathematically "predict" the interface's dynamics causing it to fluctuate in that manner.

We had made minor changes in the terms of the KPZ equation and looked for their effects. Things like increasing the power of the NL term, going from the first derivative squared to the square of the second derivative or of the height itself, changing the order of the derivative in the diffusion term, or changing the colour of the noise used from white to brown, would have very small or extreme effects (neglectful effect or causing divergences) on the results. However, those effects were only seen in the space domain, and things would remain the same regarding the fluctuations in the time domain.

Actually, this was somewhat expected since we already knew from the Central Limit Theorem that having any noise in the time domain, as long as we were not acting on it with an NL operator, would always produce Gaussian distributed noise.

After simulating the KPZ equation and the "other versions" of it mentioned above, the crucial effect of the nonlinearity in the height fluctuations was obvious. However, the only NL term we had was affecting only the space domain. We definitely needed such a term in the time domain.

When evaluating the way Kardar et al. introduced the nonlinear term (as explained a bit in chapter 2), we see that their motivation was the lateral growth they had observed in the deposition models (such as ballistic or Eden models), in which a new kind of dynamics had been shown, particularly signified by the scaling exponent being equal to 1/3 instead of 1/4.

Checking the geometry of the lateral growth in the growing interfaces, they could beautifully derive the nonlinear term they decided to add to the Edward-Wilkinson equation, which described the dynamics of the kinetic interfaces lacking lateral growth or similar NL effects.

Since we also had to come up with a new term in the time domain, we as well had to carefully observe the temporal dynamics of the experimental system and check for any nonlinear behaviour that might be causing the asymmetry in the distribution of the temporal fluctuations.

One apparent behaviour that would be a good candidate for the explanation of the temporal nonlinearity in the system was the decrease in speed of the incoming particles as the size of the aggregate increased. This behaviour has been meticulously observed and very well explained by Ilday et al. [49, 4] as a part of the feedback mechanisms governing the complex dynamics of self-assemblies. These rich feedback mechanisms are expected to induce the temporal correlations.

In brief, we understand from the observed behaviour that the amount of how fast will the aggregate grow in a later time is affected by its growth rate at the present moment.

To isolate the effect of this behaviour on the height function, we need to start by the simplest growth equation: $\partial h/\partial t = v$. When the growth velocity is constant, the second derivative of height in time is trivial. However, it will not be so anymore as the observed behaviour is introduced. To check how will it change during the growth process, we look at its forward finite difference approximation at time t, and carefully examine how the observed behaviour can be manifested even at a single time-step.

In the figure below, we can schematically show by exaggeration how the height is affected. The illustration shows the height at time t, and how it is going to change in the next two time-steps. We are keeping the time-steps constant, so it means that any change in the speed of particles will be reflected only on the small increase in height.

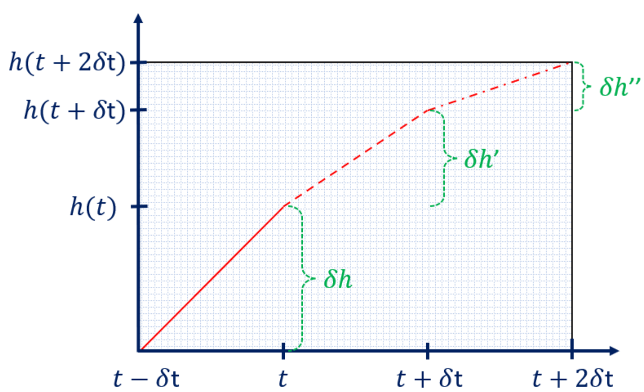


Figure 4.1: Height on each time-step.

Thus, if the incoming particles' speed is decreased depending on how much the size of the aggregate was increased, we can say that the next height element will be smaller than what it had to be depending on the size of the previous height element. Mathematically, it can be written as $\delta h_{new} = \delta h\,(1 - \alpha\,\delta h)$, where α is a constant.

The height after one time-step is $h(t+\delta t) \;=\; h(t) \;+\; \delta h\,(1-\alpha\,\delta h)$, where the next height increment has been decreased. Then, we apply the same operator to the next-next element. We see that

$$\begin{aligned}\delta h_{new} \;&= \delta h\,(1-\alpha\,\delta h)[1 \;-\; \alpha\,\delta h\,(1-\alpha\,\delta h)] \\ &= \delta h - 2\,\alpha\,(\delta h)^2 + 2\,\alpha^2\,(\delta h)^3 - \alpha^3\,(\delta h)^4 \;,\end{aligned}$$

which after neglecting the higher-than-2 powers of δh, becomes simply $\delta h_{new} = \delta h\,(1-2\alpha\,\delta h)$.

Now we can find the height after two time-steps as $h(t+2\delta t) \;=\; h(t) \;+\; \delta h\,(1-\alpha\,\delta h) \;+\; \delta h\,(1-2\alpha\,\delta h)$.

The forward finite difference approximation for the second time derivative of height at time t is:

$$\frac{\partial^2 h}{\partial t^2} \;=\; \frac{h(t+2\delta t) - 2\,h(t+\delta t) + h(t)}{(\delta t)^2}$$

Substituting the height values of the upcoming time-steps, we see that

$$\frac{\partial^2 h}{\partial t^2} \;=\; \frac{h(t) \;+\; \delta h\,(1-\alpha\,\delta h) \;+\; \delta h\,(1-2\alpha\,\delta h) - 2\,[h(t) \;+\; \delta h\,(1-\alpha\,\delta h)] + h(t)}{(\delta t)^2}$$

$$\frac{\partial^2 h}{\partial t^2} \;=\; \frac{\delta h\,(1-2\alpha\,\delta h) - \delta h\,(1-\alpha\,\delta h)}{(\delta t)^2} \;=\; -\alpha\,\frac{(\delta h)^2}{(\delta t)^2} \;=\; -\alpha\,\left(\frac{\partial h}{\partial t}\right)^2$$

since the first time derivative at time t is $\dfrac{\partial h}{\partial t} \;=\; \dfrac{\delta h}{\delta t}$.

In this way, translating the observed behaviour into mathematics, we could relate the second time derivative of the height function with the square of its first time derivative, which is a nonlinear component.

This analysis, together with the Brownian motion of the particles, was alluding to both, a nonlinear and a temporal diffusion term. The nonlinear term in the time domain being the square of the first time derivative and the temporal diffusion term being the second time derivative made us realize that they were actually the analogue versions of the respective terms in the space domain of the KPZ equation. Thus, we simply labeled the coefficients of these new terms according to the parameters in the KPZ equation. Moving the temporal NL term on the left hand of the equation changed its sign to be positive and finally, we had this new equation

$$\frac{\lambda_t}{2}\left(\frac{\partial h(x,t)}{\partial t}\right)^2 + \nu_t \frac{\partial^2 h(x,t)}{\partial t^2} + \frac{\partial h(x,t)}{\partial t} = \frac{\lambda_x}{2}\left(\frac{\partial h(x,t)}{\partial x}\right)^2 + \nu_x \frac{\partial^2 h(x,t)}{\partial x^2} + \sqrt{D}\xi(x,t)\,, \tag{4.1}$$

which is a modified version of the KPZ equation.

The next thing to be done was to simulate it and maybe modify it more according to the results we see so that we have the final version of some equation describing growing interfaces with temporal fluctuations in the T-W distribution.

Now, the difference with simulating the KPZ equation was that the number of parameters of the equation had become 5. Moreover, for the new NL term, we needed to use the sigmoidal function we were using for the one in the space domain since divergences caused by the discretisation would come again. This also meant two extra parameters, previously used c, and the new adjustable constant c_t.

Our first intention was not to change anything in the space domain, vary only the temporal terms, and then later see the effects of these new terms on the spatial fluctuations. Thus, we decided to keep previously used parameter values the same as in the KPZ simulations. They continued to be exactly as

$$\lambda_x = 7 \qquad\qquad \nu_x = D = 1 \qquad\qquad c_x = 0.3\,.$$

Doing this, we only had to decide on the new parameters, but we had no clue about what their values could be. Since this is the first work we know on adding new terms in the time domain to the KPZ equation, we could not find anything in the literature too, and the simplest way to go seemed to try many values as we could and see how they affected the interface growth.

We started by taking the same values as their respective parameters for the terms in space domain, specifically

$$\lambda_t = \lambda_x = 7 \qquad\qquad \nu_t = \nu_x = 1 \qquad \text{and} \qquad c_t = c_x = 0.3\ .$$

We saw that the interface this time was "growing" in the negative region so that our height became a depth, but the dynamics seemed not that much different. Also, the "growth" was pretty linear again. This meant that we were on the right track. We had this moving interface, whose behaviour, at first sight, was still resembling the dynamics in the KPZ equation, being produced by some new stochastic PDE having two additional temporal terms.

Then, we continued by changing the values of each of the new parameters and observed their effects. It turned out that decreasing the coefficient of the new NL term λ_t, or otherwise increasing the coefficient of the diffusion term ν_t, were both causing increase in the height of the interface.

For $\lambda_t = \lambda_x/2$, $\nu_t = 3\,\nu_x/2$ and $c_t = c_x$, we could take very nice growing interfaces similar to the ones in KPZ dynamics and with clear observable temporal fluctuations.

Analysing the data gave us great hope since the temporal fluctuations were no more in the Gaussian distribution. Moreover, their distributions were negatively skewed and with moments not so far from the T-W region.

It required many tries and errors, but in the end, we could see that within some range of parameters, the distribution of the TF was the "mirrored" T-W, precisely as the one in the experimental data.

Since we were simulating a 1+1 dimensional equation with a flat substrate as an initial condition, we were not expecting the same results with the experiments, but it turned out that it was the case.

We saw that for the below configuration of parameters:

$$\lambda_x = 7\,, \quad \lambda_t = 3.4\,, \quad \nu_x = 1\,, \quad \nu_t = 1.3 \quad \text{and} \quad D = 1\,,$$

we had obtained T-W distributed temporal height fluctuations of the interface points.

Now, before going on with the results, let us first see the details of the methods used in the simulation of this new equation.

4.1 Method

As the new equation was obtained by modifying the KPZ equation, we then accordingly modified the simulation code of the KPZ equation as well, and we tried to keep as many parts unchanged as possible.

1. *Integration Elements*

Thus, the integration elements were kept the same as: $dx = 1$ and $dt = 0.01$.

2. *Initial Conditions*

Unlike the KPZ equation, our new equation is second order in the time domain; thus, we need two initial conditions. The first, height at time 0 is unchanged, a flat substrate as $h(\mathbf{x}, 0) = h_0 = 0$. Whereas the second initial condition will be the value of the first time derivative of height at the start of the growth process, which is chosen to be constant over all space and small enough just to start the growth but not have a big effect on later dynamics. It was chosen to be

$$\frac{\partial h(\mathbf{x}, 0)}{\partial t} = 0.01 \ .$$

3. *System Size and Boundary Conditions*

The boundary conditions in space continued to be periodic for the same reasons as in the KPZ equation, and we also continued to have the same length of the substrate as $L = 2^{13}$. In the time domain, since for this new equation, the scaling can change, looking at the evolution of the spatial fluctuations, we decided to increase the total time of the process so we could see the convergence of the distribution of spatial fluctuations to some particular distribution. The new total time is 10 000 seconds.

4. *Adjusting the Nonlinear Terms*

Since the discretisation problem causing the NL terms to diverge is now present in both NL terms of this new equation, we use the same sigmoidal function $f(x) = (1 - e^{-cx})/c$ for both terms to get rid of all instabilities. Adding all the higher-order terms with very small coefficients worked well again and prevented any instability from happening.

5. *Parameters*

As discussed above, the parameters in the space domain were kept the same, whereas the new ones are varied in some range with values not far from their respective parameter values in space.

6. *Noise*

The type of noise did not change too. It is white Gaussian noise with 0 mean and unit standard deviation.

7. *Runge-Kutta Method*

We again used the Runge-Kutta (4th order) method to integrate this new equation.

To simulate this new equation, we first have to convert it to a system of two of them, which are first-order differential in the time domain. So, by defining $v \equiv \partial h/\partial t$, we have the system:

$$\begin{cases} \frac{\partial h}{\partial t} = v \\ \frac{\partial v}{\partial t} = \frac{1}{\nu_t} \left[-v - \frac{\lambda_t}{2} v^2 + \frac{\lambda_x}{2} (\frac{\partial h}{\partial x})^2 + \nu_x \frac{\partial^2 h}{\partial x^2} + \sqrt{D}\, \xi \right] \end{cases} \tag{4.2}$$

Now, let's define $\mathbf{q} \equiv \begin{bmatrix} h \\ v \end{bmatrix}$, and try to write the above system of equations as a single equation in terms of $\mathbf{q}$.

$$\begin{bmatrix} v \\ \frac{1}{\nu_t} \left(-v - \frac{\lambda_t}{2} v^2 + \frac{\lambda_x}{2} (\frac{\partial h}{\partial x})^2 + \nu_x \frac{\partial^2 h}{\partial x^2} + \sqrt{D}\, \xi \right) \end{bmatrix} = \begin{bmatrix} \frac{\partial h}{\partial t} \\ \frac{\partial v}{\partial t} \end{bmatrix} = \frac{\partial \mathbf{q}}{\partial t} \tag{4.3}$$

Using some simple matrices, it becomes:

$$\dot{\mathbf{q}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{q} + \frac{1}{\nu_t} \left\{ \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} \left(\mathbf{q} + \frac{\lambda_t}{2} \mathbf{q}^2 \right) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left(\frac{\lambda_x}{2} (\frac{\partial h}{\partial x})^2 + \nu_x \frac{\partial^2 h}{\partial x^2} + \sqrt{D}\, \xi \right) \right\} . \tag{4.4}$$

Finally, it is:

$$\dot{\mathbf{q}} = \mathbf{G}\mathbf{q} + \frac{1}{\nu_t} \left\{ \mathbf{M} \left(\mathbf{q} + \frac{\lambda_t}{2} \mathbf{q}^2 \right) + \mathbf{R} \left(\frac{\lambda_x}{2} (\frac{\partial h}{\partial x})^2 + \nu_x \frac{\partial^2 h}{\partial x^2} + \sqrt{D}\, \xi \right) \right\} , \tag{4.5}$$

where $\mathbf{G}$, $\mathbf{M}$ and $\mathbf{R}$ are the corresponding matrices in the previous equation.

So, this last equation is ready to be numerically integrated by the Runge-Kutta (4th order) method, with just a bit more care for the last part of it, having only dynamics in the space domain, and not forgetting that the variable now is a 2-by-1 matrix. In the end, we always extract h from $\mathbf{q}$ and then analyse its growth properties.

8. *Number of Runs*

Again, for the same reasons, we have to use a great amount of samples to do a proper statistical analysis, and here we saw that it was enough to use the same number of runs as in the case of the KPZ equation to have stable and credible results.

4.2 Results and Discussion

Now, let us check carefully what these new terms have introduced into the dynamics of the growing interface and how far the results are from what we saw in the case of the KPZ equation. All the analysis methods are the same as what is used for the KPZ equation. Thus, there is no change in the roughness' width formula and the definitions of spatial and temporal fluctuations.

The first figure shows the interface as it grows in the new dynamics.

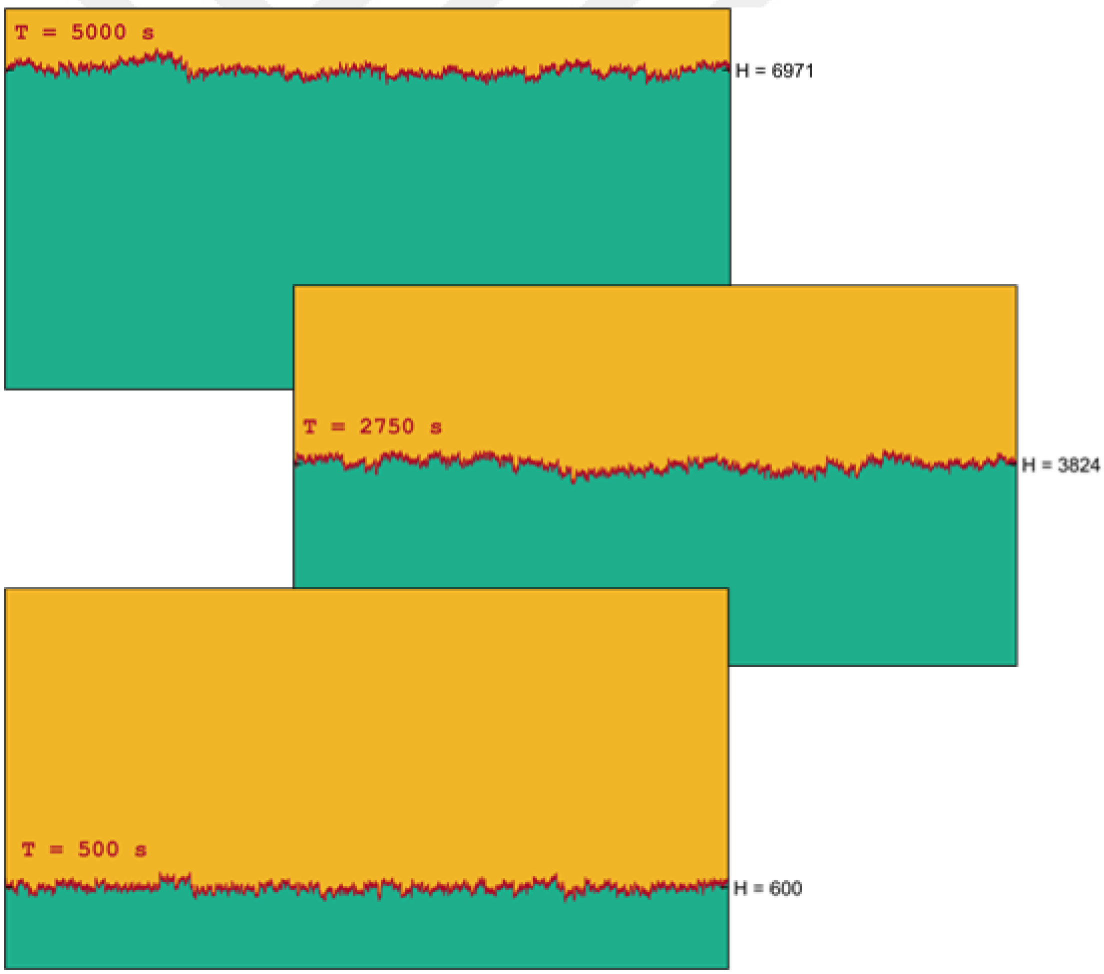


Figure 4.2: Interface moving up under new dynamics.

We see that the roughness of interface is increasing as time goes on, similar to the growth under KPZ equation. But is it also growing proportional to $t^{1/3}$?

4.2.1 Roughness' Width

Let us see the graph of the width of the interface versus time to find the answer.

1. *Roughness' Width over Time*

The standard deviation of the heights again well represents width. We increased the time of the process to 10000 seconds to better see the growing behaviour.

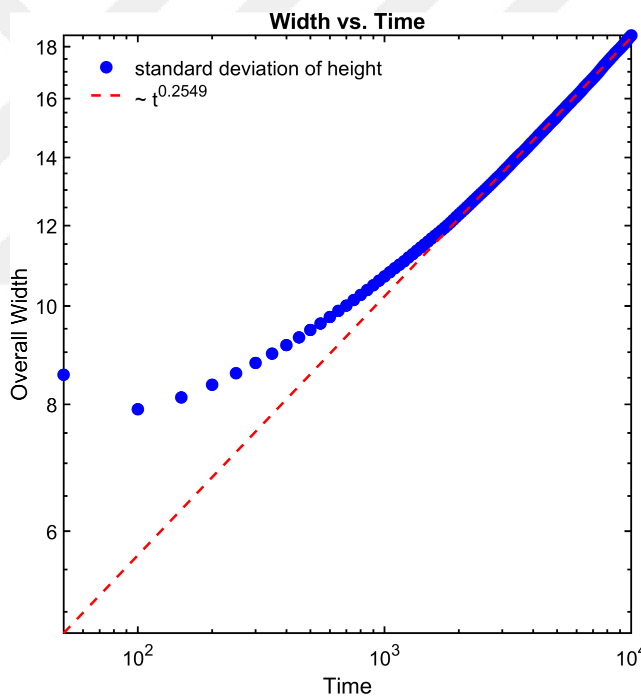


Figure 4.3: Interface width growth versus time.
Data taken from 10000 runs, with $L = 2^{13}$ and Time = 10000 s.

We see that the best approximation is very close to 1/4. The lines coincide for most of the process, from 2000 to 10 000 seconds. It is the same exponent as for the E-W universality class for 1+1 dimensional systems and also very close to the numerically found exponent for the 2+1 dimensional systems of KPZ universality class, usually found to be 0.24.

Let us now look at what happens with the distribution of the spatial fluctuations and then see if we have passed on another universality class or not.

4.2.2 Spatial Fluctuations

Spatial fluctuations are defined by the same formula:

$$\chi_x(t) \equiv \frac{h(t) - \langle h(t) \rangle}{\sqrt{\langle (h(t) - \langle h(t) \rangle)^2 \rangle}} .$$

And, to see how they look in these new dynamics, we have this figure in our help.

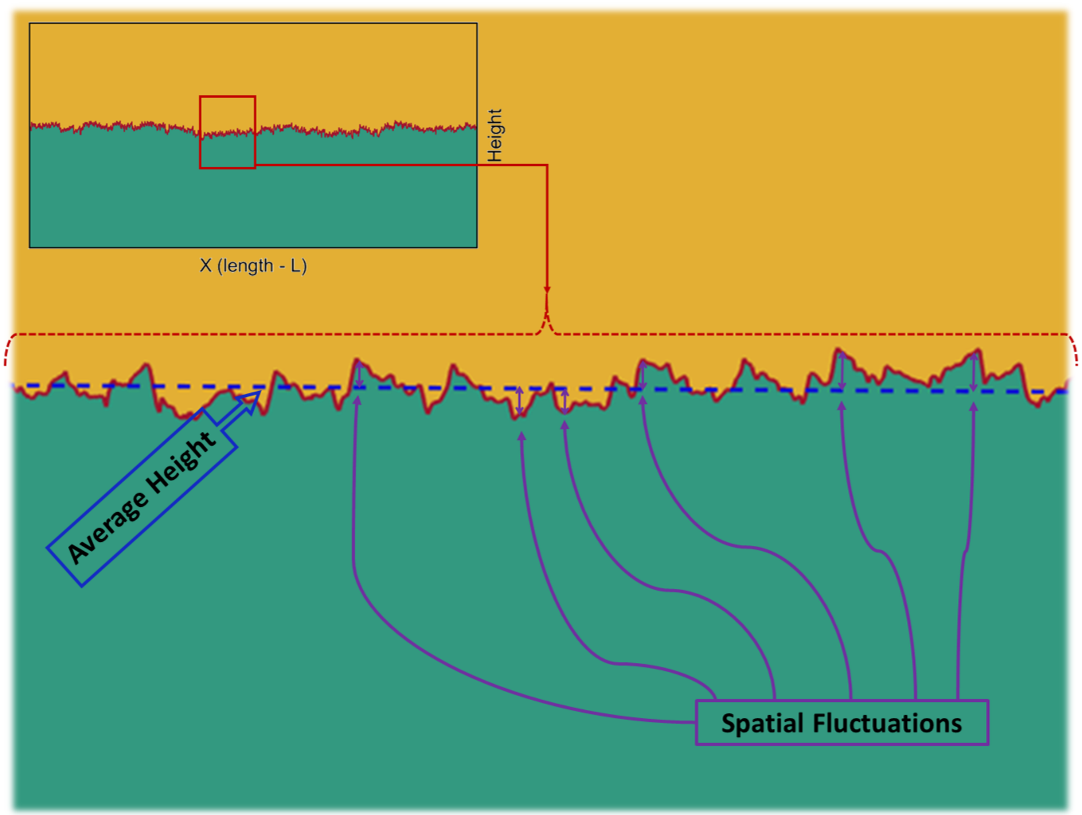


Figure 4.4: Spatial fluctuations in the new dynamics.

1. *Probability Distribution of SF*

Then we plot the histogram of these fluctuations showing their probability distribution as follows:

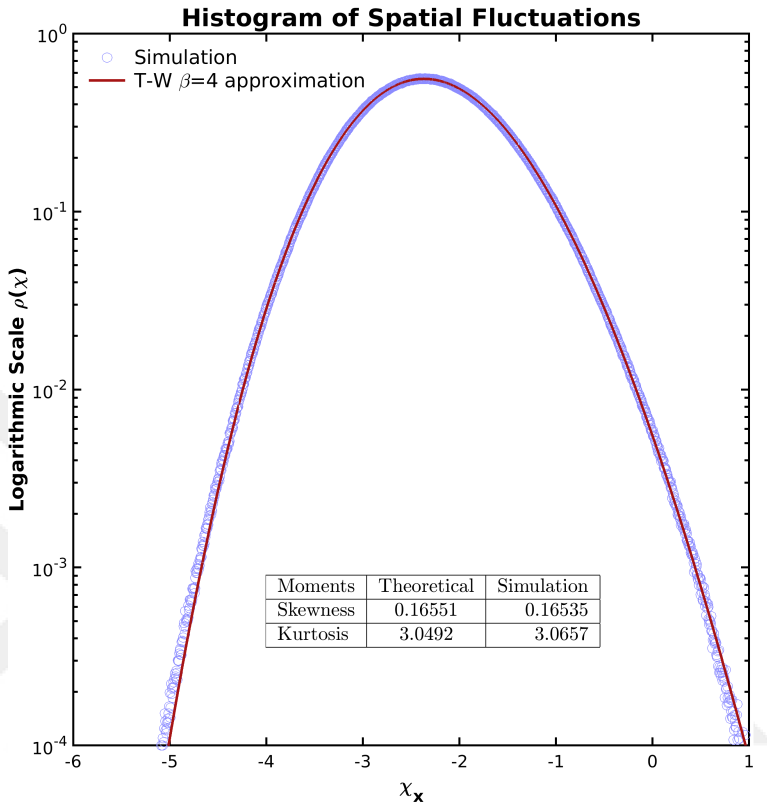


Figure 4.5: The distribution of spatial fluctuations.
Data taken from 10000 runs, with $L = 2^{13}$ and Time = 10000 s.

The spatial fluctuations from this new equation are in the Tracy-Widom $\beta = 4$ distribution. They are not anymore at the T-W $\beta = 1$ distribution as in the case of the KPZ equation, but they have not left the T-W region. This form of the Tracy-Widom distribution is the distribution of the largest eigenvalues of random matrices in the Gaussian Symplectic Ensemble. The remarkable thing about it in our discussion is that it is the first time that this type of T-W is related to the growth phenomena.

2. *Skewness and Kurtosis of SF Distribution*

The change with time of skewness and kurtosis up to their convergence is shown below. This graph shows that kurtosis converges almost immediately, whereas skewness needs much time to converge completely at the respective T-W $\beta = 4$

distribution's moment.

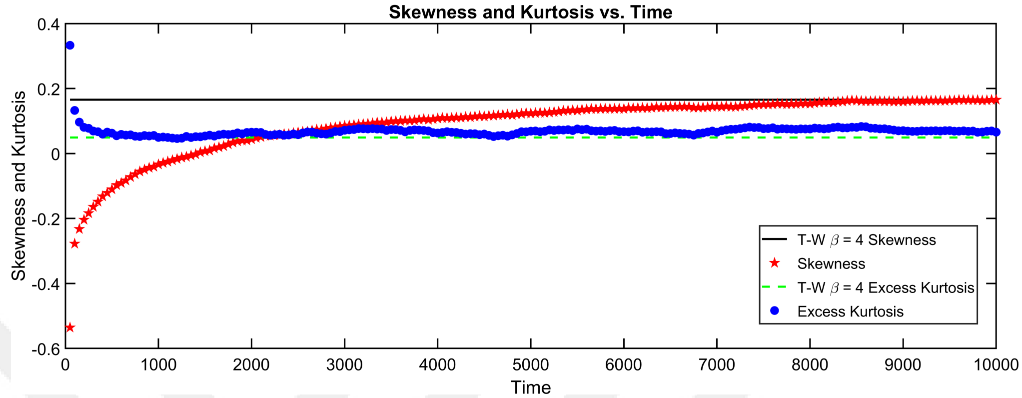


Figure 4.6: Change of skewness and kurtosis of the distribution of SF over time. Data taken from 10000 runs, with $L = 2^{13}$ and Time = 10000 s.

3. *Moments of SF Distribution*

Then, let us see how close to the theoretical values of the T-W ($\beta = 4$) distribution are all the first eight moments:

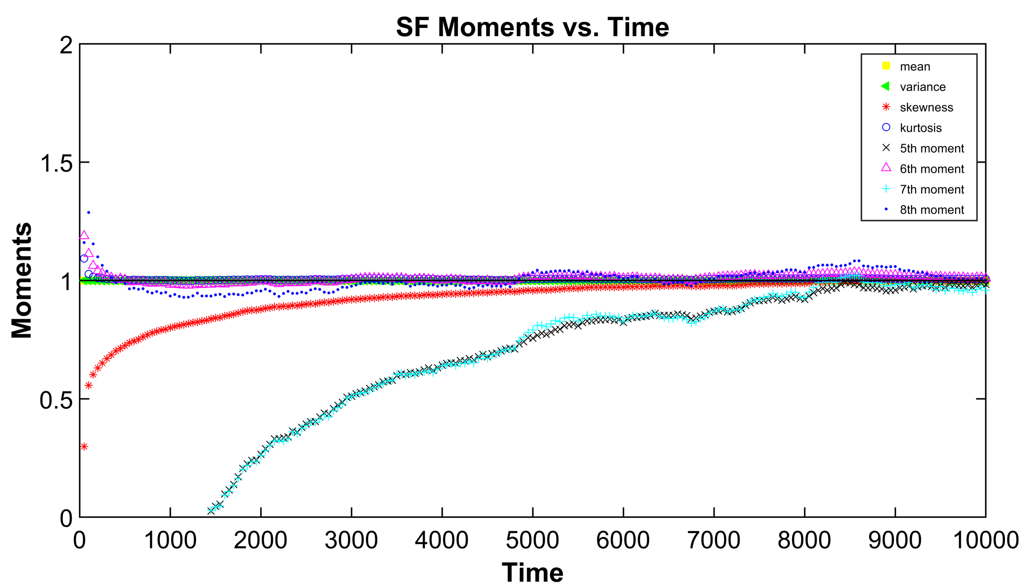


Figure 4.7: Closeness of moments of the distribution of SF to T-W ($\beta = 4$) over time. Exact moment values are normalized using their respective T-W values. Data taken from 10000 runs, with $L = 2^{13}$ and Time = 10000 s.

Again, we needed a long process of 10 000 seconds to clearly see how the distribution of spatial fluctuations is converging to this T-W distribution.

On the last instant of the process, the exact values of moments are as below:

Moments	T-W ($\beta = 4$) dist	Simulation
1st (mean)	0	0
2nd (variance)	1	1
3rd (skewness)	0.165509	0.1653
4th (kurtosis)	3.0492	3.0657
5th moment	1.66795	1.6401
6th moment	16.0106	16.1574
7th moment	17.9278	17.1751
8th moment	123.167	123.2028

Table 4.1: The exact moment values for the spatial fluctuations under the new equation.

All of them are so close to the respective theoretical values for T-W ($\beta = 4$) distribution.

The new dynamics of this equation manifested themselves even in the space domain. The scaling exponent and the distribution of the spatial fluctuations have changed significantly. What about the behaviour in the time domain?

4.2.3 Temporal Fluctuations

Temporal fluctuations defined again by the formula

$$\chi_t(x) \equiv h(x) - \overline{h(x)} \ ,$$

with the overbar ($^-$) representing the moving average, are also shown in the next figure. The variations in time are clear, but are they simply random, as in the Gaussian distribution?

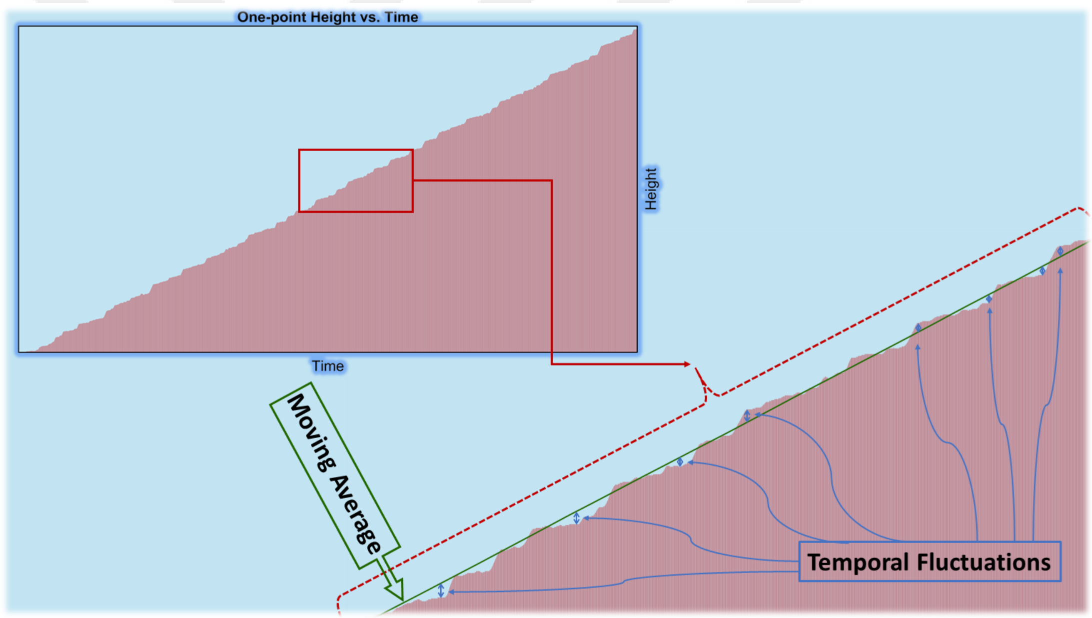


Figure 4.8: Temporal fluctuations in the new dynamics.

1. *Probability Distribution of TF*

So, let us see the distribution of them next:

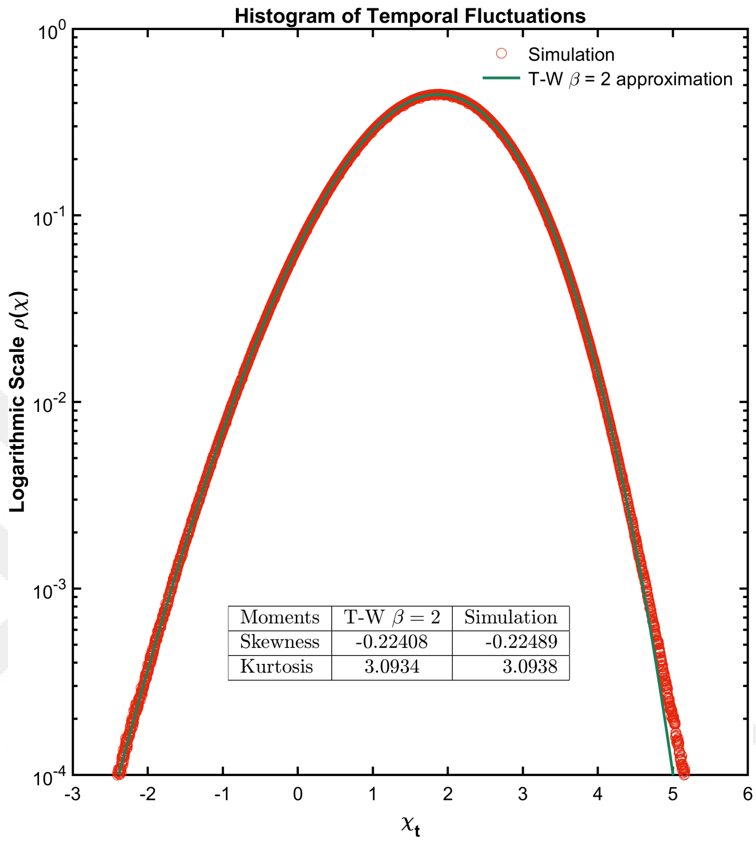


Figure 4.9: The distribution of temporal fluctuations.
Data taken from 100 runs, with $L = 2^{13}$ and Time = 5000 s.

Their distribution is the "mirrored" (negatively skewed) T-W ($\beta = 2$) distribution. This distribution is that of the largest eigenvalues of random matrices in the Gaussian Unitary Ensemble. This T-W GUE distribution is the distribution coming up in the spatial fluctuations of the systems starting with a curved substrate and being in the 1+1 dimensional KPZ universality class. However, the "mirrored" version of this T-W distribution came out in the temporal fluctuations in experiments with systems having curved aggregates growing in time.

Next, let us make the necessary checks to see how stable is this distribution. Finally, we again calculate the first eight moments of the distribution and see how they change with the span.

2. *Skewness and Kurtosis of TF Distribution*

First, let us see the behaviour of only the third and fourth moments, then compare all 8 of them to the respective theoretical values of "mirrored" T-W ($\beta = 2$) distribution.

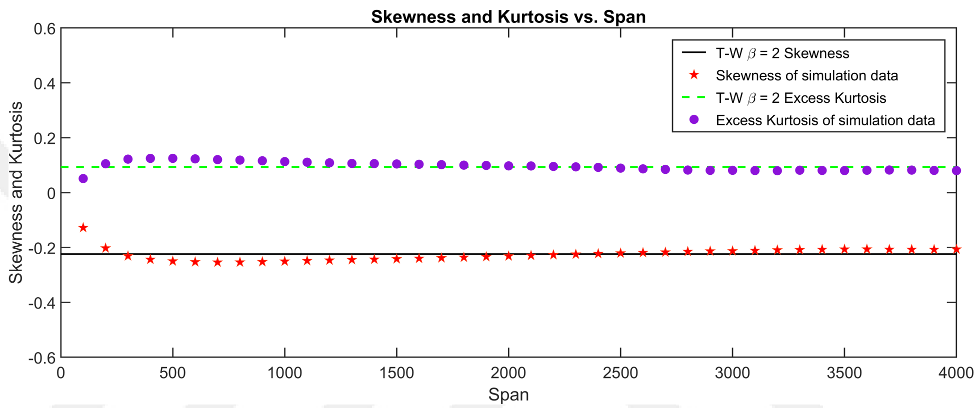


Figure 4.10: Skewness and kurtosis of the distribution of TF. Data taken from 100 runs, with $L = 2^{13}$ and Time = 5000 s.

3. *Moments of TF Distribution*

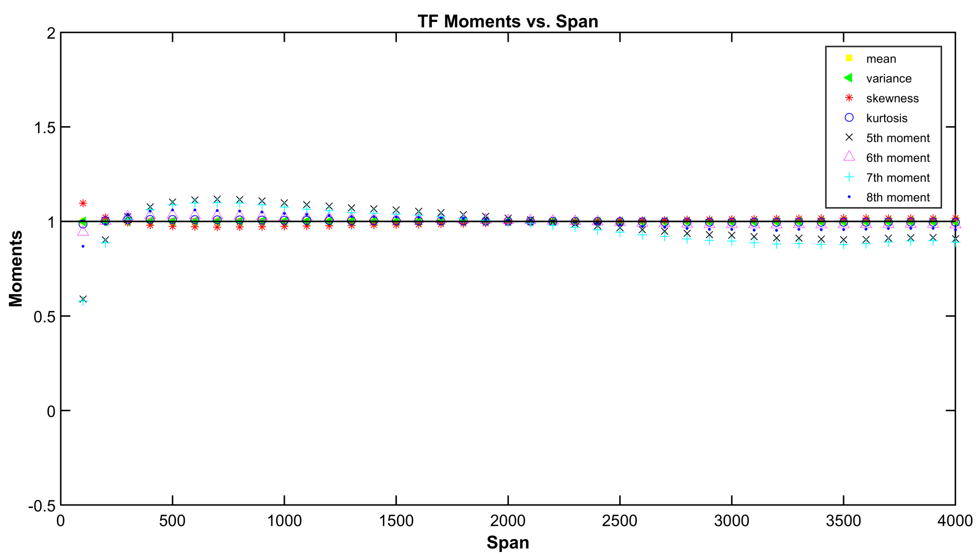


Figure 4.11: Closeness of moments of the distribution of TF to T-W ($\beta = 2$) distribution. Exact moment values are normalized using their respective T-W values. Data taken from 100 runs, with $L = 2^{13}$ and Time = 5000 s.

From these two graphs, there is no suspicion that temporal fluctuations from this new equation are in the T-W distribution.

And, for some particular span value, we have the following exact values of the first eight moments, whose closeness to their respective theoretical values of "mirrored" T-W ($\beta = 2$) distribution is obvious:

Moments	Mirrored T-W ($\beta = 2$) dist	Simulation
1st (mean)	0	0
2nd (variance)	1	1
3rd (skewness)	-0.224084	-0.2249
4th (kurtosis)	3.09345	3.0937
5th moment	-2.27945	-2.2507
6th moment	16.9079	16.9172
7th moment	-25.0513	-24.2562
8th moment	139.552	139.2021

Table 4.2: The exact moment values for the temporal fluctuations under the new equation.

The coincident results between these simulations and that of the experimental work on curved aggregates are unbelievable.

We see that modifying the KPZ equation with two new terms in the time domain produces new and different results from the KPZ equation itself, but most importantly, unique behaviours are shown, making the results very promising.

Chapter 5

Conclusion

Herein, the efforts have been to develop an equation producing growing interfaces that fluctuate in time according to the universal Tracy-Widom distribution. Starting from the experimental evidence in Dissipative Self-Assembly work [49, 4], we were searching for some mathematical way to express the effect of the feedback mechanisms working on the simply complex system of the self-assemblies. This takes us into the great work of Kardar et al. [2], who could make it possible to describe growing interfaces in complex enough systems in a single continuum equation. However, this and other works in the discrete models used for the same purpose of describing complex nonlinear surface growths, having been focused chiefly on the statistical properties of the surface's roughness in the space domain, did not include any interesting behaviour that could happen in the time domain.

However, nature is full of surprises. The statistics of temporal fluctuations were seen to be as universal as the spatial ones, having the same certain probability distribution. That is why we decided to modify the continuum KPZ equation so that, hopefully, we get the Tracy-Widom distribution in the time domain. We first simulated the KPZ equation to understand its dynamics better and to be sure of the methods we would use to analyse the modified equation. All the results we got were consistent with the well-established theory of KPZ universality. The scaling exponent was robustly taking the 1/3 value, as expected, and the spatial height fluctuations' distribution quickly converged to the particular Tracy-Widom statistics expected for the 1+1 dimensional case we were working on. Moreover, we checked the distribution of the height temporal fluctuations, which, to our knowledge, was done for the first time, and again we got the expected non-interesting symmetrical bell-curve of the Gaussian distribution.

The KPZ equation, being all linear in the time domain, was not expected to produce any other than Gaussian temporal fluctuations. Our job was to add some nonlinearity in the time domain so that the dynamics would be "steered" a bit more, and probably we would go out of the attracting basin of the Gaussian universality for the temporal fluctuations. Here, to cut it short, in the end, we had a new equation, a modified version of the KPZ equation, with the terms of it in the space domain left untouched and two additional terms in the time domain, where one of them is nonlinear. Both these terms turned out to be the analogue versions of the diffusion and nonlinear spatial terms.

This new equation could satisfy what we aimed for in the beginning. The height temporal fluctuations of the growing interface were in the universal Tracy-Widom (GUE) distribution, the same as the experimental results in the DSA experiments. We checked all eight first normalized moments of the distribution and saw an exact matching with the respective theoretical values.

The remarkable harmony of the simulation and experimental results, showing both the "mirrored" version of the T-W GUE distribution, is also giving some hints about the nature of the behaviour in the time domain. The distribution of the temporal fluctuations is precisely the same even though the geometries of the systems are different, having a flat substrate in the simulation and a curved aggregate in the experiments. For the respective geometries, it is analytically shown that in the 1+1 dimensional systems, the distributions of the spatial fluctuations are different, particularly Tracy-Widom GOE for the flat geometry and GUE for the curved one.

However, the surprise does not end here. The spatial fluctuations, having a

T-W GOE in the KPZ equation for the system we simulate, now in a longer process asymptotically take an utterly unexpected probability distribution, that of T-W GSE. Furthermore, the roughness of the interface starts to grow with a $\beta = 1/4$ exponent. New behaviours are coming out in every aspect, and it seems that no sign is left of the "old" KPZ equation.

These new two terms have caused the dynamics to be very different from the KPZ dynamics in 1+1 dimensions; however, judging from the results in the interface width, it seems that the behaviour is similar to what is observed from the KPZ equation in 2+1 dimensions, even though this new equation remains in the 1+1 dimensional class.

All these results have given a tremendous amount of evidence on the importance of researching more on what is happening in time on the surface growth phenomena. Also, they give an excellent base for a more rigorous analytical work on the continuum equations describing growth processes in the far-from-equilibrium conditions.

Finally, the remarkable experimental results combined with the numerical results of this thesis are creating more questions about the negatively skewed T-W distributions and the universal properties of T-W GSE statistics. In this way, they seem to open a new curtain in the big scene of Tracy-Widom and random matrix universality.

Bibliography

[1] A.-L. Barabási and H. E. Stanley, *Fractal Concepts in Surface Growth*. Cambridge University Press, 1995.

[2] M. Kardar, G. Parisi, and Y.-C. Zhang, "Dynamic scaling of growing interfaces," *Physical Review Letters*, vol. 56, no. 9, pp. 889 – 892, 1986.

[3] C. A. Tracy and H. Widom, "Level-spacing distributions and the airy kernel," *Communications in Mathematical Physics*, vol. 159, pp. 151–174, Jan 1994.

[4] G. Makey, S. Galioglu, R. Ghaffari, E. D. Engin, G. Yıldırım, Ö. Yavuz, O. Bektaş, Ü. S. Nizam, Ö. Akbulut, Ö. Şahin, K. Güngör, D. Dede, H. V. Demir, F. Ö. Ilday, and S. Ilday, "Universality of dissipative self-assembly from quantum dots to human cells," *Nature Physics*, vol. 16, pp. 795–801, Jul 2020.

[5] P. Meakin, "The growth of rough surfaces and interfaces," *Physics Reports*, vol. 235, no. 4, pp. 189–289, 1993.

[6] T. Halpin-Healy and Y.-C. Zhang, "Kinetic roughening phenomena, stochastic growth, directed polymers and all that. aspects of multidisciplinary statistical mechanics," *Physics Reports*, vol. 254, no. 4, pp. 215–414, 1995.

[7] F. Family and T. Vicsek, "Scaling of the active zone in the eden process on percolation networks and the ballistic deposition model," *Journal of Physics A: Mathematical and General*, vol. 18, pp. L75–L81, Feb 1985.

[8] M. Plischke and Z. Rácz, "Dynamic scaling and the surface structure of eden clusters," *Phys. Rev. A*, vol. 32, pp. 3825–3828, Dec 1985.

[9] S. F. Edwards and D. Wilkinson, "The surface statistics of a granular aggregate," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 381, no. 1780, pp. 17–31, 1982.

[10] F. Family, "Scaling of rough surfaces: effects of surface diffusion," *Journal of Physics A: Mathematical and General*, vol. 19, pp. L441–L446, jun 1986.

[11] K. Moser, J. Kertész, and D. E. Wolf, "Numerical solution of the kardar-parisi-zhang equation in one, two and three dimensions," *Physica A: Statistical Mechanics and its Applications*, vol. 178, no. 2, pp. 215–226, 1991.

[12] V. G. Miranda and F. D. A. Aarão Reis, "Numerical study of the kardar-parisi-zhang equation," *Phys. Rev. E*, vol. 77, p. 031134, Mar 2008.

[13] T. Halpin-Healy, "(2+1)-dimensional directed polymer in a random medium: Scaling phenomena and universal distributions," *Phys. Rev. Lett.*, vol. 109, p. 170602, Oct 2012.

[14] J. Quastel, "Introduction to kpz," *Current developments in mathematics*, vol. 2011, no. 1, 2011.

[15] K. A. Takeuchi, "An appetizer to modern developments on the kardar–parisi–zhang universality class," *Physica A: Statistical Mechanics and its Applications*, vol. 504, pp. 77–105, 2018. Lecture Notes of the 14th International Summer School on Fundamental Problems in Statistical Physics.

[16] I. Corwin, "The kardar–parisi–zhang equation and universality class," *Random matrices: Theory and applications*, vol. 1, no. 01, p. 1130001, 2012.

[17] K. Johansson, "Shape fluctuations and random matrices," *Communications in Mathematical Physics*, vol. 209, pp. 437–476, Feb 2000.

[18] C. A. Tracy and H. Widom, "On orthogonal and symplectic matrix ensembles," *Communications in Mathematical Physics*, vol. 177, pp. 727–754, Apr 1996.

[19] M. Prähofer and H. Spohn, "Universal distributions for growth processes in $1+1$ dimensions and random matrices," *Phys. Rev. Lett.*, vol. 84, pp. 4882–4885, May 2000.

[20] J. Gravner, C. A. Tracy, and H. Widom, “Limit theorems for height fluctuations in a class of discrete space and time growth models,” *Journal of Statistical Physics*, vol. 102, pp. 1085–1132, Mar 2001.

[21] S. N. Majumdar and S. Nechaev, “Exact asymptotic results for the bernoulli matching model of sequence alignment,” *Phys. Rev. E*, vol. 72, p. 020901, Aug 2005.

[22] T. Sasamoto and H. Spohn, “One-dimensional kardar-parisi-zhang equation: An exact solution and its universality,” *Phys. Rev. Lett.*, vol. 104, p. 230602, Jun 2010.

[23] G. Amir, I. Corwin, and J. Quastel, “Probability distribution of the free energy of the continuum directed random polymer in 1+ 1 dimensions,” *Communications on pure and applied mathematics*, vol. 64, no. 4, pp. 466–537, 2011.

[24] P. Calabrese, P. L. Doussal, and A. Rosso, “Free-energy distribution of the directed polymer at high temperature,” *EPL (Europhysics Letters)*, vol. 90, p. 20002, apr 2010.

[25] P. Calabrese and P. Le Doussal, “Exact solution for the kardar-parisi-zhang equation with flat initial conditions,” *Phys. Rev. Lett.*, vol. 106, p. 250603, Jun 2011.

[26] V. Dotsenko, “Bethe ansatz derivation of the tracy-widom distribution for one-dimensional directed polymers,” *EPL (Europhysics Letters)*, vol. 90, p. 20003, apr 2010.

[27] K. A. Takeuchi and M. Sano, “Universal fluctuations of growing interfaces: Evidence in turbulent liquid crystals,” *Phys. Rev. Lett.*, vol. 104, p. 230601, Jun 2010.

[28] A. Kuijlaars, “103Universality,” in *The Oxford Handbook of Random Matrix Theory*, Oxford University Press, 09 2015.

[29] Y. Jiao, T. Lau, H. Hatzikirou, M. Meyer-Hermann, J. C. Corbo, and S. Torquato, "Avian photoreceptor patterns represent a disordered hyperuniform solution to a multiscale packing problem," *Phys. Rev. E*, vol. 89, p. 022721, Feb 2014.

[30] S. Torquato, "Hyperuniform states of matter," *Physics Reports*, vol. 745, pp. 1–95, 2018. Hyperuniform States of Matter.

[31] E. P. Wigner, "On the statistical distribution of the widths and spacings of nuclear resonance levels," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 47, no. 4, p. 790–798, 1951.

[32] N. Rosenzweig and C. E. Porter, """ repulsion of energy levels" in complex atomic spectra," *Physical Review*, vol. 120, no. 5, p. 1698, 1960.

[33] V. V. Flambaum, A. A. Gribakina, G. F. Gribakin, and M. G. Kozlov, "Structure of compound states in the chaotic spectrum of the ce atom: Localization properties, matrix elements, and enhancement of weak perturbations," *Phys. Rev. A*, vol. 50, pp. 267–296, Jul 1994.

[34] M. Krbálek and P. Seba, "The statistical properties of the city transport in cuernavaca (mexico) and random matrix ensembles," *Journal of Physics A: Mathematical and General*, vol. 33, pp. L229–L234, jun 2000.

[35] H. L. Montgomery, "The pair correlation of zeros of the zeta function," in *Proc. Symp. Pure Math*, vol. 24, pp. 181–193, 1973.

[36] R. M. May, "Will a large complex system be stable?," *Nature*, vol. 238, pp. 413–414, Aug 1972.

[37] J. Baik, P. Deift, and K. Johansson, "On the distribution of the length of the longest increasing subsequence of random permutations," *Journal of the American Mathematical Society*, vol. 12, no. 4, pp. 1119–1178, 1999.

[38] S. N. Majumdar and S. Nechaev, "Anisotropic ballistic deposition model with links to the ulam problem and the tracy-widom distribution," *Phys. Rev. E*, vol. 69, p. 011103, Jan 2004.

[39] S. G. Alves, T. J. Oliveira, and S. C. Ferreira, "Universal fluctuations in radial growth models belonging to the KPZ universality class," *EPL (Europhysics Letters)*, vol. 96, p. 48003, nov 2011.

[40] K. A. Takeuchi, "Statistics of circular interface fluctuations in an off-lattice eden model," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, p. P05007, may 2012.

[41] T. J. Oliveira, S. C. Ferreira, and S. G. Alves, "Universal fluctuations in kardar-parisi-zhang growth on one-dimensional flat substrates," *Phys. Rev. E*, vol. 85, p. 010601, Jan 2012.

[42] C. A. Tracy and H. Widom, "Integral formulas for the asymmetric simple exclusion process," *Communications in Mathematical Physics*, vol. 279, pp. 815–844, May 2008.

[43] C. A. Tracy and H. Widom, "A fredholm determinant representation in asep," *Journal of Statistical Physics*, vol. 132, pp. 291–300, Jul 2008.

[44] V. Dotsenko and B. Klumov, "Bethe ansatz solution for one-dimensional directed polymers in random media," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 03, p. P03022, 2010.

[45] J.-P. Bouchaud and M. Potters, "824Financial applications of random matrix theory: a short review," in *The Oxford Handbook of Random Matrix Theory*, Oxford University Press, 09 2015.

[46] R. Sagron and R. Pugatch, "Universal distribution of batch completion times and time-cost tradeoff in a production line with arbitrary buffer size," *European Journal of Operational Research*, vol. 293, no. 3, pp. 980–989, 2021.

[47] B. Nadler, "Nonparametric detection of signals by information theoretic criteria: Performance analysis and an improved estimator," *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2746–2756, 2010.

[48] B. Nadler, F. Penna, and R. Garello, "Performance of eigenvalue-based signal detectors with known and unknown noise level," in *2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, 2011.

[49] S. Ilday, G. Makey, G. B. Akguc, Ö. Yavuz, O. Tokel, I. Pavlov, O. Gülseren, and F. Ö. Ilday, "Rich complex behaviour of self-assembled nanoparticles far from equilibrium," *Nature Communications*, vol. 8, p. 14942, Apr 2017.

[50] Ü. S. Nizam, G. Makey, M. Barbier, S. S. Kahraman, E. Demir, E. E. Shafigh, S. Galioglu, D. Vahabli, S. Hüsnügil, M. H. Güneş, E. Yelesti, and S. Ilday, "Dynamic evolution of hyperuniformity in a driven dissipative colloidal system," *Journal of Physics: Condensed Matter*, vol. 33, p. 304002, Jun 2021.

[51] C. Dasgupta, J. M. Kim, M. Dutta, and S. Das Sarma, "Instability, intermittency, and multiscaling in discrete growth models of kinetic roughening," *Phys. Rev. E*, vol. 55, pp. 2235–2254, Mar 1997.

[52] C. Dasgupta, S. Das Sarma, and J. M. Kim, "Controlled instability and multiscaling in models of epitaxial growth," *Phys. Rev. E*, vol. 54, pp. R4552–R4555, Nov 1996.

Appendix A

Codes

A.1 MATLAB code for simulating the KPZ equation

```
%% The Simulation of KPZ equation in 1-D using Runge-Kutta (4th order) method

%%% Parameters
lambda = 7;              % coefficient of lateral growth (nonlinear) term

nu = 1;                  % coefficient of diffusion (smoothing) term

D = 1;                   % coefficient of noise

c = 0.3;                 % the controllable parameter for restricting the values of the nonlinear term

%%% System's Boundary Conditions
```

```
L = 2^13;            % Length (1-D horizontal substrate)
T_end = 5000;        % Total time of the process (say
   5000 seconds)

%%% Integration elements

dx = 1;              % distance element
dt = 0.01;           % time element / time-step


%%% Run and store data for N times

N = 1000;                % Number of runs (samples)

t_skip = 5*dt;           % time interval we skip while
   storing data (= dt for no skipping)

t_s = T_end/t_skip;   % length of the time vector after
    skipping some intervals

t = T_end/dt;         % length of the total time vector
    (used for temporal fluctuations)

time = t_skip:t_skip:T_end; % time vector

v_0 = 0;              % add initial constant velocity

%% create the folders organized according to the
   parameters
output_folder = ['output/KPZ_lambda_' num2str(lambda) '
   /KPZ_lambda_' num2str(lambda) '_nu_' num2str(nu)];
```

```
files_folder = [num2str(N) '_files_KPZ_lamda=' num2str(lambda) '_nu=' num2str(nu) '_L=2^' num2str(log2(L)) '_T=' num2str(T_end)];
files_folder = fullfile(output_folder,files_folder);
mkdir(files_folder);
%% create a mat-file per sample (this is needed when the amount of data is huge, so that the RAM is not overloaded)
file_names_A = cell(N,1);
for n = 1:N
    file_name_A_single_run = ['A_run=' num2str(n) '.mat'];  % each run gets a unique filename according to their lab's index
    file_name_A_single_run = fullfile(files_folder,file_name_A_single_run);
    file_names_A{n} = string(file_name_A_single_run);
end
%% Run over all samples
tic
disp(['Start processing ' num2str(N) ' samples']);

parfor n = 1:N    % the work is parallelizable since each run (sample) is independent of each other, and this reduces the time significantly

    disp(['Processing the growth of sample: ' num2str(n) ' / ' num2str(N)]);

    %%% Initial conditions
    h = 0 * ones(1,L);           % initial height of the horizontal substrate

    %%% Preallocating the data matrix
```

```
    B = zeros(t_s,L+1,'int16');  % 2-dimensional matrix
        to be filled with the height values for each
       single run
    k = 0;                       % to count not-skipped
        time-steps (the ones used for storing data)

    for j = 1:1:t   % going over each time-step

        noise = randn(1,L); % Guassianly distributed
           random variable used for the noise

        %%% integration (using RK4 method)

        h_temp = [h(end) h h(1)];   % temporary h for
           periodic boundary conditions

        K1 = v_0 + (lambda/2)*(1/c)*(1-exp(-c*((h_temp
           (3:end)- h_temp(1:end-2))/(2*dx)).^2)) + nu
           *(h_temp(1:end-2)+h_temp(3:end)-2*h_temp(2:
           end-1))/(dx^2);

        %%% repeating the similar process for K2, K3
           and K4 (they are all calculated in a single
           line to not load much the workspace)

        h2 = h +(K1/2)*dt;

        h_temp2 = [h2(end) h2 h2(1)];

        K2 = v_0 + (lambda/2)*(1/c)*(1-exp(-c*((h_temp2
           (3:end)- h_temp2(1:end-2))/(2*dx)).^2)) + nu
           *(h_temp2(1:end-2)+h_temp2(3:end)-2*h_temp2
           (2:end-1))/(dx^2);
```

```
        h3 = h +(K2/2)*dt;

        h_temp3 = [h3(end) h3 h3(1)];

        K3 = v_0 + (lambda/2)*(1/c)*(1-exp(-c*((h_temp3
           (3:end)- h_temp3(1:end-2))/(2*dx)).^2)) + nu
           *(h_temp3(1:end-2)+h_temp3(3:end)-2*h_temp3
           (2:end-1))/(dx^2);

        h4 = h + K3*dt;

        h_temp4 = [h4(end) h4 h4(1)];

        K4 = v_0 + (lambda/2)*(1/c)*(1-exp(-c*((h_temp4
           (3:end)- h_temp4(1:end-2))/(2*dx)).^2)) + nu
           *(h_temp4(1:end-2)+h_temp4(3:end)-2*h_temp4
           (2:end-1))/(dx^2);

        %%% calculating h after one time-step

        h = h + (1/6)*dt*(K1 + 2*K2 + 2*K3 + K4) + sqrt
           (D*dt/dx)*noise; % we add the noise in only
           this step because it doesn't make a change
           if it is in each K above

        %%% filling the pre-allocated 2-d matrix to
           store the height values

        if mod(j,t_skip/dt) == 0

            k = k + 1;  % counting time intervals

```

```
            %%% filling the 2-d matrix in each
               unskipped time-step

            B(k,1:L) = h;
            B(k,L+1) = mean(h);

        end

        %%% Displaying the live growth of the interface
            (close this part to save time)

        figure(1)
        subplot(2,1,1)
        plot(h,'-')
        hold on
        plot(ones(1,L).*mean(h),'r-')
        hold off
        title('Interface')
        xlabel('L')
        ylabel('h','fontsize',15)
        % xlim([0 2^9])
        % ylim([0 10])
        drawnow
        pause(0.1)
        subplot(2,1,2)
        bar(h)
        xlabel('L')
        % xlim([0 2^9])
        % ylim([0 10])
        ylabel('h','fontsize',15)

    end

```

```
    %%% write the data on each file (writing on the file without loading it helps a lot in the parallelized works with huge data)
    temp_mat_A = matfile(file_names_A{n},'Writable', true);
    temp_mat_A.data = B;
end
disp(['Done processing ' num2str(N) ' samples']);
toc
```

A.2 MATLAB code for simulating the modified version of KPZ equation

```
%% The Simulation of the Modified KPZ equation in 1-D using Runge-Kutta (4th order) method

%%% Parameters
lambda = 7;      % coefficient of lateral growth (nonlinear) term
lambda_t = 3.4; % coefficient of temporal nonlinear term

nu = 1;              % coefficient of diffusion (smoothing) term
nu_t = 1.3;          % coefficient of temporal diffusion term

D = 1;                % coefficient of noise

c = 0.3;              % the controllable parameter for restricting the values of the nonlinear term
```

```
c_t = 0.3;            % the controllable parameter for
   restricting the values of the temporal NL term

%%% System's Boundary Conditions

L = 2^13;             % Length (1-D horizontal substrate)
T_end = 5000;         % Total time of the process

%%% Integration elements

dx = 1;               % distance element (pixel)
dt = 0.01;            % time element / time-step (seconds
   )


%%% Run and store data for N times

N = 100;              % Number of runs (samples)

t_skip = 5*dt;        % time interval we skip while
   storing data (= dt for no skipping)

t_s = T_end/t_skip;   % length of the time vector after
    skipping some intervals

t = T_end/dt;         % length of the total time vector
    (used for temporal fluctuations)

time = t_skip:t_skip:T_end;  % time vector

v_0 = 0;                     % add initial constant
   velocity

```

```
G=[0 1;0 0];                    % Matrices used in the integration
M=[0 0;0 -1];
R=[0;1];
P=[0 0;0 -1];

%% create the folders organized according to the parameters
output_folder = ['output/Modified_KPZ_lambda_t_' num2str(lambda_t) '/Modified_KPZ_lambda_t_' num2str(lambda_t) '_nu_t_' num2str(nu_t)];
files_folder = [num2str(N) '_files_Modified_KPZ_lamda=' num2str(lambda) '_nu=' num2str(nu) '_lambda_t=' num2str(lambda_t) '_nu_t=' num2str(nu_t) '_L=2^' num2str(log2(L)) '_T=' num2str(T_end)];
files_folder = fullfile(output_folder,files_folder);
mkdir(files_folder);
%% create a mat-file per sample
file_names_A = cell(N,1);
for n = 1:N
    file_name_A_single_run = ['A_run=' num2str(n) '.mat'];  % each run gets a unique filename
    file_name_A_single_run = fullfile(files_folder, file_name_A_single_run);
    file_names_A{n} = string(file_name_A_single_run);
end
%% Run over all samples
tic
disp(['Start processing ' num2str(N) ' samples']);

parfor n = 1:N

```

```
    disp(['Processing the growth of sample: ' num2str(n) ' / ' num2str(N)]);


    %%% Initial conditions
    h = 0 * ones(1,L);         % initial height of the horizontal substrate
    v = 0.01 * ones(1,L);      % initial velocity of the horizontal substrate (first time derivative)
    q = [h ; v];               % 2-by-L matrix with height and first time derivative

    B = zeros(t_s,L+1,'int16');    % 2-dimensional matrix to be filled with the height values for each single run
    k = 0;                             % to count not-skipped time-steps (the ones used for storing data)

    for j = 1:1:t

        noise = randn(1,L); % Guassian noise

        %%% integration

        h_temp = [h(end) h h(1)];    % temporary h for periodic boundary conditions

        K1 = G*q +(1/nu_t)*(M*q +(P*(lambda_t/(2*c_t)))*(1-exp(-c_t*(q.^2))) + R*(v_0+ (lambda/2)*(1/c)*(1-exp(-c*((h_temp(3:end)- h_temp(1:end-2))/(2*dx)).^2)) + nu*(h_temp(1:end-2)+h_temp(3:end)-2*h_temp(2:end-1))/(dx^2)));
```

```
    %%% repeating the similar process for K2, K3 and K4 (they are all calculated in a single line to not load much the workspace)

    q2 = q +(K1/2)*dt;

    h2 = q2(1,:);
    h_temp2 = [h2(end) h2 h2(1)];

    K2 = G*q2 +(1/nu_t)*(M*q2+(P*(lambda_t/(2*c_t)))*(1-exp(-c_t*(q2.^2)))+ R*(v_0+ (lambda/2)*(1/c)*(1-exp(-c*((h_temp2(3:end)- h_temp2(1:end-2))/(2*dx)).^2)) + nu*(h_temp2(1:end-2)+h_temp2(3:end)-2*h_temp2(2:end-1))/(dx^2)));

    q3 = q +(K2/2)*dt;

    h3 = q3(1,:);
    h_temp3 = [h3(end) h3 h3(1)];

    K3 = G*q3 +(1/nu_t)*(M*q3+(P*(lambda_t/(2*c_t)))*(1-exp(-c_t*(q3.^2)))+R*(v_0+ (lambda/2)*(1/c)*(1-exp(-c*((h_temp3(3:end)- h_temp3(1:end-2))/(2*dx)).^2)) + nu*(h_temp3(1:end-2)+h_temp3(3:end)-2*h_temp3(2:end-1))/(dx^2)));

    q4 = q + K3*dt;

    h4 = q4(1,:);
    h_temp4 = [h4(end) h4 h4(1)];
```

```
        K4 = G*q4 +(1/nu_t)*(M*q4+(P*(lambda_t/(2*c_t)))*(1-exp(-c_t*(q4.^2)))+R*(v_0+ (lambda/2)*(1/c)*(1-exp(-c*((h_temp4(3:end)- h_temp4(1:end-2))/(2*dx)).^2)) + nu*(h_temp4(1:end-2)+h_temp4(3:end)-2*h_temp4(2:end-1))/(dx^2)));

        %%% calculating h after one time-step

        q = q + (1/6)*dt*(K1 + 2*K2 + 2*K3 + K4) + sqrt(D*dt)*(1/nu_t)*R*noise;

        h = q(1,:);       % equalizing the first dimension of q to height

        %%% filling the 2-d matrix to store the height values

        if mod(j,t_skip/dt) == 0

            k = k + 1;  % counting time intervals

            %%% filling the 2-d matrix in the unskipped time-step to store the height values

            B(k,1:L) = h;
            B(k,L+1) = mean(h);
        end

        %%% Displaying the live growth of the interface (close this part to save time)

```

```
        figure(1)
            subplot(2,1,1)
            plot(h,'-')
            hold on
            plot(ones(1,L).*mean(h),'r-')
            hold off
            title('Interface')
            xlabel('L')
            ylabel('h','fontsize',15)
            % xlim([0 2^9])
            % ylim([0 10])
            drawnow
            pause(0.1)
            subplot(2,1,2)
            bar(h)
            xlabel('L')
            % xlim([0 2^9])
            % ylim([0 10])
            ylabel('h','fontsize',15)


    end

    temp_mat_A = matfile(file_names_A{n},'Writable',
       true);
    temp_mat_A.data = B;
end
disp(['Done processing ' num2str(N) ' samples']);
toc
```

A.3 MATLAB code for the statistical analysis of the data

```
%% Theoretiacal values (all 8 moments)
TW_mean_B_1 = -1.2065335745820; TW_var_B_1 =
   1.607781034581; TW_skew_B_1 = -0.29346452408;
   TW_kurt_B_1 = 3.1652429384; TW_5th_mom_B_1 =
   -3.03709; TW_6th_mom_B_1 = 18.3825; TW_7th_mom_B_1 =
    -34.6221; TW_8th_mom_B_1 = 167.491;

TW_mean_B_2 = -1.771086807411; TW_var_B_2 =
   0.8131947928329; TW_skew_B_2 = -0.224084203610;
   TW_kurt_B_2 = 3.0934480876; TW_5th_mom_B_2 =
   -2.27945; TW_6th_mom_B_2 = 16.9079; TW_7th_mom_B_2 =
    -25.0513; TW_8th_mom_B_2 = 139.552;

TW_mean_B_4 = -2.306884893241; TW_var_B_4 =
   0.5177237207726; TW_skew_B_4 = -0.16550949435;
   TW_kurt_B_4 = 3.0491951565; TW_5th_mom_B_4 =
   -1.66795; TW_6th_mom_B_4 = 16.0106; TW_7th_mom_B_4 =
    -17.9278; TW_8th_mom_B_4 = 123.167;

Gauss_mean = 0.00; Gauss_var = 1.00; Gauss_skew = 0.00;
    Gauss_kurt = 3.00;

%% Read all the files and store only the data needed in
    the particular analysis

x = randi([1 L],1,100);  % choose 100 random points
   from the interface
% x = L/8:L/8:L;         % choose 8 'periodic' points
   from the interface
% x = L+1;
```

```
A_t = zeros(t_s,N,length(x),'single');    % empty matrix for storing the data for temporal fluctuations analysis

parfor n = 1:N

    temp_mat_A = matfile(file_names_A{n});  % create the MatFile object to read the files

    aaa = temp_mat_A.data;    % for random points, we have to load all the 'data' file
    A_t(:,n,:) = aaa(:,x);    % storing the data

%      A_t(:,n,:) =  temp_mat_A.data(:,x); % for 'periodic' points we can read the file partially and save time
end

A_t = reshape(A_t,[t_s length(x)*N]);  % reshaping the 3-d matrix in a 2-d matrix by combining the number of runs and points in a single dimension

%% Calculating moments of TF as we iterate in a range of values of the span of the moving average

%%% Here we firstly find the moving average (S) for each run for a particular
%%% span value, and find from there the temporal fluctuations (tf) for each run,
%%% and then collect them all in a single vector. We define two 'types' of
%%% temporal fluctuations, the unnormalized ones (simply 'tf') found
```

```matlab
%%% directly from A_t - S, and the normalized ones ('tf_n') found by
%%% dividing the previous ones by their std. We calculate skewness and
%%% kurtosis for each type. Then we calculate all first 8 moments for the
%%% first type ('tf').
%%% Then, we repeat the same process for many different values of span,
%%% selecting from a wide range.

span_step = 100;      % the step we take in the "span coordinate" (in seconds)
span = span_step:span_step:(T_end-10*span_step); % the span vector
span_s = span/t_skip*dt; % the span vector modified according to the skipped time intervals

moments = zeros(8,length(span_s),'single');  % empty matrix to be filled with moments' values for each particular span

moment34 = zeros(2,length(span_s),'single');

tic
parfor s = 1: length(span_s)
    disp(['processing ' num2str(span(s)) ' span']);

    S = single(smoothdata(A_t,'movmean',span_s(s)/dt));      % smoothing the data in the time domain for each run
    %     S = single(smoothdata(A_t,'sgolay',span(s)/dt));
```

```
    tf = A_t(((span_s(s))/2/dt)+1: t_s-(span_s(s)/2/dt),:) - S((span_s(s)/2/dt)+1: t_s-(span_s(s)/2/dt),:);    % temporal fluctuations (TF) calculated for each run and point

    tf = reshape(tf,[1 numel(tf)]); % all TF collected in a single vector

    for sn=1:2
        if sn == 1
            moment34(sn,s)= skewness(tf);          % 3rd and 4th moments calculated directly
        else
            moment34(sn,s)= kurtosis(tf);
        end
    end

    tf =(tf-mean(tf))/std(tf);            % next normalization of the TF (is done so that the moments are found correctly)

    for sn=1:8
        if sn == 1
            moments(sn,s)= moment(tf,1);  % first 8 moments of TF calculated for each particular span
        elseif sn == 2
            moments(sn,s)= moment(tf,2);  % 1st and 2nd moments are always 0 and 1 respectively since data are normalized
        elseif sn == 3
```

```matlab
      moments(sn,s)= moment(tf,3);  % 3rd and 4th moment here will give the same values as the ones above
    elseif sn == 4
      moments(sn,s)= moment(tf,4);
    elseif sn == 5
      moments(sn,s)= moment(tf,5);
    elseif sn == 6
      moments(sn,s)= moment(tf,6);
    elseif sn == 7
      moments(sn,s)= moment(tf,7);
    else
      moments(sn,s)= moment(tf,8);
    end
  end
end
toc

%% Plotting figures

%%% TF Skewness and Kurtosis each vs. Span
f1 = figure('Name','TF Skew & Kurt vs. Span','units','normalized','outerposition',[0 0 1 1]);
  subplot(1,2,1)                         % skewness
    plot(span,moment34(1,:),'r*')    % plotting skewness of TF
    xlim([0 span(end)])
    ylim([-0.5 0.5])
    hold on
    plot(span,zeros(1,length(span)),'y-')          % skewness of Gaussian dist
    plot(span,-TW_skew_B_1*ones(1,length(span)),'g-')   % skewness of TW dist (beta 1)
```

```
        plot(span,-TW_skew_B_2*ones(1,length(span)),'m-')   % skewness of TW dist (beta 2)
        plot(span,-TW_skew_B_4*ones(1,length(span)),'k-')   % skewness of TW dist (beta 4)
        plot(span,TW_skew_B_1*ones(1,length(span)),'g-')  % negative skewness of the 'mirrored' TW dist (beta 1)
        plot(span,TW_skew_B_2*ones(1,length(span)),'m-')  % negative skewness of the 'mirrored' TW dist (beta 2)
        plot(span,TW_skew_B_4*ones(1,length(span)),'k-')  % negative skewness of the 'mirrored' TW dist (beta 4)
        hold off

    title('Skewness vs. Span','FontSize',17)
    xlabel('Span','fontsize',15,'fontweight','bold')
    ylabel('Skewness','fontsize',14,'fontweight','bold')
    legend('Skewness','Gaussian','T-W \beta = 1','T-W \beta = 2','T-W \beta = 4','FontSize',14)

    subplot(1,2,2)                             % kurtosis
        plot(span,moment34(2,:),'bo')  % plotting kurtosis of TF
        xlim([0 span(end)])
        ylim([2.5 3.5])
        hold on
        plot(span,3*ones(1,length(span)),'y-')                 % kurtosis of Gaussian dist
        plot(span,TW_kurt_B_1*ones(1,length(span)),'g--')  % kurtosis of TW dist (beta 1)
```

```
        plot(span,TW_kurt_B_2*ones(1,length(span)),'m--') % kurtosis of TW dist (beta 2)
        plot(span,TW_kurt_B_4*ones(1,length(span)),'k--') % kurtosis of TW dist (beta 4)
        hold off

    title('Kurtosis vs. Span','FontSize',17)
    xlabel('Span','fontsize',15,'fontweight','bold')
    ylabel('Kurtosis','fontsize',14,'fontweight','bold')
    legend('Kurtosis','Gaussian','T-W \beta = 1','T-W \beta = 2','T-W \beta = 4','FontSize',14)

%%% TF Moments vs. Span
f3 = figure('Name','TF moments vs. Span','units','normalized','outerposition',[0 0 1 1]);
    plot(span,(1 + moments(1,:)),'ys')  % plotting each vector of the scaled moments of TF for each span value
        xlim([0 span(end)])
        ylim([0 2.5])
    hold on
    plot(span,moments(2,:),'g<')
    plot(span,(1+ moments(3,:) - TW_skew_B_2),'r*')
    plot(span,(moments(4,:) / TW_kurt_B_2),'bo')
    plot(span,(moments(5,:) / TW_5th_mom_B_2),'kx')
    plot(span,(moments(6,:) / TW_6th_mom_B_2),'m^')
    plot(span,(moments(7,:) / TW_7th_mom_B_2),'c+')
    plot(span,(moments(8,:) / TW_8th_mom_B_2),'b.')
    plot(span, ones(1,length(span)),'-') % the zero vector representing the moments of the Gaussian distribution
    hold off
```

```
title('TF Moments vs. Span','FontSize',18)
xlabel('Span','fontsize',16,'fontweight','bold')
ylabel('Moments','fontsize',15,'fontweight','bold')
legend('mean','variance','skewness','kurtosis','5th moment','6th moment','7th moment','8th moment','FontSize',15)

%% -----------------------------------------------
span = 2000;        % we choose a optimal span value by examining the above graphs
span_s = span/t_skip*dt;

S = single(smoothdata(A_t,'movmean',span_s/dt));    % smoothing the data in the time domain for each run
%      S = single(smoothdata(A_t,'sgolay',spa/dt));

% -------------------------------------------------
%%% height of a single point vs. time
f4 = figure('Name','height vs. Time','units','normalized','outerposition',[0 0 1 1]);
    plot(time,A_t(:,1),'*b','MarkerSize',10)  % only for a single run (here, the first one)
    hold on
    plot(time,S(:,1),'r-','Linewidth',2)    % plot also the smoothed version of it
    hold off

title('One-point Height','fontsize',22)
xlabel('Time','fontsize',22)
ylabel('Height','fontsize',22)
set(gca,'linewidth',1.7,'Fontsize',20)

```

```
%% -----------------------------------------------------
tf_1 = A_t(span_s/2/dt:t_s-span_s/2/dt+1,:) - S(span_s
   /2/dt:t_s-span_s/2/dt+1,:);      % temporal
   fluctuations (TF) of the middle point calculated for
    each run
tf = reshape(tf_1,[1 numel(tf_1)]); % all TF collected
   in a single vector

mom_3 = skewness(tf);    % 3rd and 4th moments
   calculated directly
mom_4 = kurtosis(tf);

tf=(tf-mean(tf))/std(tf);        %  normalizing TF

tf = tf *sqrt(TW_var_B_2);
tf = tf - TW_mean_B_2;

%%% histogram of TF for a particular span value
f6 = figure('Name','TF histogram','units','normalized',
   'outerposition',[0 0 1 1]);
    % PDF from simulation
    subplot(1,2,1)
    %     g = histfit(tf,100,'normal');      % plotting
       the histogram of TF and fitting it to Gaussian
       dist
        g(1) = histogram(tf,'Normalization','pdf');
              % plotting the histogram of TF
        g(1).FaceColor = [0.6 0.6 1];
        g(1).EdgeColor = [0.6 0.6 1];

        hold on
        plottwappx_B2;           % Tracy_widom
           distribution
```

```
        hold off

    % Labeling the histogram
    title('Histogram of Temporal Fluctuations (PDF)','FontSize',18)
    xlabel('\chi_t','fontsize',16,'fontweight','bold')
    ylabel('\rho(\chi)','fontsize',15,'fontweight','bold')
    set(gca,'linewidth',1.5,'Fontsize',13)


    % Logarithmic scale of histogram for simulation
    subplot(1,2,2)          % plotting the histogram and TW approximation in the log scale
    %     semilogy(g(1).XData,g(1).YData,'o','color',[0.6 0.6 1],'markersize',8);  % simulation values
    %     hold on
    %     semilogy(g(2).XData,g(2).YData,'r-','Linewidth',2);     % Gaussian dist values

        semilogy(g(1).BinEdges(1:length(g(1).BinEdges)-1),g(1).Values,'.','color',[0.6 0.6 1],'markersize',8);

        plottwappx_log_B2;          % Tracy_widom distribution logarithmic scale

    % Labeling the histogram logarithmic scale
    title('Histogram of Fluctuations (Log scale)','FontSize',18)
    xlabel('\chi_t','fontsize',16,'fontweight','bold')
```

```
    ylabel('Logarithmic Scale \rho(\chi)','fontsize',15,'fontweight','bold')
    set(gca,'linewidth',1.5,'Fontsize',13)

    % Adding a table in the figure showing moments
    cs =[ '\begin{tabular}{ | l | c | r |} \hline Moments &  T-W Beta = 2 &  Simulation \\ \hline Skewness & ' num2str(TW_skew_B_2) ' & ' num2str(mom_3) ' \\ \hline Kurtosis & ' num2str(TW_kurt_B_2) '  & ' num2str(mom_4) ' \\ \hline\end{tabular}'];
    ha = annotation('textbox',[0.7 0.7 0.2 0.2],'edgecolor','none','position',[0.60 0.15 0.63 0.15], 'Interpreter', 'latex','fontsize',16);
    set(ha, 'String', cs);

%% Read all the files and store only the data needed in the particular analysis

t_big_skip = 50;         % time interval we skip to store data (= dt for no skipping)(seconds)

t_ss = T_end/t_big_skip;   % length of the time vector after skipping some intervals

A_s = zeros(t_ss,L,N,'single');    % empty matrix for storing the data for the spatial fluctuations analysis

tic
parfor n = 1:N
    temp_mat_A = matfile(file_names_A{n});  % create the MatFile object to read the files
```

```
    A_s(:,:,n) =  temp_mat_A.data((t_big_skip/t_skip):(t_big_skip/t_skip):T_end/t_skip,1:L); % for 'periodic' points we can read the file partially and save time
end
toc
A_s = reshape(A_s,[t_ss L*N]);  % reshaping the 3-d matrix in a 2-d matrix by combining the number of runs and points in a single dimension
%% Plotting figures
%%% Spatial Fluctuations (SF)

A_s = A_s'; % 2-dimensional matrix collecting all height values from N runs

time_s = t_big_skip:t_big_skip:T_end;

%%% graph of change of width as time passes(Both x and y axis are logarithmic scale)
f7 = figure('Name','width vs. Time','units','normalized','outerposition',[0 0 1 1]);
    plot(log(time_s),log(std(A_s)),'*b','MarkerSize',10)  % std is found from the data from all the runs for each instant
    hold on
    plot(log(time_s),log(time_s.^(1/3)),'--k')      % the t^1/3 dependance of the width of fluctuations in the KPZ universality class
    hold off

    title('Width','fontsize',22)
    xlabel('Time','fontsize',22)
```

```
    ylabel('Overall Width','fontsize',22)
    set(gca,'linewidth',1.7,'Fontsize',20)
    legend('Standart deviation (h)','t^{1/3}','Location','best')

%%% SF Moments (Skewness and Kurtosis) vs. time
f8 = figure('Name','SF Skew & Kurt vs. Time','units','normalized','outerposition',[0 0 1 1]);
    plot(time_s,skewness(A_s),'ro','MarkerSize',10)   % plotting skewness calculated from data from all runs for each instant
    hold on
    tw_skew= -TW_skew_B_1 *ones(1,length(time_s));  % skewness of TW dist (beta = 1)
    plot(time_s,tw_skew,'g-','linewidth', 2)
    plot(time_s,kurtosis(A_s)-3,'bo','MarkerSize',10) % plotting kurtosis calculated from data from all runs for each instant
    tw_kurt= (TW_kurt_B_1 -3)*ones(1,length(time_s)); % kurtosis of TW dist (beta = 1)
    plot(time_s,tw_kurt,'g--','linewidth', 2)
    hold off

    title('Change of Skewness and Kurtosis over time','fontsize',20)
    ylabel('Skewness and Kurtosis','fontsize',22)
    xlabel('Time','fontsize',22)
    set(gca,'linewidth',1.7,'Fontsize',20)
    legend('Skewness','T-W \beta = 1 Skewness','Kurtosis','T-W \beta = 1 Kurtosis','Location','best')
%% ------------------------------------------------
```

```
sf = (A_s(:,end)-mean(A_s(:,end)))/std(A_s(:,end)); % spatial fluctuations (SF) calculated at the last instant from the values from N runs altogether

    a = zeros(1,8,'single');    % empty vector for the moments of SF in the last instant
    for i=1:8
     a(i) = moment(sf,i);       % calculating first 8 moments of the SF
    end

sf = sf*sqrt(TW_var_B_1);  % shifting and shrinking the all the TF by the same values to compare the distribution with the TW distribution
sf = sf+TW_mean_B_1;    % (no effect on the skewness and kurtosis and higher moments) (multiplication with the std and shifting with the mean of TW dist)

%%% histogram of SF from the last instant of the proces
f9 = figure('Name','SF histogram','units','normalized','outerposition',[0 0 1 1]);
    % PDF from simulation
    subplot(1,2,1)
        h = histogram(sf,'Normalization','pdf');
        h.FaceColor = [0.6 0.6 1];
        h.EdgeColor = [0.6 0.6 1];
        hold on
        plottwappx_B1;          % Tracy_widom distribution
        hold off

        % Labeling the histogram
```

```
        title('Histogram of Spatial Fluctuations (PDF)','FontSize',18)
        legend('Simulation','T-W approximation','FontSize',15)
        legend('boxoff')
        xlabel('\chi_x','fontsize',16,'fontweight','bold')
        ylabel('\rho(\chi)','fontsize',15,'fontweight','bold')
        set(gca,'linewidth',1.5,'Fontsize',13)

    % Logarithmic scale of histogram
    subplot(1,2,2)
        semilogy(h.BinEdges(1:length(h.BinEdges)-1),h.Values,'.','color',[0.6 0.6 1],'markersize',8);

        plottwappx_log_B1;          % Tracy_widom distribution logarithmic scale


        % Labeling the histogram logarithmic scale
        title('Histogram of Fluctuations (Log scale)','FontSize',18)
        legend('Simulation','T-W approximation','FontSize',15)
        legend('boxoff')
        xlabel('\chi_x','fontsize',16,'fontweight','bold')
        ylabel('Logarithmic Scale \rho(\chi)','fontsize',15,'fontweight','bold')
        set(gca,'linewidth',1.5,'Fontsize',13)

```

```
        % Adding a table in the figure showing moments
        cs =[ '\begin{tabular}{ | l | c | r |} \hline Moments &  Theoretical &  Simulation \\ \hline Skewness & 0.29 & ' num2str(a(3)) ' \\ \hline Kurtosis & 3.16  & ' num2str(a(4)) ' \\ \hline\end{tabular}'];
        ha = annotation('textbox',[0.7 0.7 0.2 0.2],'edgecolor','none','position',[0.60 0.15 0.63 0.15], 'Interpreter', 'latex','fontsize',16);
        set(ha, 'String', cs);

%% --------------------------------------------------
%%% Save figures

figures_folder = ['Figures_for_Modified_KPZ_lamda=' num2str(lambda) '_nu=' num2str(nu) '_lambda_t=' num2str(lambda_t) '_nu_t=' num2str(nu_t) '_L=2^' num2str(log2(L)) '_T=' num2str(T_end)];
figures_folder = fullfile(output_folder,figures_folder);
mkdir(figures_folder);

exportgraphics(f1,[figures_folder '/TF_Skew_Kurt_vs_Span.png'],'Resolution',300);
saveas(f1,[figures_folder '/TF_Skew_Kurt_vs_Span.m']);

exportgraphics(f3,[figures_folder '/TF_moments_vs_Span.png'],'Resolution',300);
saveas(f3,[figures_folder '/TF_moments_vs_Span.m']);

exportgraphics(f4,[figures_folder '/height_vs_Time.png'],'Resolution',300);
```

```
saveas(f4,[figures_folder '/height_vs_Time.m']);

exportgraphics(f6,[figures_folder '/TF_histogram.png'],
    'Resolution',300);
saveas(f6,[figures_folder '/TF_histogram.fig']);

exportgraphics(f7,[figures_folder '/width_vs_Time.png'
    ],'Resolution',300);
saveas(f7,[figures_folder '/width_vs_Time.m']);

exportgraphics(f8,[figures_folder '/
    SF_Skew_Kurt_vs_Time.png'],'Resolution',300);
saveas(f8,[figures_folder '/SF_Skew_Kurt_vs_Time.m']);

exportgraphics(f9,[figures_folder '/SF_histogram.png'],
    'Resolution',300);
saveas(f9,[figures_folder '/SF_histogram.fig']);

close all

%% ------------------------------------------------
function plottwappx_B1
% Script to plot the approximated TW distributions
%
% see paper M.Chiani, "Distribution of the largest
    eigenvalue for real
% Wishart and Gaussian random matrices and a simple
    approximation for the
% Tracy-Widom distribution", submitted 2012, ArXiv
%
x = -6:.001:5;
[y1,~] = tracywidom_appx(x,1);

```

```
hold on
plot(x,y1,'color',[0.6, 0, 0],'LineWidth',2)
end

function plottwappx_log_B1

% Script to plot the approximated TW distributions
%
% see paper M.Chiani, "Distribution of the largest
    eigenvalue for real
% Wishart and Gaussian random matrices and a simple
    approximation for the
% Tracy-Widom distribution", submitted 2012, ArXiv
%
x = -6:.001:5;
[y1,~] = tracywidom_appx(x,1);

hold on
semilogy(x,y1,'color',[0.6, 0, 0],'LineWidth',1)
end

function plottwappx_B2
% Script to plot the approximated TW distributions
%
% see paper M.Chiani, "Distribution of the largest
    eigenvalue for real
% Wishart and Gaussian random matrices and a simple
    approximation for the
% Tracy-Widom distribution", submitted 2012, ArXiv
%
x = -5.5:.001:3.5;
[y1,~] = tracywidom_appx(x,2);

```

```
hold on
plot(-x,y1,'color',[0.6, 0, 0],'LineWidth',2)
end

function plottwappx_log_B2

% Script to plot the approximated TW distributions
%
% see paper M.Chiani, "Distribution of the largest
    eigenvalue for real
% Wishart and Gaussian random matrices and a simple
    approximation for the
% Tracy-Widom distribution", submitted 2012, ArXiv
%
x = -5.5:.001:3.5;
[y1,~] = tracywidom_appx(x,2);

hold on
semilogy(-x,y1,'color',[0.6, 0, 0],'LineWidth',1)
end

function [pdftwappx, cdftwappx] = tracywidom_appx(x, i)

%
% [pdftwappx, cdftwappx]=tracywidom_appx(x, i)
%
% SHIFTED GAMMA APPROXIMATION FOR THE TRACY-WIDOM LAWS,
     by M. Chiani, 2012
%
% TW ~ Gamma[k,theta]-alpha
%
% [pdf,cdf]=tracywidom_appx(x,2) gives the approximated
     pdf(x) and CDF(x) of the TW2
```

```
%
% [pdf,cdf]=tracywidom_appx(x,i) for i=1,2,4 gives TW1,
    TW2, TW4
%
% see paper M.Chiani, "Distribution of the largest
   eigenvalue for real
% Wishart and Gaussian random matrices and a simple
   approximation for the
% Tracy-Widom distribution", submitted 2012, ArXiv
%

kappx = [46.44604884387787, 79.6594870666346, 0,
   146.0206131050228];   %  K, THETA, ALPHA
thetaappx = [0.18605402228279347, 0.10103655775856243,
   0, 0.05954454047933292];
alphaappx = [9.848007781128567, 9.819607173436484, 0,
   11.00161520109004];

cdftwappx = cdfgamma(x+alphaappx(i), thetaappx(i),
   kappx(i));   % Chiani's appx: Tracy-Widom as a
   shifted Gamma

pdftwappx = pdfgamma(x+alphaappx(i), thetaappx(i),
   kappx(i));   % Chiani's appx: Tracy-Widom as a
   shifted Gamma

end

function [pdfgamma]=pdfgamma(x, ta, ka) % Utility: PDF
   of a Gamma
if(x > 0)
    pdfgamma=1/(gamma(ka)*ta^ka) * x.^(ka - 1) .* exp(-
       x/ta);
```

```
else
    pdfgamma=0 ;
end
end

function[cdfgamma]=cdfgamma(x, ta, ka)  % Utility: CDF
   of a Gamma
if(x > 0)
    cdfgamma=gammainc(x/ta,ka);
else
    cdfgamma=0;
end

end
```