Equation of Stars and Clouds

Pontus B K Svenson
(tfkps@fy.chalmers.se)

Master's degree thesis
Engineering Physics
Chalmers University of Technology

September 26, 1996

Abstract

In this report various aspects of the *Kardar-Parisi-Zhang* (KPZ) equation are described. After a brief introduction to the study of growth phenomena in physics, some of the numerous applications of this equation are described, and a review of the experimental situation is given. The concept of scaling is introduced, and the proof of the equivalence of the KPZ-equation and the *evaporation-adsorption* model is reviewed in detail. The exponents of the equation are calculated and compared with computer simulations of the model, in full support of the equivalence. Along the way, the concept of *renormalization* is briefly introduced and the Monte Carlo Renormalization Group described, including a computer-calculation of the critical exponents of the $2D$ Ising model.

Contents

1 Introduction

This report will deal with two seemingly unrelated subjects, renormalization (and especially the Monte Carlo Renormalization Group) and growth. Actually, since renormalization can be applied to growth-phenomena, its inclusion here is not so absurd as it may seem. In order to analyse a particular growth-equation, an Ising spin-model will be used. It is in principle possible to study the Hamiltonian of this model with renormalization group-methods. Results regarding the equation itself can also be found using Fourier-space renormalization.

Another motivation for studying growth and renormalization simultaneously is that the concept of scaling is very important for both. Scaling is the assumption that we can predict what will happen to physical observables if we change the scale on which we look at a system. Consider for example a container filled with gas. One way of describing this system is to specify the values of the thermodynamical functions (free energy, temperature, etc) at regularly spaced reference-points within the container. If we change the number of reference-points, N, used, we effectively change the scale we observe the system on. But changing N also means that we must change the free-energy per reference-point f_N — we realise immediately that

$$f_{N_1} = \frac{N_2}{N_1} f_{N_2} = B f_{N_2} \tag{1}$$

must hold. This is our first example of a *scaling relation*: changing the coordinates (from N_1 to N_2 measurement-points) requires that we also change the observables.

One way of thinking about the changes of scale is that we view the world through a camera and use the zoom-in and zoom-out functions. But the world isn't static. Sometimes it may happen that rescaling the coordinates requires that we also rescale the time-coordinate (like using slow-motion or fast-forward on a VCR). Contrary to what one might expect, the required scale-change is often *not* isotropic in time and space. Generally, a rescaling of space by a factor x necessitates a rescaling of time by a factor x^z, where often $z \neq 1$. The dynamic exponent z is our first example of a *critical exponent*; it can also be used to relate correlation lengths and relaxation times via the formula

$$\xi = \tau^{\frac{1}{z}} \tag{2}$$

This report is outlined as follows: Section 2 describes different types of growth and introduces the equations and scaling relations that will be used throughout the report. It also describes applications in other fields of physics of the KPZ-equation. Section 3 discusses renormalization and the Monte Carlo Renormalization Group, while section 4 is the most theoretical part of the report. It describes a growth model, relates

Figure 1: There are three different types of epitaxial growth. (Figure 16.6 from the book by Zangwill.)

this to the KPZ-equation, and calculates the exponents of the latter. It is followed by a description of some experimental work on growth in section 5. Section 6, finally, describes the computer simulations that have been done on the growth model.

2 Growth and the Equation

2.1 Physical descriptions of growth...

One example of the importance of growth is the subject of thin films. It's important to be able to grow thin films of materials in order to both study them further in some sort of microscope and use them in technological devices. Examples of this importance can be found in the August 16, 1996 edition of *Science*. There are a number of different ways that the experimentalists can grow films. If we for simplicity consider a flat surface that is bombarded with atoms that stick upon impact, there are three important classes of growth[16], summarised in figure 1:

- Layer-by-layer or Franck-van der Merwe growth, where the film is grown one layer at a time in a completely orderly fashion. This is the technologically most important method, because it gives the best films. Unfortunately, however, it becomes increasingly difficult to maintain this growth-type as the number of layers are increased. This is a good model for Molecular-Beam-Epitaxy (MBE), which is very much used in industry.
- Unstable or Volmer-Weber growth, where large heaps of atoms are formed before any complete layers are filled. The film will be very uneven. It will be impossible to use the grown surface as support for a thin disc — because of the heaps of atoms, it will slide off

the surface. We call this that the orientation of the surface is changed, from horizontal to slanting. The structure that forms in this type of growth is often fractal, and difficult to describe. One example of a growth model that gives rise to unstable growth is *diffusion-limited-aggregation*, DLA (see [7] and chapter 5 of [11]).

- Self-affine or Stranski-Krastanov growth, which is sort of an intermediate between the other two types. The original orientation of the substrate is preserved, but it will get rougher as time passes. The disc won't slide off the surface, but it won't be horizontal either. This is the type of growth that we'll be concerned with here.

There are many different experimental techniques that can be used to determine which type of growth that is taking place in a particular substrate, and we refer the interested reader to Zangwill[43], especially chapter 16. Some good general reviews of the subject of growth are [8, 16, 11].

2.2 ...and mathematical

The simplest way to describe a growth process is to define a height function $h(\vec{x}, t)$, which describes the height of the growing system at point $\vec{x}$ and time t. We have here assumed that the sample starts growing from a non-empty d-dimensional structure, and that the height-function is one-valued, which means that no overhangs or "holes" in the substrate may form. The article by Marsili et al[25] describes a new, mathematically heavy, approach to growth that dispenses with this condition, but still derives the same equations and gives the same results.

There are two things worth considering for h. First, how do we get an equation for it to study? Second, can we describe what the surface looks like without specifying the entire h-function?

Let's attack the first problem first:

How will the height function develop in time? The simplest model is that in the interval between t and $t + \Delta t$, the interface will grow randomly, $h(\vec{x}, t+\Delta t) = h(\vec{x}, t) + \Delta t \eta(\vec{x}, t)$. We can model this by a simple differential equation

$$\frac{\partial h(\vec{x}, t)}{\partial t} = \eta(\vec{x}, t) \tag{3}$$

where η is a random stochastic variable, most often taken to be Gaußian white noise with covariance

$$\langle \eta(\vec{x}, t)\eta(\vec{x}', t')\rangle = \gamma\delta(t - t')\delta^d(\vec{x} - \vec{x}') \tag{4}$$

This is not a very realistic model, however, since it completely neglects everything that may happen to incoming particles after they have adsorbed to the surface. We must also include the effect of the surface

Figure 2: This figure illustrates why we need a nonlinear term in the equation. (Figure 3 from Halpin-Healy and Zhang, Phys. Rep. 254:215.)

tension, that strives to make the interface as smooth as possible. This gives us the following equation

$$\frac{\partial h(\vec{x},t)}{\partial t} = \nu \nabla^2 h(\vec{x},t) + \eta(\vec{x},t) \tag{5}$$

where ν is a diffusion constant. This is the so-called Edwards-Wilkinson equation, of which we'll talk more later.

We still do not have the most general equation, however. Consider the interface shown in figure 2a. Using the Edwards-Wilkinson equation, the growth will proceed as in figure 2b. The growth will always be "straight up", perpendicular to the $h = 0$-plane. In reality, the growth will tend to be perpendicular to the local surface of the interface, as in figure 2c. How do we include this in our equation? Mathematics comes to the rescue, saying that the projection in the h-direction of this growth is proportional to $\sqrt{1 + (\nabla h)^2}$ (use Pythagora's theorem). Taylor-expanding and removing the constant term (this simply means setting the average height to zero), we finally get the Kardar-Parisi-Zhang (KPZ)-equation

$$\frac{\partial h(\vec{x},t)}{\partial t} = \nu \nabla^2 h(\vec{x},t) + \frac{\lambda}{2}(\nabla h(\vec{x},t))^2 + \eta(\vec{x},t) \tag{6}$$

In dimensionality $1+1$, this is often rewritten in terms of the slope $u = \nabla h$ as

$$\frac{\partial u}{\partial t} = \nu \nabla^2 u + \lambda u \nabla u + \nabla \eta \tag{7}$$

which is known as the noisy Burgers equation in turbulence. The Burgers equation is a simplified model of the Navier-Stokes equation, where one considers fluids without pressure.

We have reached the KPZ-equation by implicitly using many approximations, and some of them may of course be wrong. There could for instance be more non-linear terms in the equation, if we include more terms in the Taylor-expansion above. A generalisation of the KPZ-equation with a general non-linear term $\lambda|\nabla h|^{\zeta}$ has been studied by Krug and Spohn[21], and they have managed to show that this term will be irrelevant if $\zeta \geq 3$, for dimensionality $1+1$. Note also that terms $v_0 + \vec{a} \cdot \nabla h$, corresponding to $\zeta = 0$ and $\zeta = 1$, can be removed by the simple transformation $\tilde{h}(\vec{x}, t) = h(\vec{x} - t\vec{a}, t) - v_0 t$. The KPZ-equation describes growth as a local phenomenon[16]. If there are non-local influences, some other equation must be used.

It is interesting to note that the KPZ-equation doesn't describe just growth. By a simple transformation[22], we can map it to the equation describing a system of directed polymers (or, more generally, any sort of directed paths though a random media). It is useful now to define $Z(\vec{x}, t) = \exp[\frac{\lambda}{2\nu} h(\vec{x}, t)]$ and find an equation for the time-development of Z. It is easy to verify that this substitution results in the transformation of equation (6) into

"This is a one-line proof...if we start sufficiently far to the left"

$$-\frac{\partial Z(\vec{x}, t)}{\partial t} = (-\nu\nabla^2 - \frac{\lambda}{2\nu}\eta(\vec{x}, t))Z \tag{8}$$

Now this is in fact an equation that we can solve! If we imagine for a moment that time is imaginary, $dt = id\tau$, this looks very much like a Schrödinger equation with a stochastic potential $-\eta(\vec{x}, t)$. The solution can be written as a path-integral

$$Z(\vec{x}, t) = \int_{\vec{y}(0)=\vec{0}}^{\vec{y}(t)=\vec{x}} \mathcal{D}[\vec{y}(\cdot)] \exp[-\frac{1}{2\nu}\int_0^t d\tau \frac{\dot{\vec{y}}(\tau)^2}{2}] \exp[-\frac{1}{2\nu}\int_0^t d\tau \lambda \eta(\vec{y}(\tau), \tau)] \tag{9}$$

where the symbol "$\int_{\vec{y}(0)=\vec{0}}^{\vec{y}(t)=\vec{x}} \mathcal{D}[\vec{y}(\cdot)]$" simply means to sum over all possible paths in the t-$\vec{y}$-plane such that the boundary conditions are met. Now let's see if we can find a physical meaning to this equation. The fact that we have used the letter Z for the new function is suggestive — can we interpret the equation as a partition function? The first exponential in the integral is a Brownian motion from the origin to $\vec{x}$ in time t. The second exponential can be seen as a perturbation of the simple motion, caused by a random potential η.

By considering time to be just another space-coordinate we can view the KPZ-equation also as a description of a directed (in the t-direction) chain (or polymer) in a random media of dimensionality $d+1$. 2ν is the temperature, while $h(\vec{x}, t)$ is the free energy. The expression for the average free energy

$$\langle -h(\vec{x}, t)\rangle = -\frac{2\nu}{\lambda}\langle \log Z(\vec{x}, t)\rangle \tag{10}$$

is difficult to evaluate because of the random potential η. This is the same problem that occurs also in the theory of spin-glasses, and methods developed there (e.g. the replica trick) can be used[22]. Yet another application of equation (8) will be described in the section on experiments.

There are also a number of other problems in physics than can be mapped to the KPZ-equation. It is possible to study elasticity effects of two-dimensional so called smectic-A type liquid crystals [12, 17] using the $1+1$-dimensional KPZ-equation. ("Smectic" is the name given to liquid crystals that, apart from the orientational order, also have crystalline order in one dimension.) The general KPZ-equation can also be used to study the dynamics of a Sine-Gordon chain, driven lattice gases, or, via a discrete mapping which will be discussed later on, one-dimensional rings of electrons.

One of the assumptions that led us to the KPZ-equation is that the noise $\eta(\vec{x},t)$ depends on the position $\vec{x}$ over the substrate. This is good if we consider models where material is dropped from directly above, but if we want to have more realistic models, we must let the noise depend also on the height over the substrate, i.e.

$$\frac{\partial h(\vec{x},t)}{\partial t} = \nu\nabla^2 h(\vec{x},t) + \frac{\lambda}{2}(\nabla h(\vec{x},t))^2 + \eta(h,\vec{x},t) \tag{11}$$

This equation is more difficult to analyse. The other approximation of letting η depend only on h and t leads to the Quenched KPZ-equation. A possible physical application for this equation could be growth in a system where there is strong variation of temperature or pressure with altitude.

Now turn to the second problem above: to describe the surface without using the h-function.

2.3 Scaling

To describe the appearance of the surface without actually specifying the entire h-function seems like a dream too wild to be true. But it is in fact possible. Mandelbrot[24] has shown that we can take a structure and calculate *fractal dimensions* that in some way capture the essence of it. A similar thing can be done to describe the type of surfaces that we study here.

Let's look at a function $W(l,t)$ that we will take to represent the "width" of the surface over a length l and at time t. This function could be defined as a correlation function

$$W(l,t) = \langle h(x+l,t) - h(x,l)\rangle_x \tag{12}$$

or as the difference of $\max[h]$ and $\min[h]$ over a length l; the precise definition is immaterial for the reasoning here.

We must now introduce two scaling *assumptions*, that seem to be fulfilled for most growth-mechanisms. The first is that there is a *dynamical exponent* z, that relates rescalings in time and space. The assumption is that if we rescale the linear size of our system by a factor b, we must also rescale *time* by a factor b^z. The meaning of this assumption, then, is that time and space are *not* equal (unless z happens to take the value 1). The main difference between the KPZ and Edwards-Wilkinson equations is that the values of the dynamical exponent are different: 2 for EW and $\frac{3}{2}$ for KPZ.

The second assumption regards what happens to the function W when we do this rescaling. Family and Vicsek[9] were the first to state that this function would then be rescaled with a factor b^α, where α is another exponent, sometimes called the "wandering" or "roughening" exponent. We can use these two exponents α and z to describe the most general features of a surface.

Mathematically, these assumptions imply that

$$W(bl, b^z t) = b^\alpha W(l, t) \tag{13}$$

This can be used to derive many other properties of the function W. For instance, the same identity must hold, in a statistical sense, also for the height-function h itself. (For more details and consequences of this, see the section on simulations below.)

It is the existence of a "roughness" exponent that differentiates self-affine growth from unstable. Equation (13) means that there is an anisotropy in W considered as a function of l for fixed t. If we magnify a picture of a growing interface by a factor b horizontally, we must magnify it by a factor b^α vertically, if we want to get a self-similar picture. This is the defining property of a self-affine curve. In contrast, a fractal curve will look self-similar under an isotropic magnification.

Now we'll stray even farther from the subject of growth and examine the concept of renormalization, and particularly the so-called Monte Carlo Renormalization Group.

3 Renormalization

3.1 The wonderful world of phase-transitions

One of the most interesting problems in physics is that of phase-transitions. Consider a glass of water. If we heat it, it will boil and we get steam. If we instead cool it, it will freeze and turn into ice. These transitions from one state of matter into another are called phase-transitions. There are also many other such transitions in physics, for instance a magnet will become non-magnetic and a superconductor will become an ordinary conductor when the temperature is raised.

α	specific heat	$c(T) \sim (T - T_c)^{-\alpha}$
β	order parameter, no field	$m(T, h = 0) \sim (T - T_c)^{\beta}$
γ	susceptibility	$\chi \sim (T - T_c)^{-\gamma}$
δ	order parameter, small field	$m(T = T_c, h) \sim \vert h\vert^{1/\delta}$
ν	correlation length	$\xi \sim (T - T_c)^{-\nu}$

Table 1: Some critical exponents and their definitions.

In order to describe the physics at the phase-transitions, so called *critical exponents* have been introduced. These are numbers that relate physical quantities to the distance in temperature from the transition. For example, the critical exponent β relates the *order-parameter*, a description of "how much" the system is in a specific ordered phase, to the temperature as $m \sim |\frac{T-T_c}{T_c}|^{\beta}$, where T_c is the transition temperature. Other critical exponents and the physical property they are related to can be found in table 1.

The renormalization group is a very powerful theoretical method to understand the general properties of phase-transitions. It was first introduced by Wilson[41] in the early 1970's and used, among other things, to solve the Kondo-problem. The name comes from the renormalization concept used in quantum-electrodynamics. Wilson exploited the similarities between its use there and the problems in phase-transitions to develop his theory.

One example of the usefulness of the renormalization group is that universality can be understood with the renormalization group, but not with any of the previous theories of phase-transitions.

Universality is one of the perhaps most surprising results in physics. It means that the critical exponents are independent of most of the details of the system under study. Often only the dimensionality and the symmetries of the system are important. Some of the more important universality classes can be seen in table 2. A related fact, equally puzzling to pre-renormalization physicists is that the critical exponents are the same both above and below the transition temperature, provided that they both exist.

"The truth of a proposition has nothing to do with its credibility. And vice versa"
— Lazarus Long

What is renormalization, then? A complete description would take too much space, for more information the reader is instead referred to one of the many books on it, for example [4, 31, 42].

The key idea behind renormalization is to realise that there are several diffent length-scales involved in a problem, and that *not all of them are of equal importance.*

Consider water on the verge of freezing to ice. How does the mixture look like? One might expect that the transition happens simultaneously everywhere, but this is not the case. Instead, there are bubbles of *all sizes* of ice everywhere in the water. What will happen if we magnify the areas that look like they contain only water? We will again get

Dimension	Rank of order parameter	Model	Example	Order parameter
2	1	2D Ising	Adsorbed films	Surface density
2	2	2D XY	He^4-films	Superfluid phase-amplitude
2	3	2D Heisenberg		Magnetisation
>2	∞	Spherical		
3	0	Self-avoiding random walk	Collapse of polymers	Density of chain-ends
3	1	3D Ising	Uniaxial ferromagnet	Magnetisation
			Fluid near critical point	Density
			Liquid-mixture	Concentration
			Alloy near order-disorder transition	Concentration
3	2	3D XY	Planar ferromagnet	Magnetisation
			He^4 near superfluid transition	Superfluid phase-amplitude
3	3	3D Heisenberg	Isotropic ferromagnet	Magnetisation
≤ 4	-2	"very theoretic"		
4	32	Quantum chromodynamics	Quarks	

Table 2: Some different universality classes. (Adapted from Wilson, Scientific American, August 1979.)

a mixture of ice and water, that will look somewhat like the original one. If we continue this process, and are at the exact transition temperature, we will get more and more images, each containing bubbles of ice in water, and all looking alike. (It is important that we are *at* the freezing temperature, otherwise either ice or water will eventually dominate.)

We thus see that the physics of our problem is essentially the same regardless of what length-scale we use to look at it. (Most books use the Ising spin-up/spin-down model, which is perhaps easier to visualise, to argue for this. One way of removing the small length-scales is then to form block-spins of neighbouring spins and replace the original lattice with a smaller one. If we change the Hamiltonian of the system in a suitable way, the new, smaller system will have the same physical characteristics as the original one.) It is unfortunate that there is no simple experiment that enables us to see this invariance directly. Studying a container of water in the freezer, for instance, won't work.

Now, if the physics is irrelevant of the length-scale we use, couldn't we just chose the "most appropriate" length-scale and see if we understand that system? In a way, this is exactly what the renormalization group is all about. We'll simply take the original Hamiltonian of the system and construct a procedure that removes the short length-scales from the problem. It is as if we took a step back and looked at our system from a greater distance. If we imagine that we have some sort of abstract space of systems, repeated application of this procedure will cause the system to flow towards a fixed-point that describes the phase-transition, if we are at the transition temperature. If we are either above or below the transition temperature, as we must be in all calculations, the system will also be drawn towards this fixed-point, but it will never reach it completely. Instead it will make a turn close to the critical point, and eventually reach another fixed-point that describes not the transition but one of the phases.

3.2 Mathematics

The simplest way of describing the abstract space that the system travels in is to express the Hamiltonian as a sum of all possible interactions in the lattice,

$$H(\{\sigma_i\}) = \sum_{\alpha} K_\alpha T_\alpha(\{\sigma_i\}) \tag{14}$$

where T_α represent arbitrary functions of the spin-configuration $\{\sigma_i\}$ and K_α is the appropriate coupling constant. A point in the space is then represented by the coordinates $\{K_\alpha\}$, and the renormalization will cause the couplings to change. The renormalization can be seen as a transformation R that takes the coupling-constants of generation $n-1$ into those of generation n, $K_\alpha^n = R_\alpha(\{K_\beta^{n-1}\})$.

It is a remarkable fact that the critical exponents turn out to be decided by the way that the flow turns about the critical point. More formally, for so called real-space renormalization-group methods, the critical exponents are the “logarithms of the eigenvalues of the Jacobian of the renormalization transformation”. (The Jacobian of the transformation is often called the critical matrix, because is determines the critical exponents.) (The reader is again referred to the references for equations and more details.) An interesting property of the critical exponents is that they are not all unrelated – there are relations among them that must be satisfied. One example of this is the Rushbrooke equality: $\alpha + 2\beta + \gamma = 2$. This means that it is only necessary to obtain a few of the eigenvalues of the critical matrix, which simplifies the calculations. The experimentalists are also helped by this, since they don't have to do separate measurements for all the exponents.

“Anyone who cannot cope with mathematics is not fully human. At best he is a tolerable subhuman who has learned to wear shoes, bathe and not make messes in the house.”
— Lazarus Long

The presence of this sort of flow in the renormalization group raises the question of whether there can also be limit cycles or other chaotic flows in parameter-space. In two dimensions, Zamolodchikov's *c-theorem* prevents this for hermitean Hamiltonians, but there are no known generalisations of this for other dimensionalities. Such flows have been found in strongly disordered systems[40] and also in frustrated hierarchical models[2], but most of the work (e.g. [6, 5, 38]) that has been done in this area treats the consequences of such flows rather than their existence.

Without thinking too deeply about the subtleties of the problem, an intuitive argument can be given for why such flows do not occur. The exponents are related to the eigenvalues λ_i of the critical matrix by the quantities $y_i = \frac{\log \lambda_i}{\log b}$, where b is the scale-factor by which the linear size of our system has shrunk. If we apply the renormalization transformation twice, both the scale-factor and the eigenvalues λ_i will be squared. This still gives us the same exponents, since $\log a^c = c \log a$. The same applies if we construct a transformation with a scale factor b^x.

“what's so amazing about really deep thoughts”
— Tori Amos

Consider now a system with a limit cycle. This means that some of the eigenvalues will be complex numbers (actually pairs of complex-conjugated numbers, since the matrix is real). Assume that $\lambda_{i,j} = \alpha \pm i\beta$ are such a pair, and let x be such that $\lambda_{i,j}^x$ is real (i.e. $x = \frac{\pm n\pi}{\arg \lambda}$, where n is any integer). The major assumption is now that we can construct a renormalization transformation with scale factor b^x. This changes the eigenvalues to be real, and we see that we don't get the same critical exponents as before. So, we conclude that there can't be limit cycles.

There are several flaws in this argument. First, it might not always be possible to construct a transformation with scale-factor b^x (even if there are infinitely many x's to choose from). If we consider a simple-cubic lattice in d dimensions, it is easy to see that we can construct renormalizations with scale-factors

$$a\sqrt{n_1^2 + n_2^2 + \cdots} = a\sqrt{\sum_{j=1}^{d} n_j^2} \tag{15}$$

where a is the lattice-constant and n_j are integers. (This is simply all the possible distances between lattice-points.) If we call this set of numbers $\mathcal{S}$, the problem is to find an integer n such that

$$b^{\frac{\pm n\pi}{\arg\lambda}} \in \mathcal{S} \tag{16}$$

where $\lambda \in \mathcal{R}$ and $b \in \mathcal{S}$ are given parameters of the problem. Even in one dimension, equation (16) is very difficult to solve.

Another objection to the argument is that we have been very naïve as to what would happen with the connection between the exponents and the eigenvalues. Working through the standard derivation of the scaling for the free energy with complex eigenvalues gives for instance

$$(\frac{1}{t^2} + \beta^2)^{-\frac{d}{2y_1(\alpha)}} \tag{17}$$

instead of the usual $t^{\frac{d}{y_1}}$. Also, strange dependencies on non-physical coupling-constants emerge. More detailed considerations of the scaling-relations in the presence of limit cycles can be found in [5, 38].

3.3 Numerical difficulties

A problem with the renormalization group is that it is often difficult to perform a complete theoretical analysis of a system with it. Some sort of approximations must be introduced. For instance, the renormalization-flow equations are often truncated so that only a few coupling constants are kept.

Another possibility is to use some sort of simulation method to obtain the critical exponents. Let's look at how we can determine the exponents experimentally. Consider some measurement of the specific heat of a material. Since we know that the relationship should be of the form $c \sim |T - T_c|^{-\alpha}$, we can determine the exponent α simply by plotting the data in a log-log-diagram.

Is there some way we can obtain a similar graph without actually performing an experiment? This is in principle very simple. We just define a large enough model of our system in a computer and calculate the property we want to measure for all possible configurations of it. Our problem now is of a computational nature. If we consider a system of size N, where N is "large enough", and for simplicity assume that each of the N components of the system can have only the values $+1$ or -1, we must consider 2^N configurations.

This takes a very, very long time.

We are very fortunate to have a much simpler method: Monte Carlo simulation. We'll visit just a few of the possible configurations, and then average the quantity we want to measure *without* taking the configurations' Boltzman-weights into account.

Does this really work? The answer is yes, provided that we choose the configurations we visit correctly. We start with a configuration that is in some way "typical", and then see what happens if we do some random change of the configuration. If the total energy of the system is lowered by this change, the new configuration is accepted and we do whatever calculations we want with it. If the energy isn't lowered, we accept the change with a probability $e^{-\frac{\delta E}{k_B T}}$, where δE is the change in energy[1]. The Boltzman-weights are thus considered implicitly in the algorithm. Instead of multiplying the measurements with the Boltzman-weights and then summing over all states, we let the weights determine *which states the sum runs over*.

Accepting and rejecting configurations with this algorithm will take care of two necessary things:

- Given an arbitrary starting configuration, it should always be *possible* to reach any other micro-state. The process by which we choose a new trial configuration must not discriminate against some configurations;

- The principle of detailed balance must hold. This says simply that the probability of jumping from a state *a* to a state *b* must be related to the probability of the opposite jump: their ratio must be equal to the ratio of the states' Boltzman-factors.

What the algorithm does is very simple to understand: it drives the system to lower energies, but will permit the system to raise its energy by a certain amount if this amount is small compared to the temperature.

Given a temperature, we are now able to obtain any quantity we want from the system. Getting the critical exponents is then simply a matter of doing measurements for several temperatures, and plotting them in a log-log-diagram. This is still not a very good method, though. To get good accuracy, we need to have information on a lot of temperatures, which means that we will have to do the Monte Carlo simulation several times. Furthermore, the closer we get to T_c, the longer it takes to calculate the observables. This is called critical slowing down; a good discussion of it can be found in [4].

Isn't there any better way?

[1]This is the Metropolis algorithm. In general, Monte Carlo simulations can use other algorithms also[3, 31].

3.4 Monte Carlo Renormalization Group

One method that allows us to find the critical matrix of a system without much effort is the Monte Carlo Renormalization Group (MCRG). This method was invented by Swendsen[37] in the late 1970's, and has been further refined by him and by Ma. We will describe here only the most simple application of it.

Consider a lattice of size $N \times N$ with a Hamiltonian H. Just as before, we will do Monte Carlo simulations on this system, but this time we won't measure various quantities and then try to fit exponents to our data. Instead, we will magically measure the critical matrix directly.

In order to do this, we need two things. First, we must decide on a blocking-scheme, and second, we must find out exactly what terms will show up in the Hamiltonian under the renormalization transformation.

Let us write the Hamiltonian as

$$H(\{\sigma_i\}) = -h\sum_i \sigma_i - J\sum_{<ij>} \sigma_i\sigma_j + \ldots = \sum_\alpha K_\alpha T_\alpha(\{\sigma_i\}) \tag{18}$$

The functions T_α here represent all the different interactions that may show up in the renormalization process, while the "constants" K_α are their respective strengths, and will change under renormalization. Ignoring the magnetic field, the initial value is that only one K_α is non-zero, and this is the only term in the Hamiltonian that we will have to worry about when we calculate the spin-flip probabilities in the simulation. (This is very important in order to get an efficient Monte Carlo algorithm below.)

Consider the correlation function defined by

$$S^n_{\alpha\beta} = \langle T^n_\alpha T^n_\beta\rangle - \langle T^n_\alpha\rangle\langle T^n_\beta\rangle \tag{19}$$

The n here marks in which generation of block-spins the correlation function is measured. Using the formula $\langle T^n_\alpha\rangle = \frac{\partial \log Z}{\partial K^n_\alpha}$ equation (19) can obviously be written as

$$S^n_{\alpha\beta} = \frac{\partial\langle T^n_\alpha\rangle}{\partial K^n_\beta} \tag{20}$$

We can also define a correlation function between spins of different generations

$$S^{n,n-1}_{\alpha\beta} = \frac{\partial\langle T^n_\alpha\rangle}{\partial K^{n-1}_\beta} \tag{21}$$

But, using the chain-rule of differentiation, this can also be written as

$$\sum_\gamma \frac{\partial\langle T^n_\alpha\rangle}{\partial K^n_\gamma}\frac{\partial K^n_\gamma}{\partial K^{n-1}_\beta} = \sum_\gamma S^n_{\alpha\gamma}M^n_{\gamma\beta} \tag{22}$$

where $M^n_{\gamma\beta}$ is exactly the critical matrix that defines the critical exponents. So we can get the critical exponents by measuring certain correlation functions for sufficiently large n, and then solving a matrix equation.

The only problem left now is how to measure the correlation $S^{n,n-1}$ between different generations of spins. For this we need to say a little more about something that was first mentioned on page 10. The renormalization must change the Hamiltonian so that the physics isn't changed. More formally, this implies that there must exist a *projection operator* P^n_{n-1} which relates the Hamiltonians of generations n and $n-1$

$$e^{H^n} = \sum_{\{\sigma^{n-1}\}} P^n_{n-1} e^{H^{n-1}} \tag{23}$$

The sum here is a sum over all possible spin-configurations in the parent-generation $n-1$ — the projection operator represents the particular blocking used. The identity means that the operators on the left and right hand sides give the same result for any configuration of spins in the child-generation n.

Return now to equation (21) and write this as

$$S^{n,n-1}_{\alpha\beta} = \frac{\partial}{\partial K^{n-1}_\beta} \frac{\sum_{\{\sigma^n\}} T^n_\alpha e^{H^n}}{\sum_{\{\sigma^n\}} e^{H^n}} = \frac{\partial}{\partial K^{n-1}_\beta} \frac{\sum_{\{\sigma^n\}} T^n_\alpha \sum_{\{\sigma^{n-1}\}} P^n_{n-1} e^{H^{n-1}}}{\sum_{\{\sigma^n\}} \sum_{\{\sigma^{n-1}\}} P^n_{n-1} e^{H^{n-1}}} \tag{24}$$

The rest is now straight-forward, differentiation of a quotient gives

$$\frac{\sum_{\{\sigma^n\}} T^n_\alpha \sum_{\{\sigma^{n-1}\}} T^{n-1}_\beta P^n_{n-1} e^{H^{n-1}}}{\sum_{\{\sigma^n\}} \sum_{\{\sigma^{n-1}\}} P^n_{n-1} e^{H^{n-1}}} - \frac{\sum_{\{\sigma^n\}} T^n_\alpha e^{H^n}}{\sum_{\{\sigma^n\}} e^{H^n}} \times \\ \times \frac{\sum_{\{\sigma^n\}} \sum_{\{\sigma^{n-1}\}} T^{n-1}_\beta P^n_{n-1} e^{H^{n-1}}}{\sum_{\{\sigma^n\}} \sum_{\{\sigma^{n-1}\}} P^n_{n-1} e^{H^{n-1}}} \tag{25}$$

where we have again used equation (23) to remove the projection operator from the first factor of the second term. If we now just remove the "empty" sums over the spins in generation n from the last factor we finally get, as we should,

$$S^{n,n-1}_{\alpha\beta} = \langle T^n_\alpha T^{n-1}_\beta \rangle - \langle T^n_\alpha \rangle \langle T^{n-1}_\beta \rangle \tag{26}$$

All we need to do now is to measure the matrices defined by equations (19) and (26), and then solve equation (22) for the critical matrix.

The bad side of this algorithm is that we must measure many correlation functions in order to get good results, and this costs in terms of computer time. Some simplifications are obvious, however. For instance, correlation functions that mix functions that are odd and even

	α	β	γ	δ	ν
Exact values	0	0.125	1.75	15	1
Simulation	-0.046	0.1296	1.787	14.79	1.023

Table 3: Critical exponents of 2D Ising model obtained with MCRG simulations with one odd and one even interaction.

under the interchange of up- and down-spins will all vanish. It therefore suffices to measure $S^n_{\alpha\beta}$ and $S^{n,n-1}_{\alpha\beta}$ where T_α and T_β are of different parity.

The method turns out to be sensitive to a number of things. We can't start with a random configuration of spins, but must first equilibrate the lattice by doing many Monte Carlo sweeps through it without measuring anything. It is also important to start with a Hamiltonian that is reasonably close to the fixed-point, or else the linearisation won't work. This is not much of a problem, however, since one can simply try different starting Hamiltonians and compare the generated critical matrices. If we started sufficiently close to the fixed point, the critical matrix will look almost the same regardless of how many blockings that have been done.

An example of a program for MCRG can be found in appendix A, and some results for the 2D Ising model are given in table 3. The exact values are from [31], while the computer-calculated ones are the same as those calculated by Swendsen[37].

4 A discrete model for the KPZ-equation

4.1 Introduction

There are several reasons for wanting to use discrete growth models instead of the full KPZ-equation (6). Apart from making it easier to visualise the growth-process, such models are also easier to simulate on a computer. It is of course important that the model we use represents the same physics (is in the same universality class) as the KPZ-equation. In general, this is often difficult to prove analytically. Descriptions of some of the many models that have been introduced can be found in [16, 11].

The model we will use here is sometimes called the evaporation-adsorption model. One of the first descriptions of it is [32]. In order to describe the model, we'll look at a chain of length N, where each node x_i is assumed to have an integer-valued height $h(x_i)$. We'll also take the constraint that all steps in h are either $+1$ or -1, and that there are periodic boundary conditions, i.e. $h(x_{i+N}) = h(x_i)$.

Also define the steps between the nodes of the chain, σ_i, such that the heights are given by $h(x_{i+1}) = h(x_i) + \frac{\sigma_i+\sigma_{i+1}}{2}$. This is the same as saying that σ_i is one if h is increasing over site i and minus one if h is decreasing. If h is constant, σ_i will be the opposite of σ_{i-1}. (This is a somewhat strange definition of σ_i — the reason for it is that we want a single-step model; we don't want to allow steps $\sigma_i = 0$.)

The growth model is now simply to find local maxima of h and (with a certain probability) turn them into minima, and vice versa. Physically, this corresponds to letting evaporation take place at maxima, and adsorption at minima.

One advantage of the definition of σ_i above is that with it not all action is limited to extreme points of h. A flat surface will also see some evaporation/adsorption, since the σ_i's consider it to be a sequence of local maxima and minima.

Mathematically, we can write this rule as

$$(\sigma_j(t+dt), \sigma_{j+1}(t+dt)) = (\sigma_{j+1}(t), \sigma_j(t)) \tag{27}$$

with a probability

$$1 - \frac{1+\epsilon\sigma_j(t)}{2} \tag{28}$$

and unchanged otherwise and use independent updating to evolve our chain in time.

ϵ is here a parameter that will eventually be related to the parameter λ in the KPZ-equation (6). Physically, it corresponds to the difference between the rates of adsorption and evaporation: $\epsilon = 1$ means that only evaporation occurs, while for $\epsilon = 0$ the rates are equal.

The reader may wonder now why we've bothered to introduce the σ_i variables at all, since the above equations could of course just as well be expressed in h. The reason is that we can relate this model of the KPZ-equation to a quantum mechanical Hamiltonian, which uses Ising spins. The Hamiltonian is

$$\begin{aligned} H = -\sum_{j=1}^{N} \sigma_j^x\sigma_{j+1}^x + \sigma_j^y\sigma_{j+1}^y + \sigma_j^z\sigma_{j+1}^z - \frac{1}{4} + i\epsilon(\sigma_j^x\sigma_{j+1}^y - \sigma_j^y\sigma_{j+1}^x) = \\ -\sum_{j=1}^{N} \vec{\sigma}_j \cdot \vec{\sigma}_{j+1} - \frac{1}{4} + i\epsilon\vec{e}_z \cdot (\vec{\sigma}_j \times \vec{\sigma}_{j+1}) \end{aligned} \tag{29}$$

where σ^x is the normal spin-operator, $\vec{\sigma}$ is a vector of these operators, and $\vec{e}_z$ is a normal-vector in the z-direction.

In the literature, there is some ambiguity regarding the exact appearance of this equation. This is due to the fact that some authors choose to work with spins with magnitude $\frac{1}{2}$ while others use magnitude 1, and is completely immaterial for all results. Just as unimportant is the ambiguity regarding the sign of ϵ. Setting $\epsilon \to -\epsilon$ only switches between evaporation and adsorption: it is the same transformation as is obtained by letting $(h, \lambda) \to (-h, -\lambda)$ in the KPZ-equation (6).

We can show that the Hamiltonian (29) and the KPZ-model (27) are the same by just calculating the probability of switching two spins using (29). It is given by $\langle\sigma|\exp -tH|\sigma(j, j+1)\rangle \approx \langle\sigma|1 - tH|\sigma(j, j+1)\rangle$, where $\sigma(j, j+1)$ means the same state as σ but with spins j and $j+1$ switched. We get exactly the same probabilities as above. The easiest way to see this is to use the step-operators $\sigma^{\pm}$ instead of $\sigma^{x,y}$ and write the Hamiltonian as

$$-\sum_{j=1}^{N} \sigma_j^z \sigma_{j+1}^z - \frac{1}{4} + \frac{\sigma_j^+ \sigma_{j+1}^- + \sigma_j^- \sigma_{j+1}^+}{2} + \epsilon \frac{\sigma_j^+ \sigma_{j+1}^- - \sigma_j^- \sigma_{j+1}^+}{2} \tag{30}$$

We can now see by inspection that the expression $\langle\sigma|1 - tH|\sigma(j, j+1)\rangle$ gives the same results as equation (27) above.

The spin-model we have introduced can also be used to model a ring of conducting electrons. If we consider up-spins to be particles, the rule is that particles hop to the right with a probability $\frac{1-\epsilon}{2}$ and to the left with probability $\frac{1+\epsilon}{2}$. By changing the value of ϵ we can thus study the effect of introducing anisotropies in conductive rings.

4.2 Renormalization again

In section 3, we talked almost exclusively about the Monte Carlo renormalization group, and it is difficult to imagine how that would help us understand the KPZ-equation. Where are the block-spins?

But there are other ways to use renormalization. A very general way of solving a partial differential equation is to transform it into Fourier-space, and express the original solution as a Fourier-integral (or -series). This we can of course do with the KPZ-equation (6), but the integrals that arise are not very nice to deal with. The Fourier coefficients of the height function can be expressed as[22]

$$\begin{aligned} \hat{h}(k,t) = e^{-2\nu k^2 t}\hat{h}_0(k) + \int_0^t ds (e^{-\nu k^2 (t-s)} \times \\ \times [\hat{\eta}(k,s) - \frac{\lambda}{2(2\pi)^d} \int d^d q q \cdot (k-q) \hat{h}(k-q,s)\hat{h}(q,s)]) \end{aligned} \tag{31}$$

which looks rather messy.

The only way to solve this expression is to do it perturbatively. Remember now the key idea behind renormalization: *not all length-scales are of equal importance.* One way to get rid of the small length-scales in this equation is to integrate out the modes that have large k-values. This will then give us a new equation, somewhat similar to equation (31). If we manage to find a transformation of the parameters of the old equation into those of the new one, the renormalization is complete.

After a bit of mathematics[22], this transformation takes the form

$$\frac{\partial \lambda}{\partial l} = \frac{2-d}{2}\lambda + K\frac{2d-3}{4d}\lambda^3 \tag{32}$$

where K is a constant. We see that the equation has a fixed point $\lambda = 0$. Linear stability analysis tells us that this fixed point is repulsive for $d < 2$ but gets attractive as the dimensionality of the substrate is increased. This is important. It means that for a one-dimensional substrate, the non-linearity in the KPZ-equation will be important for *all* non-zero λ. In higher dimensions, on the other hand, for small λ's the non-linear term won't be important and the behaviour of the system will be governed by the Edwards-Wilkinson equation instead of the KPZ-equation.

Since the dynamical exponents of these two equations are different, we can take any physical system, do some measurements on it, and then determine which universality class it belongs to, even if we can't understand the system analytically.

4.3 What about the exponents?

One way of determining the values of the exponents for the KPZ-equation (6) is to use the mapping to paths in a random media, equation (8)[22]. This equation is then analysed with renormalization group methods. An easier way of obtaining these is however to use the theory of (the equivalent) Burgers equation (7), particular to dimension $1+1$, and a general relation between α and z which we will prove first.

Consider the rescaled interface given by $\tilde{h}(x,t) = b^{-\alpha}h(bx, b^z t)$. To get the relation, we will insert this into the KPZ-equation (6), and use the fact that we know how the noise η is rescaled. This knowledge comes simply from the assumption that the covariance of η is

$$\langle \eta(x,t)\eta(x',t')\rangle = \gamma\delta(t-t')\delta(x-x') \tag{33}$$

where γ is a constant. We see that the terms in the KPZ-equation are transformed according to

$$\begin{aligned} \frac{\partial}{\partial t}\tilde{h} = b^{-\alpha}b^z\frac{\partial}{\partial t}h \quad &, \quad \nabla^2\tilde{h} = b^{-\alpha}b^2\nabla h \\ (\nabla\tilde{h})^2 = b^{-2\alpha}b^2(\nabla h)^2 \quad &, \quad \tilde{\eta} = b^{-\frac{z+d}{2}}\eta \end{aligned} \tag{34}$$

Rearranging, this gives us

$$\frac{\partial}{\partial t}\tilde{h} = \nu b^{z-2}\nabla^2\tilde{h} + \frac{\lambda}{2}b^{\alpha+z-2}(\nabla\tilde{h})^2 + b^{\frac{z-2\alpha-d}{2}}\tilde{\eta} \tag{35}$$

which is the appropriate equation for the rescaled height. Now let $b \to \infty$. Physical common sense tells us that $\alpha > 0$, and it is also a fact that for growth processes $z < 2$. This can be accepted by comparing with pure diffusion, where $z = 2$ exactly, i.e. fluctuations spread in time as $\sqrt{t}$. Now growth ought to be a faster process than diffusion, so we conclude that $z < 2$.

With these constraints on the exponents, we see that the non-linear term will dominate as b grows. But then the identity $\alpha + z = 2$ must also hold, lest the scale-invariance of the equation be broken. This is a very good help to us, since it means that we only have to calculate one exponent explicitly; the other we get for free. "TANSTAAFL"

To calculate α, we will use the Burgers equation (7). Consider this equation first without a non-linear term. If we let $u(x, t = 0)$ be uncorrelated Gaußian white noise, the diffusion and noise terms in the Burgers equation will conserve this property. This is physically obvious: neither the addition of noise nor the diffusive spread should be enough to change the distribution of u much.

Surprisingly, the Gaußian noise is also invariant when we turn on the non-linearity! To see this, consider the equation

$$\frac{\partial u}{\partial t} = \frac{\lambda}{2}\frac{\partial u^2}{\partial x} \tag{36}$$

and assume that

$$\langle u(x,t)u(x',t)\rangle = \gamma\delta(x - x') \tag{37}$$

holds for $t = 0$. One way of seeing that this identity is true also for finite t is to calculate the probability density distribution function and see that it is independent of time. Equation (37) implies that for $t = 0$, the density for $u(x)$ is $\exp[-\frac{u^2(x)}{2\gamma}]$. The joint density for u defined on a strip of length L (eventually, the limit $L \to \infty$ will be taken) will then be

$$\exp[-\frac{1}{2\gamma}\int_0^L dx u^2(x,t)] \tag{38}$$

(the stochastic variables $u(x)$ and $u(x')$ are independent (see equation (37)), so their joint distribution will be the product of their densities).

Ignoring constants, differentiation of this gives

$$(\int_0^L dx \frac{\partial}{\partial t} u^2(x,t)) \times \exp[-\frac{1}{2\gamma}\int_0^L dx u^2(x,t)] \tag{39}$$

The prefactor can be rewritten, using equation (36), as

$$\int_0^L dx u \frac{\partial u}{\partial t} \propto \int_0^L dx u \frac{\partial u^2}{\partial x} = [u^3]_0^L - \int_0^L dx \frac{\partial u}{\partial x} u^2 = 0 \tag{40}$$

because of periodic boundary conditions.

So the density distribution does *not* change, and property (37) holds for all times even with a non-linear term. This is a surprising result, since it means that the non-linearity is irrelevant for the value of α. It remains to be seen what this value is. We can write, using property (37)

$$W^2(x-x',t) = \langle (h(x,t)-h(x',t))^2 \rangle = \langle (\int_x^{x'} dy u(y,t))(\int_x^{x'} dz u(z,t)) \rangle =$$
$$= \int_x^{x'} dy \int_x^{x'} dz \langle u(y,t)u(z,t) \rangle = \int_x^{x'} dy \int_x^{x'} dz \gamma \delta(y-z) \propto |x-x'| \quad (41)$$

and we see that α must be $\frac{1}{2}$. This holds for many growth models, but *only* in dimension $1+1$. When we combine it with the result $\alpha + z = 2$, which holds for the KPZ-equation in all dimensionalities, we get that $z = \frac{3}{2}$.

Next, we'll try to relate the model (27) with the KPZ-equation (6).

4.4 Is it really KPZ?

For a long time, there were no direct proofs that the model (27) and equation (6) really were in the same universality class, but this has now been proved by the numerical work by Neergaard and den Nijs[27] and Fogedby, Eriksson, and Mikheev's theoretical analysis[10]. (For the special case of $\epsilon = \pm 1$, this has previously been proved by Gwa and Spohn[13].)

Neergaard and den Nijs have measured the mass gap of the KPZ-model as a function of the linear size of the chain. They have used the method of transfer-matrices to obtain the exact values for the mass gap, instead of using Monte Carlo simulations to estimate it. (Of course, even though their methods are exact, errors are introduced when they solve numerically for the eigenvalues of the transfer matrix. These errors are however much smaller than those for MC-simulations, which enables conclusions to be drawn for much smaller sizes.)

Using the information on the mass gaps (which can be thought of as the difference in energy between the groundstate and the first excited state), the values for the dynamical exponent z can be calculated, since it is known that the mass gap scales as $m \sim N^{-z}$.

This scaling relation can be seen quite easily.[2] First, how does a characteristic energy look like? We want an expression $\exp[-E/E_0]$, where E_0 is some kind of cut-off for energy — mass gaps larger than that don't matter much. This looks very much like the quantum mechanical time-evolution operator $\exp[-Ht]$. So the cut-off in energy/mass gap corresponds to the inverse of the cut-off in time. But this is just the relaxation-time, which we know is related to the correlation length as $\tau \sim \xi^z$, by equation (2). Thus we are led to the equation $m \sim \xi^{-z}$

[2] Thanks to Per Fröjdh for help with this argument.

which is almost what we want. The correlation length we can replace with the linear size of the sample, if we just want a quick estimate of it. We are finally left with the scaling relation $m \sim N^{-z}$.

Some of the data of Neergaard and den Nijs are reproduced below in table 4. The data show that the exponent z has the value of the KPZ universality class for all non-vanishing λ.

So there is some numerical evidence at least that the model (27) really represents the KPZ-equation (6). One step further has been taken by Fogedby, Eriksson, and Mikheev[10], who have shown that the continuum-limit of the Hamiltonian (29) gives rise to the same dispersion relation as the KPZ-equation does. They have also shown that the symmetries of the KPZ-equation are present in the operator-algebra of the Hamiltonian and have managed to get an explicit relationship between the parameters of the models, ϵ and λ. More numerical evidence for the equivalence will be given below in the section on simulations.

The rest of this section consists of a more detailed and complete description of the work by Fogedby et al[10]. It is hoped that it will supplement their article by providing an easy-to-understand review of the work. First, we will see how the Hamiltonian (29) is changed by going over to action-angle variables and then taking the continuum limit of the Hamiltonian. Define therefore

$$\begin{gathered} S_j^x = s - \tfrac{1}{2}(\hat{u}_j^2 + \hat{\varphi}_j^2) \\ S_j^y = \sqrt{s}\hat{\varphi}_j \qquad\qquad S_j^z = \sqrt{s}\hat{u}_j \end{gathered} \tag{42}$$

where it should be noted that we have quantised along the x-direction instead of the more usual z: $\langle 0|S_j^x|0\rangle = s$. This maps the Hamiltonian (29) into

$$\begin{aligned} &-\sum_{j=1}^{N}(s - \frac{1}{2}[\hat{u}_j^2 + \hat{\varphi}_j^2])(s - \frac{1}{2}[\hat{u}_{j+1}^2 + \hat{\varphi}_{j+1}^2]) + s\hat{\varphi}_j\hat{\varphi}_{j+1} + s\hat{u}_j\hat{u}_{j+1} \\ &-\frac{1}{4} + i\epsilon[(s - \frac{1}{2}[\hat{u}_j^2 + \hat{\varphi}_j^2])\sqrt{s}\hat{\varphi}_{j+1} - \sqrt{s}\hat{\varphi}_j(s - \frac{1}{2}[\hat{u}_{j+1}^2 + \hat{\varphi}_{j+1}^2])] = \\ = &-\sum_{j=1}^{N}\frac{1}{4}(\hat{u}_j^2 + \hat{\varphi}_j^2)(\hat{u}_{j+1}^2 + \hat{\varphi}_{j+1}^2) - \frac{s}{2}(\hat{u}_j^2 + \hat{u}_{j+1}^2 + \hat{\varphi}_j^2 + \hat{\varphi}_{j+1}^2) + s[\hat{\varphi}_j\hat{\varphi}_{j+1} + \\ &\hat{u}_j\hat{u}_{j+1}] + i\epsilon[s^{\frac{3}{2}}(\hat{\varphi}_{j+1} - \hat{\varphi}_j) + \frac{\sqrt{s}}{2}(\hat{\varphi}_j(\hat{u}_{j+1}^2 + \hat{\varphi}_{j+1}^2) - \hat{\varphi}_{j+1}(\hat{u}_j^2 + \hat{\varphi}_j^2))] \end{aligned} \tag{43}$$

where the constant terms have been removed. The continuum limit now entails the following substitutions

$$\begin{gathered} \hat{u}_j \to \sqrt{a}\hat{u}(x), \quad \hat{u}_{j+1} \to \sqrt{a}(\hat{u}(x) + a\hat{u}'(x)) \\ \hat{\varphi}_j \to \sqrt{a}\hat{\varphi}(x), \quad \hat{\varphi}_{j+1} \to \sqrt{a}(\hat{\varphi}(x) + a\hat{\varphi}'(x)) \\ \textstyle\sum_{j=1}^{N} \longrightarrow \int \frac{dx}{a} \end{gathered} \tag{44}$$

where a is the lattice-constant and $'$ refers to the derivative with respect to x.

After the substitution, the "real" part of the Hamiltonian looks like

$$\int \frac{dx}{a}\frac{a^2}{4}(\hat{u}(x)^2+\hat{\varphi}(x)^2)[\underbrace{\hat{u}(x)^2+\hat{\varphi}(x)^2}_{3}+\underbrace{2a\hat{u}(x)\hat{u}'(x)+2a\hat{\varphi}(x)\hat{\varphi}'(x)}_{4}$$
$$+\underbrace{a^2\hat{u}'(x)^2+a^2\hat{\varphi}'(x)^2}_{2}]-\frac{sa}{2}[\underbrace{2\hat{u}(x)^2+2\hat{\varphi}(x)^2}_{5}+\underbrace{2a\hat{u}(x)\hat{u}'(x)+2a\hat{\varphi}(x)\hat{\varphi}'(x)}_{1}$$
$$+\underline{a^2\hat{u}'(x)^2+a^2\hat{\varphi}'(x)^2}]+sa(\underbrace{\hat{\varphi}(x)^2+\hat{u}(x)^2}_{5}+\underbrace{a\hat{\varphi}(x)\hat{\varphi}'(x)+a\hat{u}(x)\hat{u}'(x)}_{1}) \quad (45)$$

We see first that some terms(1) admit immediate integration, while others(2) are of order $\mathcal{O}(a^3)$ and will disappear when the limit $a \to 0$ is taken. The term (3) is a constant, terms (4) disappear since

"This type of math should only be allowed in closed rooms between N consenting adults." — Terry Pratchett

$$\int \hat{u}^2\hat{\varphi}\hat{\varphi}' = 0 - \int \hat{\varphi}(2\hat{u}\hat{u}'\hat{\varphi}+\hat{u}^2\hat{\varphi}') \quad (46)$$

and $\int \hat{u}^3\hat{u}'$ is integrable. Finally, terms (5) kill each other pair-wise.

The "imaginary" part isn't much prettier:

$$\int \frac{dx}{a}(sa)^{\frac{3}{2}}[\underbrace{\hat{\varphi}(x)}_{5}+\underbrace{\hat{\varphi}'(x)}_{1}-\underbrace{\hat{\varphi}(x)}_{5}]+\frac{\sqrt{s}a^{\frac{3}{2}}}{2}[\underbrace{\hat{\varphi}(x)\hat{u}(x)^2}_{5}+\underline{2a\hat{u}(x)\hat{u}'(x)\hat{\varphi}(x)}+$$
$$+\underbrace{a^2\hat{\varphi}(x)\hat{u}'(x)^2}_{2}+\underbrace{\hat{\varphi}(x)^3}_{5}+\underbrace{2a\hat{\varphi}(x)^2\hat{\varphi}'(x)}_{1}+\underbrace{a^2\hat{\varphi}(x)\hat{\varphi}'(x)^2}_{2}+$$
$$-\underbrace{\hat{\varphi}(x)\hat{u}(x)^2-\hat{\varphi}(x)^3}_{5}-\underline{a\hat{u}(x)^2\hat{\varphi}'(x)}-\underbrace{a\hat{\varphi}(x)^2\hat{\varphi}'(x)}_{1}] \quad (47)$$

The terms disappear for the same reasons as above. Note that the remaining terms are in fact the same since by partial integration

$$\int \hat{u}\hat{u}'\hat{\varphi} = [\hat{u}^2\hat{\varphi}] - \int \hat{u}(\hat{u}'\hat{\varphi}+\hat{u}\hat{\varphi}') = -\int \hat{u}\hat{u}'\hat{\varphi} - \int \hat{u}^2\hat{\varphi} \quad (48)$$

Introducing the constant $J = \sqrt{sa}$ and rescaling time by a factor $\sqrt{s}a^{\frac{3}{2}}$, we finally get

$$H = \int dx \frac{J}{2}(\hat{u}'(x)^2+\hat{\varphi}'(x)^2)+i\epsilon\hat{u}(x)^2\hat{\varphi}'(x) \quad (49)$$

which is the proper continuum limit of equation (29). Note that the rescaling of time is permitted since it is $\exp[-Ht]$ that is the physically important operator, and that this rescaling is compatible with a dynamical exponent $z = \frac{3}{2}$. (Further evidence of the correctness of this rescaling will be given below.)

To show that this Hamiltonian is indeed equivalent to the KPZ-equation, we now need the correct equation for the probability distribution $f(u,t)$. The distribution $f(u,t)$ is the probability of having a particular set $u(x)$ of steps at time t. The step from the equation describing the time-evolution of u (the "Langevin" equation) to that describing the time-evolution of f (the "Focker-Planck" equation) will now be motivated. The difference between this procedure and that described in textbooks on statistical physics[35] is that u is here a function (and hence $f(u,t)$ a functional), and not a position or velocity (the phase-space is a Hilbert-space instead of $\mathcal{R}^n$).

We start from the Master equation $\frac{\partial f}{\partial t} = \sum W(u,u')f(u') - W(u',u)f(u)$ where the sum is over all possible functions u' and $W(u,u')$ is the probability of a transition between the states u and u'. Now we need the following *Volterra series*[36], functional analogues of the ordinary Taylor-expansions for functions,

$$
\begin{aligned}
f(u,t) &= f(u_0,t) + \int dx[u(x)-u_0(x)]\frac{\delta f(u_0,t)}{\delta u(x)} + \\
&+\frac{1}{2}\int dxdx'[u(x)-u_0(x)][u(x')-u_0(x')]\frac{\delta^2 f(u_0,t)}{\delta u(x)\delta u(x')} + \mathcal{O}(||u-u_0||^3) \\
W(u,v) &= W(u_0,v) + \int dx'[u(x')-u_0(x')]\frac{\delta W(u_0,v)}{\delta u(x')} + \\
&+\frac{1}{2}\int dx'dx''[u(x')-u_0(x')][u(x'')-u_0(x'')]\frac{\delta^2 W(u_0,v)}{\delta u(x')\delta(u(x''))} + \mathcal{O}(||u-u_0||^3) \qquad (50)
\end{aligned}
$$

where $\frac{\delta}{\delta u(x')}$ is the functional derivative with respect to $u(x')$ and $||\cdot||$ is some norm. Formally, the proof of these expansions is entirely similar to that for Taylor-expansions, except that in order to keep the right hand side a functional we must introduce an integration $\int dx$ each time we do a partial integration of u.

Insertion of (50) in the Master equation leads to

$$
\frac{\partial f}{\partial t} = -\int dx \frac{\delta}{\delta u(x)}[A(u,x)f(u,t)] + \frac{1}{2}\int dxdx' \frac{\delta^2}{\delta u(x)\delta(u(x'))}[B(u,v,x,x')f(u,t)] \qquad (51)
$$

with $A(u,x) = \sum_v v(x)W(v,u)$ and $B(u,v,x,x') = \sum_v v(x)v(x')W(v,u)$. There is actually a physical interpretation of A and B. Since W is the probability of switching to another state, A is simply the average change in $u(x)$ per unit time, while B is the change in $u(x)u(x')$. But then A can also be written

$$
\begin{aligned}
A(u,x) &= \frac{\langle u(x,t+\tau) - u(x,t)\rangle}{\tau} = \frac{1}{\tau}\int_t^{t+\tau} dt' \langle \frac{du(x,t')}{dt'} \rangle = \\
&= \frac{1}{\tau}\int dt' \langle \lambda u \partial u/\partial x + \nu \partial^2 u/\partial x^2 \rangle + \frac{1}{\tau}\int dt' \langle \partial \eta/\partial x \rangle \qquad (52)
\end{aligned}
$$

by using the Burgers equation (7). Now the average of $\partial\eta/\partial x$ is zero, while taking the average of the other term is the same as setting $t' = t$, so that finally $A(u,x) = \lambda u \partial u/\partial x + \nu \partial^2 u/\partial x^2$ The expression for B is similar:

$$B(u,x,x') = \frac{1}{\tau}\int dt'dt''\langle \frac{u(x,t')}{dt'}\frac{u(x',t'')}{dt''}\rangle =$$

$$= \frac{1}{\tau}\int dt'dt''\langle[\lambda u(x,t')\frac{\partial u(x,t')}{\partial x}+\nu\frac{\partial^2 u(x,t')}{\partial x^2}][\lambda u(x',t'')\frac{\partial u(x',t'')}{\partial x'}+\nu\frac{\partial^2 u(x',t'')}{\partial x'^2}]\rangle +$$

$$+\frac{1}{\tau}\int dt'dt''\langle\frac{\partial\eta(x,t')}{\partial x}\frac{\partial\eta(x',t'')}{\partial x'}\rangle \tag{53}$$

where the "cross-terms" disappear since the average of $\partial\eta/\partial x$ is zero. We see that the first term is of order $\mathcal{O}(\tau)$ and can be ignored when the limit $\tau \to 0$ is taken, while using $\langle\eta(x,t)\eta(x',t')\rangle = \gamma\delta(x-x')\delta(t-t')$ transforms the second into

$$B = \gamma\frac{\partial^2}{\partial x\partial x'}\delta(x-x') \tag{54}$$

Finally, then, we are left with

$$\frac{\partial f}{\partial t} = -\int dx\frac{\delta}{\delta u}[(\lambda u\frac{\partial u}{\partial x}+\nu\frac{\partial^2 u}{\partial x^2})f]$$

$$+\frac{\gamma}{2}\int dx d\tilde{x}[\frac{\partial^2}{\partial x\partial\tilde{x}}\delta(x-\tilde{x})]\frac{\partial^2 f}{\partial u\partial u'} = \mathcal{L}(u)f \tag{55}$$

which is the appropriate Focker-Planck equation.

The equivalence between $\mathcal{L}$ and H can now be shown. If the operators u and $\frac{i\delta}{\delta u}$ in $\mathcal{L}$ are transformed according to

$$\frac{i\delta}{\delta u} \to \sqrt{\frac{\nu}{\gamma}}(\hat{\varphi}(x) - i\hat{u}(x)) \tag{56}$$

and

$$u \to \sqrt{\frac{\gamma}{\nu}}\hat{u}(x) \tag{57}$$

the operator $\mathcal{L}$ is changed into

$$\int dx i\sqrt{\frac{\nu}{\gamma}}(\hat{\varphi}-i\hat{u})[\lambda\frac{\gamma}{\nu}\hat{u}\hat{u}'+\nu\sqrt{\frac{\gamma}{\nu}}\hat{u}'']+$$

$$+\frac{\gamma}{2}\int dx dx'[\frac{\partial^2}{\partial x\partial x'}\delta(x-x')]\frac{-\nu}{\gamma}(\hat{\varphi}(x)-i\hat{u}(x))(\hat{\varphi}(x')-i\hat{u}(x')) \tag{58}$$

Doing two partial integrations in the double-integral turns this into

$$\int dx i\frac{\gamma}{\nu}\lambda\varphi\hat{u}\hat{u}' + \frac{\gamma}{\nu}\lambda\hat{u}^2\hat{u}' + i\nu\hat{\varphi}\hat{u}'' + \nu\hat{u}\hat{u}''+$$

$$-\frac{\nu}{2}\int dx dx'\delta(x-x')[\hat{\varphi}'(x)-i\hat{u}'(x)][\hat{\varphi}'(x)-i\hat{u}'(x')] \tag{59}$$

which by using $\int \hat\varphi \hat u \hat u' = -\int \hat\varphi \hat u' \hat u - \int \hat\varphi' \hat u^2$, and remembering that periodic boundary conditions kill integrated terms, becomes

$$-\int dxi\sqrt{\frac{\gamma}{\nu}}\frac{\lambda}{2}\hat\varphi'\hat u^2 + \int dxi\nu\hat\varphi\hat u'' - \int dx\nu(\hat u')^2 - \int dx\frac{\nu}{2}(\hat\varphi')^2 + \\ + \int dx\frac{\nu}{2}(\hat u')^2 + \int dx\nu i\hat\varphi'\hat u' \qquad (60)$$

and is simplified to

$$-\int dxi\frac{\lambda}{2}\sqrt{\frac{\gamma}{\nu}}\hat\varphi'\hat u^2 - \frac{\nu}{2}[(\hat\varphi')^2 + (\hat u')^2] \qquad (61)$$

since $\int \hat\varphi \hat u'' = -\int \hat\varphi' \hat u'$. We see now that $\mathcal{L} = -H$ if we make the identifications $J = \frac{\nu}{2}$ and $\epsilon = \frac{\lambda}{2}\sqrt{\frac{\gamma}{\nu}}$. This proves the equivalence between the spin-hamiltonian and the growth-model, since the time-development of a state $|\cdot\rangle$ corresponding to $P(u,t)$ is given by $d|\cdot\rangle/dt = -H|\rangle$, to first order. It is interesting to note that whereas there are three "free" parameters (ν, λ, and γ) in the equation, there are only two (ϵ and J) in the spin-chain version of the problem.

Now forget that $\hat u$ and $\hat\varphi$ are actually operators, and consider them to be "ordinary" fields. Their time-evolution is then determined by the Poisson-bracket with H

$$\frac{\partial\varphi}{\partial t} = iJ\frac{\partial^2 u}{\partial x^2} + 2\epsilon u\frac{\partial\varphi}{\partial x} \qquad (62)$$

$$\frac{\partial u}{\partial t} = -iJ\frac{\partial^2 \varphi}{\partial x^2} + 2\epsilon u\frac{\partial u}{\partial x} \qquad (63)$$

where we have used the relation $\{y(x), \varphi(y)\} = \delta(x-y)$.

What kind of solutions do these equations have? For now, assume that they will be *solitons* propagating with a velocity v. A soliton is a generalised wave-packet. Solutions of the ordinary wave-equation have two common properties:

- localised wave-packets travel with uniform velocity and without having their shape distorted;
- separate wave-packets can collide and still continue with their original velocities and shapes.

The addition of a non-linear term to the wave-equation will destroy these properties. If we also add a dispersive term, however, there will be a small number of solutions that still have these two properties. These solutions are called solitons ([33], chap. 2).

We can now rewrite equations (62) and (63), substituting $-v\partial/\partial x$ for $\partial/\partial t$. Doing this, multiplying the equations with $\frac{\partial u}{\partial x}$ and $-\frac{\partial\varphi}{\partial x}$, and summing them gives

$$0 = \frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x} + \frac{\partial\varphi}{\partial x}\frac{\partial^2\varphi}{\partial x} \propto \frac{\partial}{\partial x}[(\frac{\partial u}{\partial x})^2 + (\frac{\partial\varphi}{\partial x})^2] \qquad (64)$$

Integrating this and recalling that for solitons the derivatives of u and φ vanish at $x = \pm\infty$ leads to the relation

$$\frac{\partial u}{\partial x} = \pm i \frac{\partial \varphi}{\partial x} \tag{65}$$

which, inserted in the equation of motion (63), in turn gives us the equation

$$-vu' = \pm Ju'' + 2\epsilon uu' \tag{66}$$

Integration gives

$$u' = \pm\frac{\epsilon}{J}(u^2 + \frac{v}{\epsilon}u + \frac{C}{\epsilon}) = \pm\frac{\epsilon}{J}(u - u_+)(u - u_-) \tag{67}$$

where we have used the boundary values $u_\pm = u(x \to \pm\infty)$ (remember that u' must be zero at infinity) and the relation $u_+ + u_- = -\frac{v}{\epsilon}$.

Equation (67) is yet another of that rare kind of equation that can be solved exactly. It is equivalent to

$$\int \frac{du}{u^2 - (u_x + u_-)u + u_+u_-} = \pm\frac{\epsilon}{J}\int dx = \frac{\epsilon}{J}x \tag{68}$$

with formal solution

$$K\frac{2u - (u_+ + u_-) - |u_+ - u_-|}{2u - (u_+ + u_-) + |u_+ - u_-|} = \exp\left[-\frac{|\epsilon|}{J}|u_+ - u_-|(x - x_0)\right] \tag{69}$$

(the constant K comes from the lower limit of the left side integral above; it can be ignored). This can be simplified to

$$u = \frac{u_- + u_+ \exp\left[-\frac{|\epsilon|}{J}|u_+ - u_-|(x - x_0)\right]}{1 + \exp\left[-\frac{|\epsilon|}{J}|u_+ - u_-|(x - x_0)\right]} \tag{70}$$

which describes a kink in u on its way from u_- (at $x = -\infty$) to u_+ (at $x = +\infty$). Because the ground-state is $u = 0$, an excitation above it must go to zero at infinity. This we can achieve if we pair the solutions in two, with $u_\mp = -\frac{v}{\epsilon}$ and $u_\pm = 0$ for kink $\binom{1}{2}$. We now just have to insert the solution u into the continuum version of the Hamiltonian to get the energy for a pair $E = 2H(u) = -\frac{2|v^3|}{3\epsilon^2}$.

If we now also get an expression for the momentum, we can eliminate v and get a dispersion relation. The momentum operator is given by $\hat{P} = \int dx\hat{u}\partial\hat{\varphi}/\partial x$, a strange expression that we will now attempt to motivate.

From classical mechanics, we know that momentum is the generator of spatial translation. In quantum mechanics, the operator $\mathcal{F}(x) = \exp[-i\hat{P}x]$ translates a state a distance x. For a spin-chain of length N, we can view this translation as a transformation of all spins $\sigma_j^{new} = \sigma_{j-x}^{old}$. If we assume that there is an operator $\mathcal{F}_j(x)$ that translates spin j into position $j+x$, we can write the operator $\mathcal{F}$ as $\prod_j \mathcal{F}_j(x)$.

If we assume that $\mathcal{F}_j$ can be written as $\exp[-i\hat{P}_j x]$, we can use the fact that $\prod \exp[\cdot] = \exp[\sum \cdot]$, to write the momentum as $\hat{P} = \sum_j \hat{P}_j$.

An expression for $\hat{P}_j$ is easy to motivate. The operator

$$\frac{1}{i}[\hat{\sigma}_j^z(\hat{\sigma}_{j+1}^+ - \hat{\sigma}_{j+1}^-) + \hat{\sigma}_{j+1}^z(-\hat{\sigma}_{j+1}^+ + \hat{\sigma}_{j+1}^-)] \tag{71}$$

will do, as is easily seen by letting it operate on different kets. If we now hand-wave away all questions about constants and other details, we can write the momentum as

$$\hat{P} = \sum_j \frac{1}{i}[\hat{\sigma}_j^z(\hat{\sigma}_{j+1}^+ - \hat{\sigma}_{j+1}^-) + \hat{\sigma}_{j+1}^z(-\hat{\sigma}_{j+1}^+ + \hat{\sigma}_{j+1}^-)] = \sum_j \frac{\hat{\sigma}_j^z}{i}(\hat{\sigma}_{j+1}^+ - \hat{\sigma}_{j+1}^- - (\hat{\sigma}_j^+ - \hat{\sigma}_j^-)) \tag{72}$$

or, using $\hat{\sigma}^y = \frac{\hat{\sigma}^x - \hat{\sigma}^-}{2i}$ and ignoring constants

$$\sum_j \hat{\sigma}_j^z(\hat{\sigma}_{j+1}^y - \hat{\sigma}_j^y) \propto \sum_j \hat{u}_j(\hat{\varphi}_{j+1} - \hat{\varphi}_j) \tag{73}$$

which in the continuum limit will go into

$$P = \int dx \hat{u}(x) \frac{\partial \hat{\varphi} x}{\partial x} \tag{74}$$

which is what we started with. It must be stressed that this is not a rigorous derivation of $\hat{P}$ (too many details have been skipped over), but is meant to serve only as a motivation for the expression (74).

Now again forgetting that $\hat{u}$ and $\hat{\varphi}$ are operators, using the relation (65) between $\partial u/\partial x$ and $\partial \varphi/\partial x$ and the expression (70) for u, it is easy to see that the momentum for a pair is $-i(\frac{v}{\epsilon})^2 \mathbf{sgn}(v)$ and hence that the dispersion relation is

$$E = -\frac{2}{3}|\epsilon|(iP\,\mathbf{sgn}(v))^{3/2} \tag{75}$$

We see that the dynamical exponent is indeed $3/2$, if we imagine that we can label the pairs by their momenta:

$$t \sim \omega^{-1} \sim E^{-1} \sim p^{-3/2} \sim k^{-3/2} \sim r^{3/2} \tag{76}$$

If we add the operators $\hat{M} = \int dx \hat{u}$ and $\hat{\Phi} = \int dx \hat{\varphi}$ (which generate rotations around the z- and y-axles) to $\hat{P}$ and $\hat{H}$ (which generates translations in time, as usual) and use the commutation relation between $\hat{u}$ and $\hat{\varphi}$, $[\hat{\varphi}(x), \hat{u}(x')] = i\delta(x-x')$, we get an operator algebra with two non-zero commutators:

$$[\hat{\Phi}, \hat{M}] = iL, \quad [\hat{H}, \hat{\Phi}] = 2\epsilon\hat{P} \tag{77}$$

This can be interpreted as a justification of the rescaling of time with a factor $a^{3/2}$ before equation (49). Had this renormalization not been performed, the constant 2ϵ above would have depended on a. But this algebra should naturally be independent of the particular scale which we have chosen for the system. If a rescaling of space introduces a scale-dependency in the algebra, we must do another rescaling of time to remove it. This gives us additional proof that the dynamical exponent is $\frac{3}{2}$.

The approximations used to obtain the results are:

- The continuum limit of the Hamiltonian (29) has been taken, in order to show its equivalence to the KPZ-equation (6);
- The resulting equations of motion have been converted to their non-quantum mechanical versions, the classical solutions have then been quantised in order to label them using k.

These approximations are very reasonable, and there is little reason to doubt them, especially since the results confirm the general trend obtained in the simulations described in section 6.

5 Experiments

This section on experiments is far from complete — the huge number of papers published on the KPZ-equation prevents this. Neither does it claim to provide the best examples of where the KPZ-equation *is* completely applicable. Instead we will show some cases with good correspondence between theory and experiments, more with only partial correspondence, and some with none at all. It is hoped that this will at least show that the KPZ-equation is not only a mathematical curiosity and give a representative sampling of the experiments performed to date, as well as further describe the complicated interplay between theory and reality in this particular branch of physics.

Sadly, there is no experimental data available for one case where the KPZ-equation is believed to be correct, and which we all see every day. Consider a beam of water impacting on a slanted surface, such as a plate that you are about to dish. After the water has first impacted on the plate, it will spread out to cover all of it. During this brief period of time, the appearance of the interface between wet and dry will be governed by the $1+1$-dimensional KPZ-equation. Here it is gravity that gives rise to the diffusion-like term, while the other terms occur for the usual reasons.

"Natural Laws have no pity."
— Lazarus Long

5.1 Stars

It is now time to turn to some of the measurements that conform completely to the theoretical predictions of the KPZ-equation. Our first example, which unfortunately takes place in dimensionality $3+1$, instead of the theoretically better investigated $1+1$ concerns the reason why we exist. One of the mysteries of the universe is why there are stars and galaxies at all. Why is the energy-density distribution in space inhomogeneous? To put it in fancier words, what caused the initial, flat metric to turn into a wrinkled one?

"We could look up at the stars and never be afraid again."

The standard explanation for this is that the differences in density are caused by quantum fluctuations early in the universe's history. This might be all the explanation we need. During some time-periods, there could, however, also be influence from large-scale stochastic fluctuations, similar to those occurring in surface growth.

Berera and Fang[1] use the universality of the fluctuation-dissipation theorem to argue that these fluctuations must exist. They claim that dissipation effects have been very much worked upon in cosmology, and hence argue that it is interesting to study also stochastic fluctuations.

How do we get a KPZ-type equation from the standard equations of cosmology? First, it is sufficient to use a non-relativistic theory. This is true only for the stochastic fluctuation-driven structure formation. Much of the structure formation of universes takes place during the inflation period, and for this period a relativistic treatment is necessary. During these periods dissipation effects are not very important and the structure formation is driven entirely by quantum effects.

Let's take as our starting point therefore a linear approximation of the standard momentum equation[19]

$$\frac{\partial \vec{v}}{\partial t} + \frac{\dot{R}}{R}\vec{v} + \frac{\dot{R}}{R}(\vec{r}\cdot\nabla)\vec{v} + \frac{v_s^2}{\rho_0}\nabla\rho + \nabla\phi = 0 \tag{78}$$

The density ρ, velocity v and gravitational potential ϕ are here the deviations from the smooth solutions, R is the cosmic scale-factor, while v_s is the sound-velocity. Assume now that there is a velocity-potential ψ such that $v = -\nabla\psi$. It is also true (in the linear approximation) that the velocity is proportional to the force induced by gravity. We therefore have $\rho = -f\nabla \cdot v$, where f is a constant related to the Hubble constant. This actually tells us that the gravitational potential ϕ and the velocity potential ψ are in fact proportional to each other, and $\vec{v} \propto -\nabla\phi$. Substitution of these equations into equation (78) gives

$$\frac{\partial \vec{v}}{\partial t} + \frac{\dot{R}}{R}\vec{v} + \frac{\dot{R}}{R}(\vec{r}\cdot\nabla)\vec{v} = \frac{v_s^2}{\rho_0}f\nabla^2\vec{v} + 4\pi G f\vec{v} \tag{79}$$

which is a Langevin equation without stochastic noise. This can however be added because we know that dissipation will give rise to fluctuation, as argued above. Doing this and also adding a non-linear term (which is necessary because otherwise the noise term is irrelevant) gives us

$$\frac{\partial \vec{v}}{\partial t} + \frac{\dot{R}}{R}\vec{v} + \frac{\dot{R}}{R}(\vec{r}\cdot\nabla)\vec{v} + (\vec{v}\cdot\nabla)\vec{v} = \frac{v_s^2}{\rho_0}f\nabla^2\vec{v} + 4\pi G f\vec{v} + \vec{\Xi} \tag{80}$$

(the addition of the non-linear term can be motivated by a desire to leave the linear approximation and try to capture some features of the full momentum equation).

This is the equation we must use to describe the evolution of the density-distribution in all eras where dissipation is important, while general relativity isn't. Besides the reheating stage during the early evolution of the universe, this might also be the case for collisions of galaxies.

To get the Burgers equation we must now use our assumption that dissipation is the most important effect in this equation. We can then ignore the self-gravitation ($4\pi G f v$-term) and Hubble-expansion (all R-terms) and are, after perhaps too many approximations, finally left with the Burgers equation.

The equation for the gravitational potential must then be the KPZ-equation. The exponents of the $3+1$-dimensional KPZ-equation are not known exactly, but their approximate values are consistent with those observed for the matter distribution of the universe.

5.2 Clouds

"orange clouds raining in my head"
— Tori Amos

Pelletier[30] has shown that the $2+1$-dimensional KPZ-equation is also applicable to the appearance of certain clouds. Using satellite and space-shuttle photos of cumulus cloud-covers, Pelletier has calculated a cumulative frequency distribution for the area of the clouds. (That is, he has calculated how many clouds that have an area larger that Λ.) For the KPZ-equation (6) this should scale with Λ as $\Lambda^{-\frac{2-\alpha}{2}}$, giving ($\alpha = 0.38$ for $d = 2+1$) an exponent of approximately 0.8. The data shows excellent agreement over almost ten decades.

What, then, in the atmosphere are the causes of the different terms in the KPZ-equation?

We work now in dimensions $2+1$, and let h represent the height of a point of an originally horizontal layer of very thin fluid in the atmosphere. Two things will happen to warm air in the atmosphere:

- The air will rise vertically as it is heated;
- The air will also *expand* (and cool) as it reaches regions of lower pressure higher up in the atmosphere.

The first of this will cause instabilities in the fluid layer, which are most easily modelled as stochastic noise, giving us the third term in equation (6).

The tendency for air to expand/contract as it ascends/descends gives us the non-linear term, for this we can model as a tendency of the interface to grow not vertically but always perpendicular to itself. By the same reasoning as in the introduction, we can write this as a term proportional to $(\nabla h)^2$.

Now, because there's viscosity in air, there will be forces between ascending hot air and the surrounding gas, and this we can model as an effective surface tension, which finally leads us to the last term in the equation.

Pelletier is the source of many interesting examples of the KPZ-equation in action. He has also observed plankton in the oceans[29] and found that their abundance can be described by a version of the KPZ-equation. How a particular species spreads in an ecosystem is one of the central problems in ecology. Apart from plankton in the sea, it might also be possible to study the colonisation habits of various kinds of trees or even of insects. Using data from lidar[3] observations, Pelletier has calculated the power-spectrum and found that the observed scaling is the same as that of the KPZ-equation. The equation he has used is a variant of the diffusion-equation, where the growth-rate is considered to be a stochastic variable

$$\frac{\partial c}{\partial t} = D\nabla^2 c + \eta c \tag{81}$$

[3]"Light detection and ranging", a radar-like system that uses laser-pulses instead of microwaves.

But this we recognise as equation (8), which described walks in random media and was the result of an exponential transformation of the KPZ-equation.

This equation is physically more correct than the version with a constant growth-rate and a random production (the Edwards-Wilkinson equation (5)), since it lets the reproduction of the plankton be influenced by both the randomness and the concentration of plankton in different places.

5.3 Electrochemistry

Let's now look at one case where the KPZ equation is only partially applicable. Consider slow-rate electrochemical deposition of copper. The growth of the surface is obviously dependent on the number of active cations. For low currents at least, local effects should thus play a significant role, leading us to suspect that the KPZ equation might be applicable. If we, on the other hand, increase the current, the growth should be more influenced by diffusion in the electrolyte — a non-local effect. This type of growth is not well understood, and it is interesting to study the competition between low and high current behaviour.

One such study has been performed by Kahanda et al [15]. Their setup was simple. A 25 μm thick copper foil was used as cathode and initial growth surface. Opposite from the growth area a copper rod, acting as anode, was placed. There was also a reference rod to control the potential. A video camera was used to take pictures of the growing surface.

It is obvious that the characteristics of the surface growth is dependent on the current strength. To obtain quantitative results regarding the crossover from DLA- to KPZ-type growth, the power-spectrum of the height was calculated. Scaling tells us that the long time behaviour of $S(k,t)$ is $\frac{1}{k^{1+2\alpha}}$.

Fitting the data to this law gives $0.5 < \alpha < 0.6$ for all currents. This is consistent with KPZ, which says $\alpha = 0.5$ for $D = 1+1$. Another characteristic of KPZ-growth is that the width should grow as t^{β} with $\beta = 1/3$ for small times. This is however *not* the case for the experimental data – they imply that $\beta > 1$. This is most likely due to non-local effects that cause other terms to appear in the KPZ-equation.

5.4 Other results

An interesting method of observing thin films is described by Krim et al[20]. They use scanning tunnnel microscopy to observe the surface formed by bombarding iron with Ar-ions. This is of course a model example of where the reasoning leading to the KPZ-equation (6) is correct. It is very fortunate, therefore, that the results they get is that the roughness exponent is $\alpha = 0.53 \pm 0.02$, which indicates that the theoretical value 0.5 probably is correct. (They haven't measured the exponent β, though, so we don't know if the KPZ-equation is completely applicable.)

We are not so fortunate when it comes to the interface between two different fluids in a porous media. Rubio et al[34] have found that the exponents here are very dependent on the type of fluids that are interacting. Depending on the relative viscosity of the fluids, we may get diffusion-limited-aggregation (DLA), a sort of percolation, or interfaces that aren't even fractals. One reason that we don't get KPZ-type behaviour might be experimental errors. It is possible that the experiments are not truly two- or three-dimensional, but that some sort of crossover-behaviour is seen. The value of the roughness exponent (0.71) doesn't rule out this possibility.

This is probably also the reason why the measurements on various kinds of bacteria[39, 26] don't give the right exponents. Since there is necessarily several layers of bacteria that interact with each other, we get neither a truly two-dimensional nor a three-dimensional (because of the thinness of the layers) behaviour.

This explanation won't help in the case of forest-fires, however. Zhang et al[44] have examined the propagation of fire-fronts on thin papers, which ought to be a good model for forest fires, and have found a roughness exponent that is also well above that of the KPZ-equation. Since the paper is so thin, they get a truly two-dimensional situation, and because the already-burnt paper doesn't interact at all with the non-burnt paper, they avoid some of the troubles that plague the experiments on fluid-displacement. It is reassuring to know that all three of these experiments have obtained approximately the same value for α, a happy coincidence for which Zhang et al[44] have proposed an explanation. Throughout the theoretical treatment of the KPZ-equation, it is always assumed that the noise η is Gaußian; this leads to the familiar exponents. If this is not the case, however, Zhang has shown that the roughness exponent will become larger than 0.5. Another explanation could be that non-local effects (such as the pinning of a section of the interface) are important.

5.5 Conclusions

We have seen that in some cases there still is not very good agreement between experiments and theory. Even though we think that the KPZ-equation, and the physical reasoning applied to derive it, should be applicable to a broad range of physical phenomena, there are cases where Nature lets us down in this regard.

Why is this so? We have seen that it is not because of higher-order non-linear terms in the equation (since they are irrelevant), but there might still be other terms missing. Another possibility[16] is that

"They savoured the strange warm glow of being much more ignorant than ordinary people, who were only ignorant of ordinary things."
— Terry Pratchett

we must use more coarse-grained variables than h, and hence solve several coupled differential equations instead of just the KPZ-equation. It is reassuring to know that the number of experiments performed on growth is growing at a very fast rate every year. Hopefully, some of these experiments will provide the key insight necessary to *grok the fullness*[14] of growth.

6 The Simulations

This section will present the simulations that have been performed. First, the various quantities that have been measured will be presented, and their relationships to the scaling exponents will be proved. Then, the *polynuclear growth* (PNG) model will be introduced, and the results of simulations of it as well as the model given by (27) presented. Since there is nothing mysterious about the PNG-model, and there is independent data available to compare with, this is a good test of the programs later used to obtain data for the KPZ-model.

Source code in C++ for the programs used are available in appendix B. Only the program for the KPZ-model are included. The only difference between it and that for the PNG-model is that the growth-rule (represented in the code by the *evolve()*-function) is different.

"It has been discovered that C++ provides a remarkable facility for concealing the trivial details of a program — such as where its bugs are."

Also included (in appendix C) is a Java-applet that shows how an initially flat interface evolves in time using the growth rule (27). The self-similar properties of the interface can be seen by running this program, but it is difficult to see if there really is any difference in the growth-behaviour for different values of ϵ. It is easy to modify the program to study other growth models, such as the PNG-model.

6.1 What shall we measure?

In order to measure the dynamical exponents we must use the scaling properties of the height variable. Further, we will concentrate on measuring the exponent $\beta = \frac{\alpha}{z}$ (for the KPZ-equation (6), this simplifies to $\beta = \frac{2}{z} - 1$, since $\alpha + z = 2$ for it). This can be motivated by noting that α has the same value for both the Edwards-Wilkinson (5) and KPZ (6) equations — measurements of it do not help us investigate possible exotic cross-over behaviours in model (27) as ϵ is varied. The simplest way to see that α is $\frac{1}{2}$ for the model is to iterate the growth rule until a steady-state is reached and then measure the variance for different sizes and ϵ's.

Remember now the condition for a surface to be self-affine,

$$h(x,t) = b^{-\alpha} h(bx, b^z t) \tag{82}$$

Caution must be used in the interpretation of this equation. It is not applicable locally — it means only that the statistical properties of h remain the same if we do the indicated rescaling.

Let's see what this means for the height difference correlation function

$$G(r,t) = \langle |h(x,t) - h(x+r,t) \rangle_x \tag{83}$$

where the notation $\langle \cdot \rangle_x$ means averaging over all x. In the limit $t \to \infty$, G relaxes towards equilibrium and becomes independent of time — multiplying t with b^z shouldn't change anything. But this multiplication introduces (by dynamical scaling) a factor $b^{-\alpha}$ in front of G and changes its first argument from x to bx. The equality

$$\langle |h(x,t) - h(x+r,t)\rangle_x = b^{-\alpha}\langle |h(bx, b^z t) - h(b(x+r), b^z t)\rangle_x \tag{84}$$

must then hold, for large t. (There is a small error here in that r has been ignored on the right side, but this is unimportant for this reasoning.) But then the x-dependency of h must cancel the factor $b^{-\alpha}$, which leads to the width being

$$\xi_\perp(r) \equiv \lim_{t\to\infty} G(r,t) = Cr^\alpha \tag{85}$$

To measure this exponent presents no difficulties at all, and nobody questions that it has the value $\frac{1}{2}$ for the KPZ-equation (and the model (27)). It has even been argued[23] that the only possible values for the roughness exponent is $\frac{1}{2}$ or 1, if we want the surface to be self-affine. (López and Rodríguez[23] introduce a distinction between a *local* and a *global* roughness exponent (most simulations measure the local exponent). For a truly self-affine surface, the exponents must be equal. This can happen only for $\alpha = n/2$, n integer.)

Now let's insert the scaling condition (82) into the definition of G and see what condition we get on it. We see immediately that $G(r,t) = b^{-\alpha}G(br, b^z t)$ must be fulfilled. We are of course free to chose the scaling factor b arbitrarily, and setting $b = 1/r$ gives us (with f a scaling function)

$$G(r,t) = \xi_\perp(r) f(r/t^{1/z}) \tag{86}$$

which is the interesting equation for us. There is a connection between the correlation in space and that in time, $\xi_{||}(r,t) \sim t^{1/z}$ (which we recognise as the dynamical scaling, equation (2), from before). For scales $r \gg t^{1/z}$ we have a smooth surface — the roughening hasn't yet had time enough to affect the surface on this length-scale. G must then be independent of r, which means that the scaling function f must go as

$$f(y) \sim y^{-\alpha} \tag{87}$$

for large y, and that

$$G(r,t) \sim t^{\alpha/z} \sim \xi_{||}^\alpha \tag{88}$$

for $r \gg \xi_{||}(t)$. The dynamic exponent z is thus the measure of how fast the surface fluctuations spread, and of how the surface roughens in time. We see also that for small times and large r, the evolution of G is determined by the exponent $\beta = \alpha/z$.

We have seen that there are many functions that scale with α, z, and β, but we don't know anything about which functions that are easiest to measure in a simulation. It turns out that this depends quite a bit on what model we want to simulate, which is unfortunate. For the

KPZ-model (27), we have decided to measure both the variance of the height and the height difference correlation function defined by

$$G(r=400,t) \equiv |h(x,t) - h(x+400,t)\rangle_x \sim t^{beta} \quad (89)$$

for small t.

Both measurements give the same results for the exponents, and neither of them seem to give better accuracy than the other. The measurement of $G(400,t)$ is a little more efficient computationally, but not much.

6.2 Polynuclear growth

This subsection will touch briefly on the PNG-model, which is one of the classics in crystal growth (see [22] and references therein). The reason for its inclusion here is that it is a simple model for which it is easy to measure the exponents — it can thus serve as a good test for the programs used for the KPZ-model. A simple description of this model is that you randomly spray particles on a surface. Once a particle has stuck to the surface, it immediately begins to grow and form an island. The islands can also merge into larger masses of land. Mathematically, the evolution of h can be described by

$$h(i,t+1) = \max(h(i-1,t), h(i,t), h(i+1,t)) + \delta \quad (90)$$

where δ is one with probability p and zero otherwise.

The model (90) is interesting because there will be transitions for large p, but for small p at least the exponents are those of the KPZ-equation. Figure 3 shows the function $G(400,t)$ for a value of $p = 0.01$. The drawn curve corresponds to a value for β of $1/3$.

6.3 Results, at last

Now the time has finally come to determine if there is anything interesting in the KPZ-model. Unfortunately, the results previously obtained by Plischke et al[32], Fogedby et al[10], and Neergaard and den Nijs[27] are confirmed totally by these simulations: There is nothing exotic in the KPZ-model.

The expected result for the Edwards-Wilkinson equation is $z_{EW} = 2$ and $\beta_{EW} = \frac{1}{4}$, while that for the KPZ-equation is $z_{KPZ} = \frac{3}{2}$ and $\beta_{KPZ} = \frac{1}{3}$. It is of course highly unlikely that we would get exactly β_{KPZ} for small ϵ — we can't avoid the finite-size effects completely. It is the *trend* of the results that is important.

Table 4 shows a comparison of the results obtained here with those previously obtained by other methods. The accuracy is not very good, but we see that we move away from the EW-values for even as small ϵ's

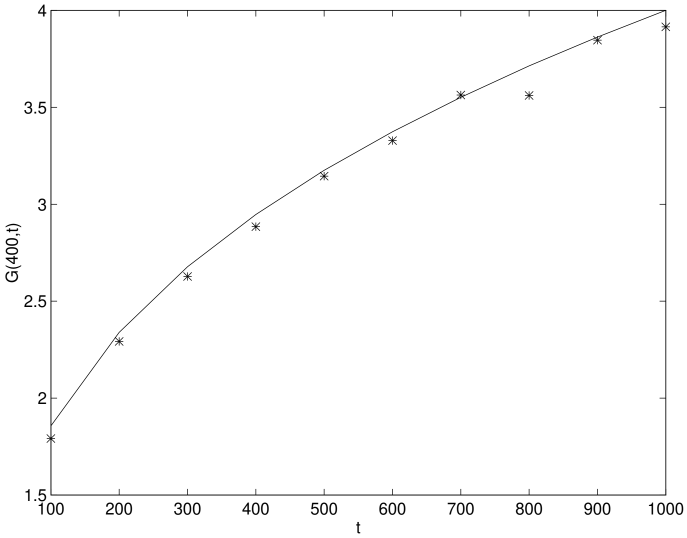

Figure 3: The height difference correlation function for the PNG model with $p = 0.01$. The data indicate a value for β of $\frac{1}{3}$.

	z-value obtained by		
ϵ	Plischke et al	Neergaard and den Nijs	Monte Carlo simulation
0.0	1.92	1.99	2
0.01			1.92
0.02			1.92
0.04			1.92
0.08			1.91
0.2		1.96	1.78
0.4		1.88	1.78
0.5	1.47		1.66
0.6		1.78	1.61
0.8		1.68	1.56
1.0	1.41	1.58	1.51

Table 4: The value of z obtained here and by others. Note that the values from the simulations cannot be compared directly to those obtained by Plischke et al and Neergaard and den Nijs. This is because finite-size effects are much more important in the simulations performed here. Also, they measure the exponent z directly, while we here measure β and thus suffer from error-propagation effects: $\Delta z = -\frac{\alpha}{\beta^2}\Delta\beta \approx \pm 0.05$. The trends of the other simulations are confirmed, though.

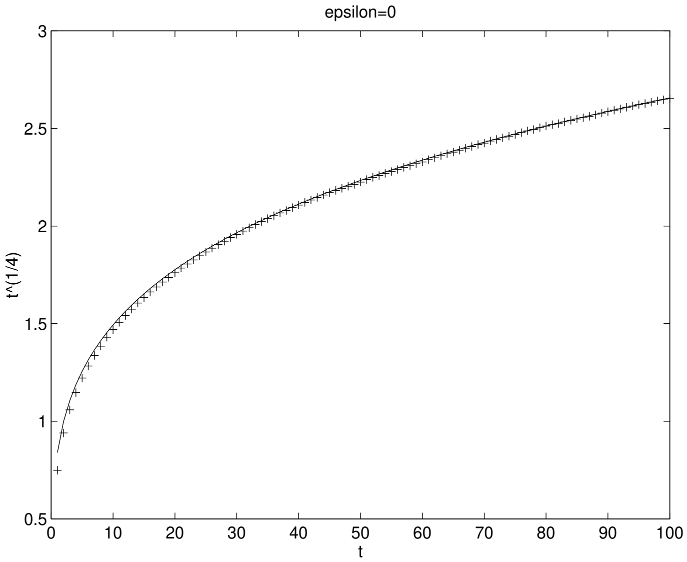

Figure 4: The height difference correlation function for the KPZ model with $\epsilon = 0$. The data indicate a value for β of $\frac{1}{4}$, compatible with the Edwards-Wilkinson equation.

as 0.01. (The same is true for ϵ even smaller, but here the results suffer heavily from finite-size effects.) The results from Plischke et al[32] are for a substrate size of 3072; they estimate that one of 50000 is needed before good convergence is reached. Neergaard and den Nijs[27] get their results with a size of only 17, while the simulations here use sizes of 5000. Figures 4 to 6 show the function $G(400, t)$ for some different values of ϵ. One of the difficulties with using these figures to get values for β is that it is not only the scaling function that changes with β, but also the range of validity for it (above it was shown that the scaling applies only for $t \ll l^z$). This means that there is one additional variable (the "cut-off" in time) that must be taken into account if one uses some sort of least-squares fit to obtain β. This effect can be seen clearly in figures (4) and (6) — the range of validity for the scaling decreases as z decreases.

"If it can't be expressed in figures, is is not science; it is opinion"
— Lazarus Long

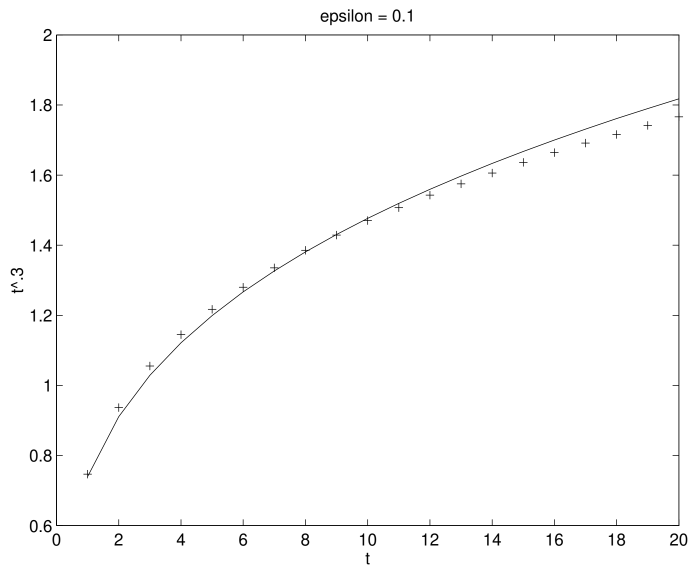

Figure 5: The height difference correlation function for the KPZ model with $\epsilon = 0.1$. The data indicate a value for β of $.3$, which is almost the KPZ-value.

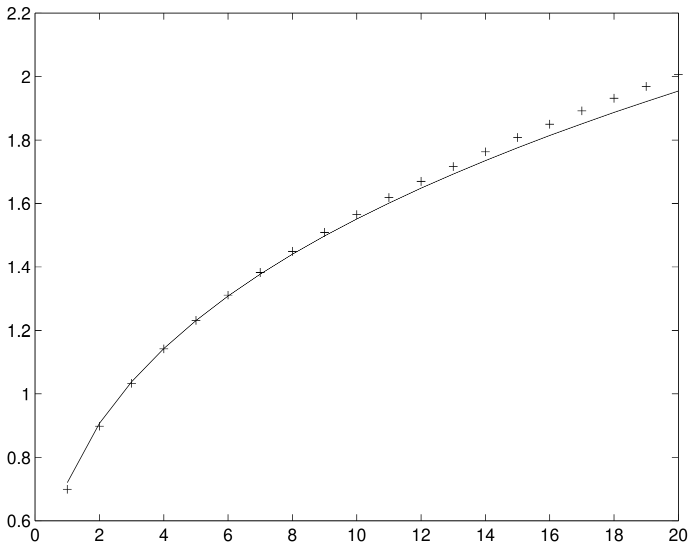

Figure 6: The height difference correlation function for the KPZ model with $\epsilon = 1$. The data indicate a value for β of $\frac{1}{3}$, compatible with the KPZ-equation.

7 Exercises for the Reader

As always, there is much more work left to do than has been done. It would be very interesting to use the methods from [27, 32] and see if the results would confirm the simple results obtained here also for small ϵ. They would also give much better accuracy for the exact values of the crossover-function. The down-side to these methods is that they require much more computing resources, particularly for small ϵ.

It would also be interesting, but probably very difficult, to try to extend the simulations to higher dimension, where there *is* a crossover for non-vanishing ϵ, and see if the results obtained theoretically by Kim[18] could be confirmed. This presents both theoretical and computational difficulties, see [28]. Another intriguing topic would be to look more at when a growth mechanism is self-affine and when it isn't[23]. In particular, the conclusions reached by López and Rodríguez[23] should be compared with the explanations for the large values of α briefly mentioned on page 34 — self-affinity might be a more rare phenomenon in growth than previously thought. Investigating Kardar's[16] suggestion that the noise η is correlated only over *long* distances would also be fun.

Last, a lot of time could be wasted trying to think of new Hamiltonians that could have chaotic renormalization flows. This could be done in two ways. Either by considering different terms directly in the Hamiltonian, and analysing these with e.g. MCRG, or inventing new discrete models to be simulated, and then trying to derive any interesting models that show up from a Hamiltonian.

"The time you enjoy wasting is not wasted time" — Bertrand Russel

Acknowledgements

It is impossible to list all those who deserve to be mentioned here. To all of you (friends, family, teachers, past and present): thankyou. Most of all, however, Henrik Johannesson must be thanked. He thought of the interesting idea to do an MS-thesis on the KPZ-equation, supported me when I didn't understand something and explained a lot of physics to me. The freedom (in deciding what to work on and when to do it) which he has given me during this project is also very much appreciated. Thanks must also go to Per Fröjdh for explaining the mass gap and its scaling to me.

A Monte Carlo Renormalization Group code

This appendix contains source-code in C++ for a Monte Carlo Renormalization of a square lattice. For simplicity, only two interactions (one even, one odd) are included, and only three generations of blocked spins are calculated. The modifications to more interactions and more generations are easy.

```
/* mcrg1int.cpp
** see Swendsen: PRB 20:5 (2079) p 2080
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define FALSE            0
#define TRUE            -1

/*  perform 3 renormalizations (ngen lattices needed),
**  w/ blocking factor bs, initial size size*size,
**  do equitime MC-sweeps to equilibriate, then
**  collect data iter times, doing nstep MC sweeps between each
**  measurement
*/

int     iter=30000;
const   size=108, nspin=size*size, bs=3, size1=size/bs,
        size2=size1/bs, size3=size2/bs, ngen=4;
int     lattice0[size][size], plus0[size], minus0[size], counter=0,
        lattice1[size1][size1], plus1[size1], minus1[size1],
        lattice2[size2][size2], plus2[size2], minus2[size2],
        lattice3[size3][size3], plus3[size3], minus3[size3];
const   equitime=3000, nstep=5;

double  h=0, K=.44, exp8, exp4;

void            init()
{
/*  init first generation lattice randomly,
**  also init plus/minus arrays to avoid modulo's below
*/

    for(int x=0;x<size;x++) {
        for(int y=0;y<size;y++)
            lattice0[x][y] = rand()%2 ? 1 : -1;
        plus0[x] = x+1; minus0[x] = x-1;
    }
    plus0[size-1] = 0; minus0[0] = size-1;

    for(int x=0;x<size1;x++) {
        plus1[x] = x+1; minus1[x] = x-1;
    }
    plus1[size1-1] = 0; minus1[0] = size1-1;
```

```
    for(int x=0;x<size2;x++) {
        plus2[x] = x+1; minus2[x] = x-1;
    }
    plus2[size2-1] = 0; minus2[0] = size2-1;

    for(int x=0;x<size3;x++) {
        plus3[x] = x+1; minus3[x] = x-1;
    }
    plus3[size3-1] = 0; minus3[0] = size3-1;
}


inline int      calc_mag(int i)
{
/*  calculate magnetisation of lattice i    */

    int         m=0;

    switch(i) {
        case 0:
        for(int x=0;x<size;x++)
            for(int y=0;y<size;y++)
                m += lattice0[x][y];
        break;
        case 1:
        for(int x=0;x<size1;x++)
            for(int y=0;y<size1;y++)
                m += lattice1[x][y];
        break;
        case 2:
        for(int x=0;x<size2;x++)
            for(int y=0;y<size2;y++)
                m += lattice2[x][y];
        break;
        case 3:
        for(int x=0;x<size3;x++)
            for(int y=0;y<size3;y++)
                m += lattice3[x][y];
        break;
    }
    return        m;
}


inline int      calc_nsum(int i)
{
/*  calculate nearest-neighbours interaction in lattice i*/

    int         sum=0;

    switch(i) {
        case 0:
        for(int x=0;x<size;x++)
            for(int y=0;y<size;y++)
                sum += lattice0[x][y] * (lattice0[x][plus0[y]] +  \
                                         lattice0[plus0[x]][y]);
```

```
        break;
        case 1:
        for(int x=0;x<size1;x++)
            for(int y=0;y<size1;y++)
                sum += lattice1[x][y] * (lattice1[x][plus1[y]] +  \
                                         lattice1[plus1[x]][y]);
        break;
        case 2:
        for(int x=0;x<size2;x++)
            for(int y=0;y<size2;y++)
                sum += lattice2[x][y] * (lattice2[x][plus2[y]] +  \
                                         lattice2[plus2[x]][y]);
        break;
        case 3:
        for(int x=0;x<size3;x++)
            for(int y=0;y<size3;y++)
                sum += lattice3[x][y] * (lattice3[x][plus3[y]] +  \
                                         lattice3[plus3[x]][y]);
        break;
    }
    return        sum;
}

inline int      flipcost(int x, int y)
{
/*  calculate cost of flipping spin (x,y) using hamiltonian
**  w/ only nearest-neighbour interaction
*/
    int         neighbours = lattice0[x][plus0[y]] + lattice0[plus0[x]][y] \
                          + lattice0[minus0[x]][y] + lattice0[x][minus0[y]];

    return      2*lattice0[x][y]*neighbours;
}

inline int      sign(int a)
{
    return      a<0 ? -1 : 1;
}

/*  the blockX functions implement the renormalization.
**  they return the value of spin (x,y) in lattice X
**  by forming the appropriate block-spin in lattice (X-1)
**  and using majority rule. (there are many spin-flips in lattice 0
**  between each use of the other lattices, so it's more efficient
**  to do it this way than to update all the lattice's after every
**  flip.)
*/

inline int      block1(int x, int y)
{
    int         sum=0;

    for(int i=0;i<bs;i++)
        for(int j=0;j<bs;j++)
            sum += lattice0[bs*x+i][bs*y+j];

    return      sign(sum);
```

```
}

inline int      block2(int x, int y)
{
    int         sum=0;

    for(int i=0;i<bs;i++)
        for(int j=0;j<bs;j++)
            sum += lattice1[bs*x+i][bs*y+j];

    return      sign(sum);
}

inline int      block3(int x, int y)
{
    int         sum=0;

    for(int i=0;i<bs;i++)
        for(int j=0;j<bs;j++)
            sum += lattice2[bs*x+i][bs*y+j];

    return      sign(sum);
}

inline void     do_blocking()
{
/*  do all the renormalizations, using blockX above */

    for(int x=0;x<size1;x++)
        for(int y=0;y<size1;y++)
            lattice1[x][y] = block1(x,y);

    for(int x=0;x<size2;x++)
        for(int y=0;y<size2;y++)
            lattice2[x][y] = block2(x,y);

    for(int x=0;x<size3;x++)
        for(int y=0;y<size3;y++)
            lattice3[x][y] = block3(x,y);
}

inline double   mrand()
{
/*  return pseudo-random number between 0 and 1 */
    return      double(rand())/double(RAND_MAX);
}


inline double   exptab(int en)
{
/* return precalculated boltzman-factors exp(-K*en)
** en == 2*spin*sum(neighbours) = 2*1*( 0 | 2 | 4) = 0 | 4 | 8
**/
    switch(en) {
        case 8:
            return exp8;
        case 4:
```

```
            return exp4;
        case 0:
            return 1.0;
        default:
            counter++;          /* this really shouldn't happen..*/
            return exp(-K*en);
    }
}

inline int      reallyflip(int energy)
{
    return      (energy<=0) || (mrand()<=exptab(energy));
}


inline void     montesweep()
{
/*  for all spins(in lattice0): flip spin if this lowers energy,
**  else flip w/ probability exp(-energychange)  (the metropolis algorithm)
*/

    for(int x=0;x<size;x++)
        for(int y=0;y<size;y++) {
            int         deltaE = flipcost(x,y);

            if(reallyflip(deltaE))
                lattice0[x][y] *= -1;
        }
}


inline void     equilibrate()
{
/*  make lattice0 "typical" for the given temperature (K-value) */

    for(int i=0;i<equitime;i++)
        montesweep();
}

/*  the interactions and the correlations necessary to get
**  the critical matrix. OPmixed[i] is the expectation value of
**  the product of OP in generations i and (i-1).
**  note that terms mixing even and odd interactions are zero
*/

int         odd[ngen], oddsquared[ngen], oddmixed[ngen];
int         even1[ngen], even1squared[ngen], even1mixed[ngen];

inline void     initdata()
{
    for(int i=0;i<ngen;i++) {
        odd[i] = oddsquared[i] = oddmixed[i] = 0;
        even1[i] = even1squared[i] = even1mixed[i] = 0;
    }
}

inline void     getdata()
```

```
{
/*  collect all the expectation values. (not all are necessary
**  for all generations)
**  odd[i] = 0 at the critical temperature
*/
    int         tempodd[ngen], tempeven1[ngen];

    for(int i=0;i<ngen;i++) {
        tempodd[i] = abs(calc_mag(i));
        tempeven1[i] = calc_nsum(i);
        even1[i] += tempeven1[i];
    }
    for(int i=1;i<ngen;i++) {
        oddsquared[i] += tempodd[i]*tempodd[i];
        oddmixed[i] += tempodd[i]*tempodd[i-1];

        even1squared[i] += tempeven1[i]*tempeven1[i];
        even1mixed[i] += tempeven1[i]*tempeven1[i-1];
    }
}

void            printdata()
{
/*  the critical matrix is just 2 numbers here (since only 1 odd
**  and 1 even interactions are included). otherwise, the
**  matrices A and B are calculated and M = inv(B) A
*/

    for(int i=1;i<ngen;i++) {
        /* odd[i] == 0 */
        double       corr1 = double(oddmixed[i])/iter;
        double       corr2 = double(oddsquared[i])/iter;

        printf("%%%% iteration %d\n", i);
        printf("odd: A=%g; B=%g;M=A/B=%g;\n", corr1, corr2, corr1/corr2);

        double corr3 = double(even1mixed[i])/iter - \
                       double(even1[i])/iter * double(even1[i-1])/iter;
        double corr4 = double(even1squared[i])/iter - \
                       double(even1[i])/iter * double(even1[i])/iter;

        printf("even: A=%g; B=%g;M=A/B=%g;\n", corr3, corr4, corr3/corr4);
    }
}

int             main(int ac, char **av)
{
/*  init data, do data collecting, print results, exit  */

    clock_t     t1, t2;

    iter = 30000;
    K = 0.440687;       // Kc from Table II in Swendsen

    if(ac==2)           // optional argument overrides iter above
        iter = atoi(av[1]);
```

```
    printf("%%%% iter %d K %g\n", iter, K);

    init();
    initdata();
    equilibrate();
    t1 = clock();

    exp8 = exp(-8*K);
    exp4 = exp(-4*K);
    for(int i=0;i<iter;i++) {
        for(int j=0;j<nstep;j++)
            montesweep();
        do_blocking();
        getdata();
    }


    t2 = clock();
    printf("%%%% time: %d, approx %g sec, %d measurements,"
             "%d considered spinflips\n",
             t2-t1, double(t2-t1)/CLOCKS_PER_SEC, iter, iter*nstep*nspin);
    if(counter != 0)
        printf("%%%% counter = %d\n", counter);
    printdata();
}
```

B Simulation code

This appendix contains code in C++ to simulate the growth model defined by equation (27), and measure the variance and the correlation function (89).

```
/* kpz.cpp */

#include <assert.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <stream.h>

#ifndef TRUE
#define TRUE -1
#define FALSE 0
#endif

/*  collect data for chain of size 5000, using 1000 different
**  realisations of the growth process
*/

const int        size=5000;
const int        nmeas = 1000;
int              *chain, *height, *plus, *minus;
double           *corr;
```

```
void            alloc_chain()
{
/*  allocate arrays for the spins and heights   */

    assert(chain = new int [size]);
    assert(plus = new int [size]);
    assert(minus = new int [size]);
    assert(height = new int [size]);
}

void            free_chain()
{
/*  return allocated memory */

    delete [] height;
    delete [] minus;
    delete [] plus;
    delete [] chain;
}

void            init_chain()
{
/*  start with a flat interface */

    for(int i=0;i<size;i++) {
        chain[i] = i&1 ? 1 : -1;
        height[i] = 0;
        plus[i] = i+1;
        minus[i] = i-1;
    }
    plus[size-1] = 0;
    minus[0] = size-1;
}

inline double   mrand()
{
    return      double(rand())/double(RAND_MAX);
}

inline void     evolve_chain(double epsilon)
{
/*  implements the growth rule. a random site is picked and
**  evaporation/adsorption takes place with certain probabilities
*/

    for(int n=0;n<size;n++) {
        int         i = rand() % size;
        double      prob = .5 * (1 + chain[i] * epsilon);

        if(mrand() <= prob) {
            int         foo = chain[i];
            chain[i] = chain[plus[i]];
            chain[plus[i]] = foo;
            height[i] = height[minus[i]] + (chain[i]+chain[minus[i]])/2;
            height[plus[i]] = height[i] + (chain[plus[i]]+chain[i])/2;
        }
```

```
    }
}


/* get_stat2() measures the variance of the height */

inline void     get_stat2(int tim)
{
    double      var=0, m=0, s=0;

    for(int i=0;i<size;i++) {
        m += height[i];
        s += height[i] * height[i];
    }
    m /= size; s/= size;
    var = sqrt(s - m*m);
    corr[tim] += var;
}


inline int      add(int dis, int ind)
{
    return      (ind+dis) % size;
}


/* get_stat1 measures the correlation function G(400) */

inline void     get_stat1(int tim)
{
    int g=0;
    const r = 10*40;
    for(int i=0;i<size;i++) {
        g += abs(height[i] - height[add(r, i)]);
    }
    corr[tim] += double(g)/size;
}

inline void     print_stat(int tim)
{
/*  output data collected for time tim
**  (this could be more than just one number)
**  NOTE: the data is averaged over nmeas different realisations
**  of the growth process
*/
    cout << corr[tim]/nmeas << ';' << endl;
}


int             main(int ac, char **av)
{
    alloc_chain();
    const       steps = 1000;   /* # time steps */
    const       interval = 1, tmeas = steps/interval;
    /*  interval = # of growth events between each measurement */

    const double    eps_tab[] = {.01, .02, .03, .04, .05};
```

```
    /*  data is collected for all epsilons in eps_tab in a single run */
    const       num_eps = sizeof(eps_tab)/sizeof(double);

    cout << "%% " << nmeas << " measurements" << endl;

    for(int nn=0; nn<num_eps; nn++) {
    /*  collect/print data for each epsilon separately. */

        double        epsilon = -eps_tab[nn];

        cout << "%%size " << size << ", steps " << steps;
        cout << ", epsilon " << epsilon << endl;
        cout << "%% tid i enheter av SIZE" << endl;
        cout << "%% tid = " << interval << ":" << interval;
        cout << ":" << steps << endl;

        /*  allocate and init data-array    */
        assert(corr = new double [tmeas]);
        for(int j=0;j<tmeas;j++)
            corr[j] = 0.0;

        for(int n=0;n<nmeas;n++) {
        /*  for each epsilon, nmeas different interfaces are
            grown and data averaged over these */

            init_chain();   /* start with flat interface */
            for(int tid=0; tid<steps;tid+=interval) {
            /*  evolve interface but only collect data every interval
                time step
            */
                for(int i=0;i<interval;i++)
                    evolve_chain(epsilon);
                get_stat1((tid)/interval);
            }
        }

        cout << "%% tid   G(400, t)" << endl;
        cout << 'm' << " = [" << endl;
        for(int tid=0;tid<steps;tid+=interval) {
            cout << tid+interval << '\t';   /* print time and data */
            print_stat((tid)/interval);
        }
        cout << "];" << endl;
        cout << "t = m" << "(:,1);" << endl;
        cout << "gt = m" << "(:,2);" << endl;
        delete [] corr;
        cout << endl << endl;
    }

    free_chain();       /* free allocated memory    */
}
```

C A Java growth program

This appendix contains code in Java to visually simulate the evolution of the growth model (27). It can be run on any Java-supporting platform, most easily by using the program `appletviewer` from the JDK.

Assuming that the code is in a file `kpz.java` and that the following text is present in a file `kpz.html`

```
<html>
<head>
<title> KPZ Applet</title>
</head>
<body>
<applet code="kpz.class" width=800 height=300></applet>
</body>
</html>
```

the applet can be run by first compiling the Java code (`javac kpz.java`) and then running appletviewer (`appletviewer kpz.html`) or any other Java-compatible browser.

```
// kpz.java

import java.awt.*;
import java.util.Random;

public class kpz extends java.applet.Applet implements Runnable {

    Thread      mythread = null;        // mythread "is" the applet

    Button      Restart  = new Button("Restart"); // user interface buttons
    Button      Pause  = new Button("Pause");
    Button      Stop = new Button("Stop");
    Button      Single_step = new Button("Single Step");
    Choice      C = new Choice();

    Random      rn = new Random();

    int         wx, wy, siz;        // "physical" variables
    int         plus[], minus[], Height[];
    double      epsilon;

    boolean     kludge=false; // kludge is used (in a really ugly way) to
                            // clear screen after Restart-button is pressed
    public void init() {
        wx = size().width;          // init the window
        wy = size().height;
        resize(wx,wy);
        setBackground(Color.white);
        siz = wx;

        plus = new int [siz];       // init the physics
        minus = new int [siz];
```

```
    Height = new int [siz];

    for(int i=0;i<siz;i++) {
        plus[i] = i+1; minus[i] = i-1;
    }
    plus[siz-1] = 0; minus[0] = siz-1;

    for(int i=0;i<siz;i++)      // start w/ almost flat interface
        Height[i] = (i&1)==0 ? 1 : 2;   // remember single-step rule!


    Panel p = new Panel();      // init the look of the user interface
    // FlowLayout() handles placement of buttons etc in the panel
    p.setLayout(new FlowLayout());
    // add buttons to panel
    p.add(Pause); p.add(Restart); p.add(Stop); p.add(Single_step);

    // add "menuitems" to C
    C.addItem("Epsilon = 0.0"); C.addItem("Epsilon = 0.1");
    C.addItem("Epsilon = 0.2"); C.addItem("Epsilon = 0.3");
    C.addItem("Epsilon = 0.4"); C.addItem("Epsilon = 0.5");
    C.addItem("Epsilon = 0.6"); C.addItem("Epsilon = 0.7");
    C.addItem("Epsilon = 0.8"); C.addItem("Epsilon = 0.9");
    C.addItem("Epsilon = 1.0");
    C.select("Epsilon = 1.0");
    epsilon = 1.;
    p.add(C);           // add "menu" to panel

    add("North",p);     // add panel to window
    Restart.disable();
    C.disable();
    kludge=false;
}   // init()

public void paint(Graphics g) {
    // paint() is called whenever the window needs repainting
    int         max=Height[0], min=Height[0];

    if(kludge) {
        g.setColor(Color.white);    // clear screen
        g.fillRect(0,0,wx,wy);
        kludge=false;
    }

    for(int i=1;i<siz;i++)      // find max/min height
        if(Height[i] > max)
            max = Height[i];
        else if(Height[i] < min)
            min = Height[i];

    if(3*max > wy-20-50) {  // has interface reached "top" of window?
        for(int i=0;i<siz;i++)  // if so, adjust viewing so that
            Height[i] -= min;   // growth continues from bottom
        max -= min; min = 0;    // of window

        g.setColor(Color.white);  // and clear screen
        g.fillRect(0,0,wx,wy);
```

```
    }

    g.setColor(Color.black);

    for(int i=0;i<siz;i++)      // draw the interface
        g.drawLine(i, wy, i, wy - 3*Height[i]-20);
} // paint()

// also override update() => no flickering
// (inherited update() does fillRect() + paint())
public void update(Graphics g) {
    paint(g);
} // update()

public void run() {     // called whenever applet should run
    while (mythread != null){
        step();             // evolve interface
        repaint();          // and paint it
        try {
            Thread.sleep(180);      // pause briefly
        } catch (InterruptedException e) {
            break;          // any exception => exit program
        }
    }
}   // run()

public void start() {   // called whenever applet starts
    if (mythread == null) {
        mythread = new Thread(this);
        mythread.start();
    }
}   // start()

public void stop() {    // stop applet
    if (mythread != null) {
        mythread.stop();
        mythread = null;
    }
}   // stop()

public boolean mouseDown(java.awt.Event evt, int x, int y) {
    requestFocus();         // handle mouse-clicks
    return true;
}   // mousedown()

public boolean action(Event evt, Object arg) {
// handle events -- choices or clicks on buttons
    if (evt.target instanceof Choice) {
        if("Epsilon = 0.0".equals(arg))
            epsilon = 0;
        else if("Epsilon = 0.1".equals(arg))
            epsilon = .1;
        else if("Epsilon = 0.2".equals(arg))
            epsilon = .2;
        else if("Epsilon = 0.3".equals(arg))
            epsilon = .3;
        else if("Epsilon = 0.4".equals(arg))
```

```
            epsilon = .4;
        else if("Epsilon = 0.5".equals(arg))
            epsilon = .5;
        else if("Epsilon = 0.6".equals(arg))
            epsilon = .6;
        else if("Epsilon = 0.7".equals(arg))
            epsilon = .7;
        else if("Epsilon = 0.8".equals(arg))
            epsilon = .8;
        else if("Epsilon = 0.9".equals(arg))
            epsilon = .9;
        else if("Epsilon = 1.0".equals(arg))
            epsilon = 1.;
    } else {
        if ("Stop".equals(arg)) {   // check for button labels,
            stop();                 // do action, and then disable/
            Stop.disable();         // enable buttons according
            Restart.enable();       // to which state action took
            C.enable();             // us to
            Pause.disable();
        } else if ("Restart".equals(arg)) {
            Stop.enable();
            Restart.disable();
            C.disable();
            Pause.enable();
            Pause.setLabel("Pause");
            for(int aa=0;aa<siz;aa++)   // start w/ "flat" interface
                Height[aa] = (aa&1)==0 ? 1 : 2;
            kludge=true;    // this is an ugly kludge to clear the screen
            start();
        } else if ("Pause".equals(arg)) {
            stop();
            Pause.setLabel("Resume");
        } else if ("Resume".equals(arg)) {
            start();
            Pause.setLabel("Pause");
        } else if("Single Step".equals(arg)) {
            stop();
            Pause.setLabel("Resume");
            step();
            repaint();
        }
    }
    return true;
}   // action()

public void step() {        // evolve Height[] one time-step
    for(int i=0;i<siz;i++) {
        int      x = rn.nextInt() % siz;
        if(x<0)
            x = -x;

        double      evaporation = .5*(1-epsilon);
        double      deposition = .5*(1+epsilon);

        if((Height[minus[x]]>Height[x])&&(Height[plus[x]]>Height[x])) {
            if(rn.nextDouble() <= deposition)   // minimum
```

```
                    Height[x] += 2;
            } else if((Height[minus[x]]<Height[x])&&(Height[plus[x]]<Height[x])) {
                if(rn.nextDouble() <= evaporation)  // maximum
                    Height[x] -= 2;
            }
        }

    }   // step()

}   // kpz
```

References

[1] A. Berera and L.-Z. Fang. Stochastic fluctuations and structure formation in the universe. *Phys. Rev. Lett.*, 72(4):458, 1994.

[2] A. N. Berker and S. R. McKay. Hierarchical models and chaotic spin glasses. *J. Stat. Phys.*, 36(5):787, 1984.

[3] K. Binder, editor. *The Monte Carlo Method in Condensed Matter Physics.* Topics in Applied Physics. Springer-Verlag, 1995.

[4] J. Binney, N. Dowrick, A. Fisher, and M. Newman. *The Theory of Critical Phenomena.* Oxford University Press, 1992.

[5] P. H. Damgaard. Stability and instability of renormalization group flows. *Int. J. Mod. Phys. A*, 7(27):6933, 1992.

[6] B. Derrida, J.-P. Eckmann, and A. Erzan. Renormalization groups with periodic and aperiodic orbits. *J. Phys. A*, 16:893, 1983.

[7] A. Erzan, L. Pietronero, and A. Vespignani. The fixed-scale transformation approach to fractal growth. *Rev. Mod. Phys.*, 67(3):545, 1995.

[8] F. Family. Dynamic scaling and phase transitions in interface growth. *Physica A*, 168:561, 1990.

[9] F. Family and T. Vicsek. Scaling of the active zone in the Eden process on percolation networks and the ballistic deposition model. *J. Phys. A*, 18(2):L75, 1985.

[10] H. C. Fogedby, A. B. Eriksson, and L. V. Mikheev. Continuum limit, Galilean invariance, and solitons in the quantum equivalent of the noisy Burgers equation. *Phys. Rev. Lett.*, 75(10):1883, 1995. Also available as cond-mat/9503028.

[11] C. Godrèche, editor. *Solids far from Equilibrium.* Cambridge University Press, 1991.

[12] L. Golubović and Z. Wang. Kardar-Parisi-Zhang model and anomalous elasticity of two- and three-dimensional smectic-A liquid crystals. *Phys. Rev. E*, 49(4):2567, 1994.

[13] L.-H. Gwa and H. Spohn. Bethe solution for the dynamical-scaling exponent of the noisy Burgers equation. *Phys. Rev. A*, 46(2):844, 1992.

[14] R. A. Heinlein. *Stranger in a Strange Land.* G.P. Putnam's, 1961.

[15] G. L. M. K. S. Kahanda, X. qun Zou, R. Farrel, and P. zen Wong. Columnar growth and kinetic roughening in electrochemical deposition. *Phys. Rev. Lett.*, 68(25):3741, 1992.

[16] M. Kardar. Dynamic scaling phenomena in growth processes. Preprint cond-mat/9507019.

[17] A. Kashuba. *Phys. Rev. Lett.*, 73(16):2264, 1994.

[18] D. Kim. Bethe ansatz solution for crossover scaling functions of the asymmetric *xxz* chain and the Kardar-Parisi-Zhang-type growth model. *Phys. Rev. E*, 52(4):3512, 1995.

[19] E. Kolb and M. S. Turner. *The Early Universe*. Addison-Wesley, 1989.

[20] J. Krim, I. Heyvaert, C. V. Haesendonck, and Y. Bruynseraede. Scanning tunneling microscopy observation of self-affine fractal roughness in ion-bombarded film surfaces. *Phys. Rev. Lett.*, 70(1):57, 1993.

[21] J. Krug and H. Spohn. Universality classes for deterministic surface growth. *Phys. Rev. A*, 38(8):4271, 1988.

[22] J. Krug and H. Spohn. Kinetic roughening of growing surfaces. In C. Godrèche, editor, *Solids far from Equilibrium*. Cambridge University Press, 1991.

[23] J. M. López and M. A. Rodríguez. Lack of self-affinity and anomalous roughening in growth processes. Preprint cond-mat/9603180.

[24] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W H Freeman, 1982.

[25] M. Marsili, A. Maritan, F. Toigo, and J. R. Banavar. Stochastic growth equations and reparametrization invariance. Preprint cond-mat/9606026.

[26] S. Matsuura and S. Miyazima. Self-affine fractal growth front of *aspergillus oryzae*. *Physica A*, 191:30, 1992.

[27] J. Neergaard and M. den Nijs. Crossover scaling functions in one dimensional dynamic growth models. *Phys. Rev. Lett.*, 74(5):730, 1995. Also available as cond-mat/9406086.

[28] T. Newman and A. Bray. Strong-coupling behaviour in discrete Kardar-Parisi-Zhang equations. Preprint cond-mat/9604071.

[29] J. D. Pelletier. Analysis and modeling of scale-invariance in plankton abundance. Preprint ao-sci/9605002.

[30] J. D. Pelletier. Kardar-Parizi-Zhang model for the fractal structure of cumulus cloud fields. Preprint ao-sci/9605001.

[31] M. Plischke and B. Bergersen. *Equilibrium Statistical Physics*. World Scientific Press, 1994.

[32] M. Plischke, Z. Rácz, and D. Liu. Time-reversal invariance and universality of two-dimensional growth models. *Phys. Rev. B*, 35(7):3485, 1987.

[33] R. Rajaraman. *Solitons and Instantons*. North-Holland, 1981.

[34] M. Rubio, C. Edwards, A. Dougherty, and J. Gollub. Self-affine fractal interfaces from immiscible displacement in porous media. *Phys. Rev. Lett.*, 63(16):1685, 1989.

[35] A. Sjölander. *Irreversible Statistical Physics*. Chalmers, 1990.

[36] C. F. Stevens. *The Six Core Theories of Modern Physics*. The MIT Press, 1995.

[37] R. H. Swendsen. Monte Carlo renormalization-group studies of the $d = 2$ Ising model. *Phys. Rev. B*, 20(5):2080, 1979.

[38] B. A. Veytsman. Limit cycles in renormalization group flows: thermodynamics controls dances of space patterns. *Phys. Lett. A*, 183:315, 1993.

[39] T. Vicsek, M. Cserzö, and V. K. Horváth. Self-affine growth of bacterial colonies. *Physica A*, 167:315, 1990.

[40] A. Weinrib and B. I. Halperin. Critical phenomena in systems with long-range-correlated quenched disorder. *Phys. Rev. B*, 27(1):413, 1983.

[41] K. G. Wilson. *Phys. Rev. B*, 4:3174, 1971.

[42] J. Yeomans. *Statistical Mechanics of Phase Transitions*. Oxford University Press, 1992.

[43] A. Zangwill. *Physics at Surfaces*. Cambridge University Press, 1988.

[44] J. Zhang, Y.-C. Zhang, P. Alström, and M. Levinsen. Modeling forest fire by a paper-burning experiment, a realization of the interface growth mechanism. *Physica A*, 191:383, 1992.