

Critical exponents of the KPZ equation via multi-surface coding numerical simulations

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2000 J. Phys. A: Math. Gen. 33 8181

(<http://iopscience.iop.org/0305-4470/33/46/303>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 171.66.16.123

The article was downloaded on 02/06/2010 at 08:35

Please note that [terms and conditions apply](#).

Critical exponents of the KPZ equation via multi-surface coding numerical simulations

Enzo Marinari[†], Andrea Pagnani[‡] and Giorgio Parisi[†]

[†] Dipartimento di Fisica, INFN and INFN, Università di Roma ‘La Sapienza’, P A Moro 2, 00185 Roma, Italy

[‡] Dipartimento di Fisica and INFN, Università di Roma ‘Tor Vergata’, Viale della Ricerca Scientifica 1, 00133 Roma, Italy

E-mail: Enzo.Marinari@roma1.infn.it, Andrea.Pagnani@roma2.infn.it and Giorgio.Parisi@roma1.infn.it

Received 12 May 2000, in final form 17 October 2000

Abstract. We study the KPZ equation (in $D = 2, 3$ and 4 spatial dimensions) by using a restricted solid-on-solid discretization of the surface. We measure the critical exponents very precisely, and we show that the rational guess is not appropriate, and that $D = 4$ is not the upper critical dimension. We are also able to determine very precisely the exponent of the sub-leading scaling corrections, that turns out to be close to unity in all cases. We introduce and use a *multi-surface coding* technique, that allows a gain of the order of 30-fold over usual numerical simulations.

1. Introduction

The KPZ equation [1], in its apparent simplicity, involves many issues that need clarification. The continuum equation (which describes the local growth of an interface profile) is

$$\frac{\partial h}{\partial t} = \nu \bar{\nabla}^2 h + \frac{\lambda}{2} (\bar{\nabla} h)^2 + \eta(\vec{r}, t). \quad (1)$$

The real behaviour of this equation is not well understood. The absence of a complete mean field theory does not help, and the fact that we have to understand the effects of a strong-coupling fixed point makes the development of a perturbative renormalization approach very difficult.

A practical approach for numerical studies of the problem is to consider lattice restricted solid-on-solid (RSOS) surfaces (see e.g. [2] for a review): one considers a discretized height field on a D -dimensional lattice, and imposes the constraint that the absolute value of the distance between two neighbouring surface elements can only take the values zero and one.

Previously performed extensive numerical simulations [3, 4] do not clarify the situation completely. After a number of interesting field theoretical results [5], the introduction of new renormalization group based techniques is a potentially promising direction of development [6].

The main theoretical points which still deserve proper explanation are twofold: whether equation (1) has a finite upper critical dimension ($D_>$) or not, which is still not clear, and the exact quantification of the related critical exponents. Regarding arguments in favour of $D_> = 4$ we address the reader to [7], while arguments supporting $D_> = \infty$ can be found in [6, 8].

In this light we introduce here a new numerical technique and present some precise numerical simulations that allow us to estimate critical exponents. Thanks to the new, precise technique we succeed in clarifying some questions: we show, for example, that a conjecture of [9] is not founded, and we give quantitative estimates about the behaviour of sub-leading corrections.

2. The multi-surface coding

The precise results on which this paper is based are due to a new technique we introduce to simulate RSOS surfaces, where adjacent elements of the surface can only be at a distance of 1, 0 or -1 lattice spacings. The technique is a generalization of the so-called *multi-spin coding* technique (well discussed by Rieger [10] in the context of the study of disordered systems), where, by using the fact that the ± 1 spins can be represented by Boolean logical variables, one stores 64 copies of the system in a single computer word (we will assume in the following we are using a 64-bit computer). The main idea is that we can simulate, with basically the same cost as a normal single simulation, 64 copies of the system, by rewriting the basic operations (like summing spins for computing the effective force) as Boolean operations, and by exploiting the fact that when, for example, the computer is calculating an AND logical bit, it is indeed doing that 64 times at once. The gain of such an approach over a usual single-spin simulation is of the order of 30-fold: one gains a factor of 64 for the number of systems that updates at once, and loses a factor of the order of two in computational complexity (adding the spin in the Boolean form takes more time [10]).

We generalize here the Ising case to the case of a *field of differences*, that can have three values, since a given element of the surface can be at the same height as a given neighbour, or one step behind it, or one step ahead of it. The method is new as applied to such a system, where one does not have to compute the value of an energy, but uses the Boolean operations to determine whether the element can be moved without violating the geometrical constraint.

Let us think for simplicity about $2D$, where we have a $2D$ support where a height field (the surface height) $h_i = h_{x,y}$ is defined. One surface element h_i has $2D = 4$ first neighbours: in the RSOS model the difference $\Delta_{i,\mu} \equiv h_i - h_{i\pm\hat{\mu}}$, where $\mu = x, y$, can only take the three values, $-1, 0$ and $+1$. We store the DL^2 values of the $\Delta_{i,\mu}$ (and, as we will discuss further, each $\Delta_{i,\mu}$ needs two bits to be stored. This is a redundant way to store the information, since a factor of two in memory could be easily saved, but it is very convenient from the point of view of computer time: as usual one trades memory for time, and avoids a cumbersome reconstruction by storing more information).

If a given element is behind its neighbour the difference is -1 , if it has the same height as the neighbour the difference is 0 , while if it is ahead of the neighbour the difference is $+1$. Since we have three allowed values we can represent each of these differences with two bits (that we will call respectively H, high bit, and L, low bit): we can, for example, code with 00 the situation where the given element is behind its neighbour, with 01 the situation where they have the same height and with 10 the situation where it is ahead (the value 11 is forbidden).

If the element is ahead of even a single one of the $2D$ neighbours the move is forbidden (it would violate the RSOS constraint). It is clear this is easily implemented in our coding: one just needs to perform a logical OR of the $2D$ H bits related to the site i , and if it is unity at least one of the neighbours is behind and the move is not allowed. Let us get a better understanding of why this is so. We start comparing our element i to the first of its $2D = 4$ neighbours (let us say the one in the positive direction $+1$): if i is ahead of the neighbour the relevant H bit (that we have stored in $\Delta_{i,+1}$) is unity, and we already know we cannot move. In contrast if i is at the same height as the neighbour or behind it we have that, as far as this neighbour

is concerned, the element i can advance by one unit. Now we look at the second neighbour, where the same reasoning holds: if we look at the logical OR of the two relevant high bits we will find that we cannot move if this quantity is unity. Looking at all the $2D$ neighbours we see that the move is forbidden if the logical OR of the $2D$ H bit is unity: clearly this operation, as for all the others present in the core of the code, updates with a single computer cycle the 64 copies of the system. After doing that, if the element i cannot be moved nothing has been done: if it is moved we have to update the $\Delta_{i,\mu}$ related to the site i and to the neighbours, to describe the new situation.

The codes simulating systems in a different number of spatial dimensions (in our case from $D = 2-4$) are simple generalizations of each other (when adding a dimension one has to add checks on the new neighbours and their update).

As usual in these kinds of simulation the random noise is implemented with a random choice of the site to be updated. We also implement a 50% probability of really updating a surface element that according to the RSOS constraint could be updated. This is very important, since starting from random independent surfaces is not enough: because of our parallel scheme the sites of our 64 copies have to be updated in the same order, and such an updating algorithm with unit updating probability is attractive, and the 64 configurations asymptotically at large times become equal [11]. The probability of not accepting an allowed change, that depends on each of the 64 configurations, solves this problem.

We are aware of a parallel algorithm to update surfaces [12]: it is very different in spirit from our algorithm (since it parallelizes on different sites of the lattice). We have not compared in detail the performances of the two algorithms, but we believe that on one hand the algorithm of [12] is more general, and not limited to RSOS models, but on the other hand our algorithm is more regular (there are no exceptional loops), and it probably performs better for the model we study.

3. The numerical simulation

We have based this paper on numerical simulations of lattices of volume $V \equiv L^D$. The spatial dimensionality D is the spatial support, where a one-dimensional surface takes values. We study the $D = 2, 3$ and 4 cases. At a given time t of the dynamical evolution the position of the surface can be expressed by the values $h_i(t)$ (that we reconstruct by the differences we store in our code, see the former section) where i represents, in lexicographic order, D numbers labelling the spatial sites.

We consider a dynamics which generates a RSOS growth: distances of first neighbouring elements of the surface cannot be larger than unity. At each trial step we move elements of the surface that are not constrained not to do so because of the RSOS restriction with probability $\frac{1}{2}$: we have to use a probability different from unity in order to keep independent the different surfaces we simulate in the same computer word.

We consider a large number of different lattice sizes. In $D = 2$ we take L going from 5 to 641, in $D = 3$ we consider L going from 5 to 103 while in $D = 4$, L goes from 5 to 28. All our data have been averaged over 64 different dynamical runs.

Because of the way we determine critical exponents, by trying to measure precisely the asymptotic time behaviour, we use very long runs, and we always try to check that we have reached the asymptotic plateau in a clear way (see the discussion and the figures in the next section). Let t be the time labelling sweeps of our simulation (we define a sweep as the trial update of V random sites). In any of our simulations we run T updating sweeps: we give in table 1 the number of full lattice sweeps for each run on different lattice sizes and numbers of dimensions. We measure the observable every 1000 lattice sweeps: when analysing the

Table 1. Number of full lattice sweeps for each run on different lattice sizes and numbers of dimensions.

$L(2D)$	$T(2D)$	$L(3D)$	$T(3D)$	$L(4D)$	$T(4D)$
5	7.50002e+06	11	7.5e+06	7	786 450
41	7.50002e+06	13	7.5e+06	11	1.2e+06
79	6.00002e+06	17	7.5e+06	13	1.2e+06
157	6.891 e+06	19	7.5e+06	15	1.2e+06
317	1.647 5e+06	23	7.5e+06	17	898 050
641	233 070	25	3.749 05e+06	20	510 150
		33	1.454 2e+06	22	523 650
		37	1.042 65e+06	25	786 450
		61	484 350	28	456 150
		83	696 600		
		103	409 050		

large-time asymptotic behaviour of the system we discard the first half sweeps (we call T_0 the time of our first measurement): this is a very conservative attitude, but we prefer to be safe in not having any systematic bias by paying the price of perhaps making the statistical error 10% larger than the best we could do.

We define the time-dependent observables

$$\overline{h(t)} \equiv \frac{1}{V} \sum_{i=1}^V [h_i(t)] \quad (2)$$

and

$$w_k(t) \equiv \frac{1}{V} \sum_{i=1}^V [(h_i(t) - \overline{h(t)})^k] \quad (3)$$

that we compute for $k = 2, 3$ and 4 and for different D and L values (when needed we will label w with the superscript (L), to make clear to which lattice size we are referring). We define the large-time asymptotic limit of $w_k(t)$ as

$$w_k^{(L)} \equiv \frac{1}{T - T_0 + 1} \sum_{t=T_0}^T w_k(t) \quad (4)$$

for simulation on a lattice of linear size L . We always check (this is one of the crucial points of this paper, which we will discuss in more detail in the next section) that T_0 and T are large enough to make our result unbiased in the precision of our statistical error.

4. Analysis

We will discuss here the analysis of our numerical data. We want to determine the critical exponents of the asymptotic behaviour of the w_k we have defined in the former section. For example, we have that the asymptotic infinite-time value of w_2 has a leading scaling behaviour

$$w_2^{(L)} = L^{2\chi} \quad (5)$$

while at intermediate times (large enough to be in the scaling region but small enough not to feel the finite size of the lattice)

$$w_2(t) = t^{\frac{2\chi}{z}}. \quad (6)$$

The first behaviour is obtained by taking large times on different lattice sizes, and by studying the asymptotic time value as a function of L . The second behaviour is studied by simulating large lattices, and analysing the behaviour of the systems for times larger than unity, but very much smaller than the thermalization time (at the given value of the lattice size). An exact (Galilean) invariance of the KPZ equation implies that $z + \chi = 2$. Deciding whether it is better to measure accurately z or χ is a practical matter.

As opposed to the choice, for example, of [4], here we have mainly based our analysis on fitting the behaviour of the large-time asymptotic value $w_k^{(L)}$, and we have only used fits to the intermediate-time behaviour to substantiate our results. In fact we believe that it is very difficult to determine a precise quantitative estimate of the exponent of the time scaling. The problem with the time-dependent behaviour of $w_k(t)$ is that, in order to obtain an unbiased value, one needs a double cutoff, both at small and large times. At small times the behaviour of $w_k(t)$ is not a pure power, and one has to discard a small lattice, and/or to use corrections to scaling, in order to remove this effect. At large times one starts to feel the finiteness of the lattice system, and a new crossover (toward the asymptotic, constant-time behaviour) intervenes. In other words a careful analysis of the time exponent needs a double sliding window, moving both at small and large times. We also find that the crossover effects at large times are very important: even on large lattices one can soon see systematic effects on the exponent estimate due to the finiteness of the lattice.

In contrast, the asymptotic time behaviour only needs one cutoff, that excludes small times, where the asymptotic value has not yet been reached. This is easy, and we do it by using a logarithmic division of our data. In this way we are able to check with very high precision that we are computing an unbiased (effective, L -dependent) exponent. Again, we will also show results obtained from a direct fit of z , to show they are consistent with the χ values we determine.

Our main analysis is made by fitting at the same time the three sets of data [13]:

$$\begin{aligned} w_2 &\simeq A_2 L^{2\chi} (1 + B_2 L^{-\omega}) \\ w_3 &\simeq -A_3 L^{3\chi} (1 + B_3 L^{-\omega}) \\ w_4 &\simeq A_4 L^{4\chi} (1 + B_4 L^{-\omega}). \end{aligned} \quad (7)$$

We always compare this fit to the best fit of w_2 alone (typically by only including the large volumes and by ignoring scaling corrections) and check that things are coherent. We also check independently that, for example, w_3 really scales as $w_2^{\frac{3}{2}}$: this is clearly true for our data. Without the use of our whole set of data defined in (7) we would not have been able to determine ω with a reasonable statistical precision.

With this definition the exponent χ is the same as χ of [4] and as α of [6] (our definition of dimensionality of the system excludes the time dimension, and is always only the dimension of the space).

The error analysis is made by using a jack-knife approach [14]: we divide our statistical sample into ten parts all including all of the data but one-tenth (each part excludes a different tenth of the data), we fit ten times the behaviour of, for example, (7), and compute the error on A_k , B_k , χ and ω by the fluctuations of the ten results (multiplying the error by a factor of ten, due to the fact that the individual parts we have formed are correlated [14]). In short, we present a reliable estimate of the statistical errors over the quantities we determine.

The first point we have to discuss is the exponents of finite-size corrections that appear in equation (7). Do the corrections to even and odd momenta really scale with the same exponent? This has been questioned in [15], and we provide here accurate evidence that $\omega_2 = \omega_3 = \omega_4 \equiv \omega$ in our model.

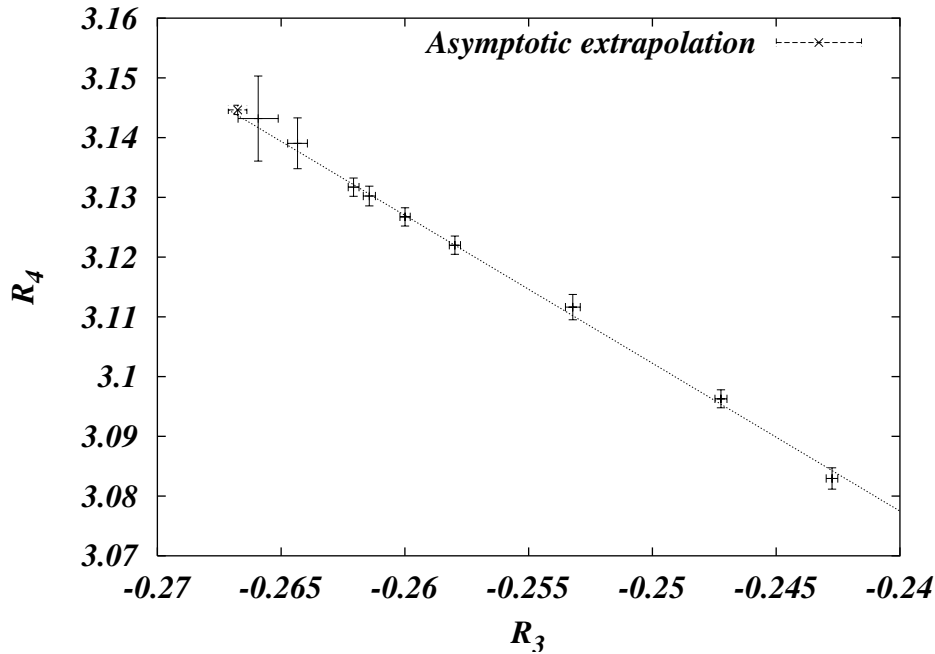


Figure 1. R_4 versus R_3 , for $D = 2$.

Let us start by arguing that we are not in a situation in which the scaling exponent of the odd moments, ω_{odd} is *smaller* than the scaling exponent of the even moments, ω_{even} (this is the opposite scenario to the one proposed in [15]).

Following [15] we define

$$R_3 \equiv \frac{w_3}{w_2^{3/2}} \quad R_4 \equiv \frac{w_4}{w_2^2}. \quad (8)$$

The exponent χ disappears from these ratios, and, in general circumstances, if $\omega_{\text{odd}} < \omega_{\text{even}}$, one has that asymptotically for large L , ignoring sub-leading corrections,

$$R_3 \simeq c_3 + d_3 t^{\omega_{\text{odd}}} \quad R_4 \simeq c_4 + d_4 t^{\omega_{\text{even}}} \quad (9)$$

i.e.

$$R_4 \simeq c + d (R_3 + c)^{\omega_{\text{even}}/\omega_{\text{odd}}} \quad (10)$$

and R_3 is a linear function of R_4 if and only if $\omega_{\text{odd}} \geq \omega_{\text{even}}$ (in which case the two ratios asymptotically scale with the same exponent). We plot in figure 1 R_4 versus R_3 , and notice that the linearity is impressive. In these data there is no sign of a discrepancy among scaling exponents of odd and even momenta, and they surely exclude that $\omega_{\text{odd}} < \omega_{\text{even}}$. The last point on the left in the figure is our asymptotic extrapolation, with the (small) statistical error attached (the best estimate of [15] is $R_3 = -0.27 \pm 0.01$ and $R_4 = 3.15 \pm 0.02$, well compatible with our data but with a very much larger error).

We now use figure 2 to also exclude the case $\omega_{\text{odd}} > \omega_{\text{even}}$, establishing in this way that for our model $\omega_{\text{odd}} = \omega_{\text{even}}$: in figure 2 we plot the effective scaling exponent obtained by using separately the data for $w_2(L)$, $w_3(L)$ and $w_4(L)$. The three quantities do all depend linearly, with very good approximation, on $\frac{1}{L}$, showing that we have the same exponent of the scaling corrections. Again, the impressive linearity of the data implies that we are measuring

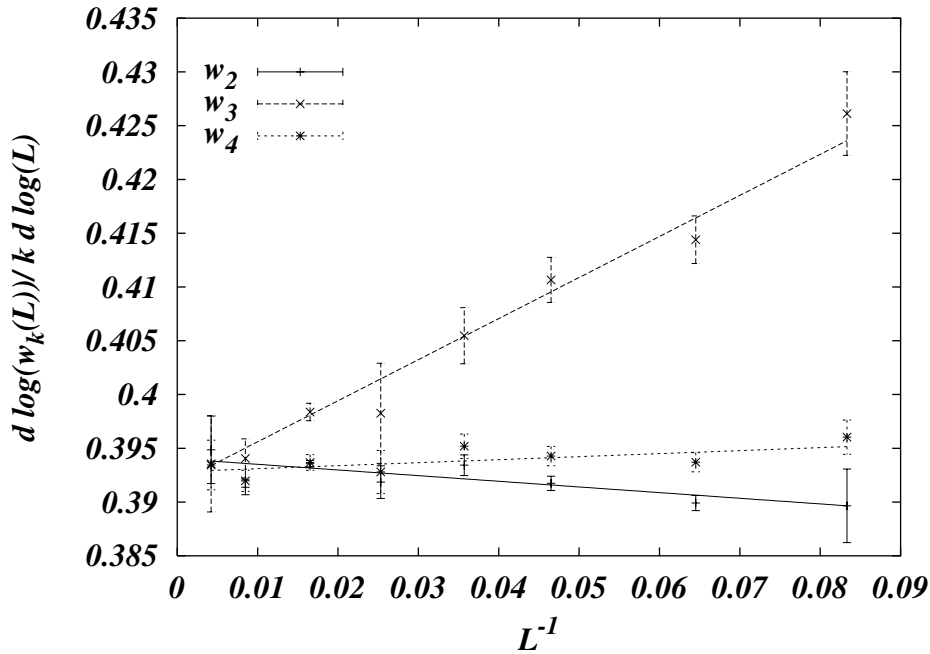


Figure 2. $L \frac{d \log(w_k(L))}{k dL}$ versus $\frac{1}{L}$ in $2D$, for $k = 2, 3, 4$.

precisely a single exponent of finite-size corrections, and also that we are not misled by finite-size corrections.

Let us now discuss the determination of the exponent χ .

Let us start by discussing the $D = 2$ case. A simple analysis of $w_2^{(L)}$ without corrections to scaling shows that a fit to lattices of linear size from $L = 19$ give a good value of chi squared and $\chi = 0.393$. A systematic analysis of the form (7) by including a lattice with $L \geq 11$ gives our final best value of

$$\chi_{D=2} = 0.393 \pm 0.003 \quad \omega_{D=2} = 1.1 \pm 0.3. \tag{11}$$

We plot the rescaled w_2 versus the rescaled time in figure 3: it is clear that the asymptotic plateau is exposed with good accuracy, and that the scaling is very good. In addition, the best fit to the form (7) is very good: we plot the numerical data for w_2 , w_3 and w_4 versus L and the best fit in 4.

We can compare with the rational guess of [9] that would give here $\chi_R = 0.4$. Indeed Kim and Kosterlitz in [9] conjecture that $\chi(d) = \frac{2}{d+3}$ (seemed to fit reasonably with the numerical results available at the time). Here Lässig [16], by using an operator product expansion, also finds $\chi(d = 2) = 0.4$. Our result is three standard deviations from the rational guess, which represents a safe distance. However, since we are dealing with a very complex situation, with many corrections that can possibly affect the result (sub-sub-leading corrections, short time, small volume, ...), we perform a further check to determine whether $\chi_R = 0.4$ is a plausible result. We fix $\chi = 0.4$, and now fit our data with seven and not eight free parameters ($A_2, A_3, A_4, B_2, B_3, B_4$ and ω). Now we obtain a very small value of $\omega \simeq 0.28$, and a chi squared that increases by a factor of ten from our previous best fit (where it was of order one per degree of freedom).

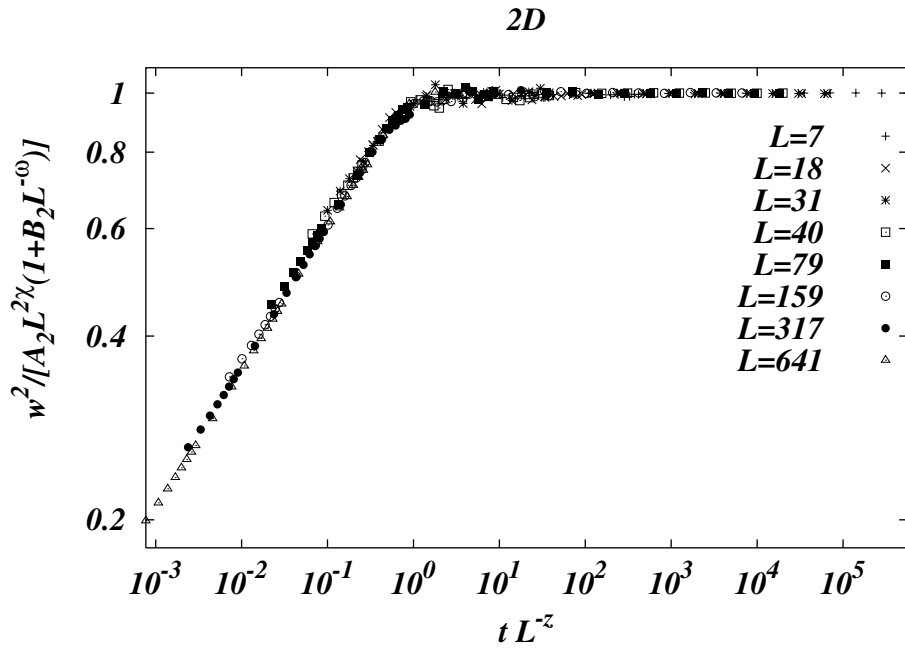


Figure 3. Scaling plot of the rescaled w_2 versus the rescaled time for the 2D case.

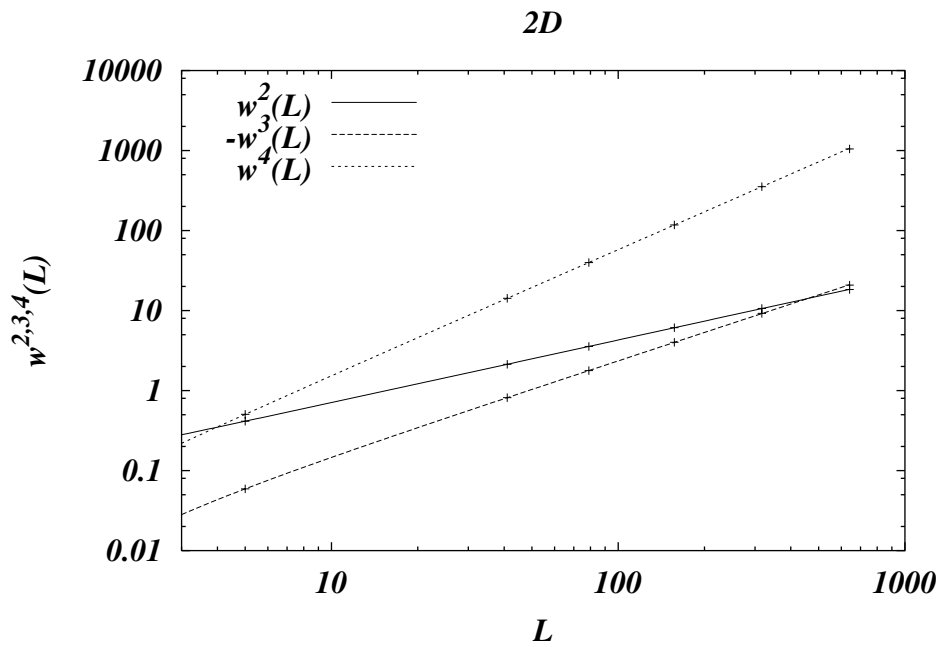


Figure 4. Numerical data for w_2 , w_3 and w_4 versus L and best fit for the 2D case.

In order to show that the fact that we have been able to exclude that the exponent takes the value 0.4 is not due to the hypothesis that the exponent of the sub-leading correction is the

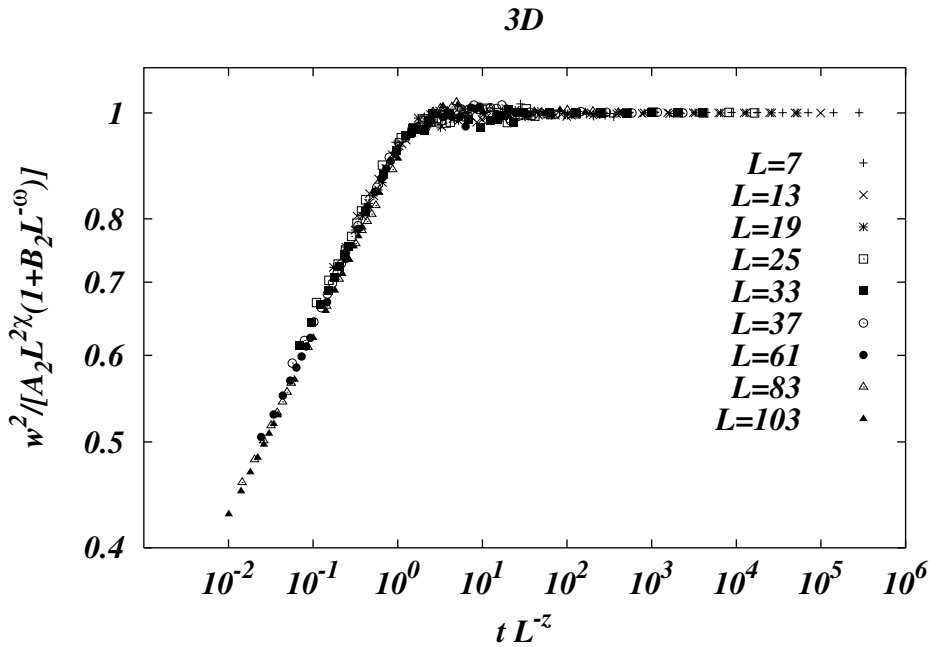


Figure 5. Scaling plot of the rescaled w_2 versus the rescaled time for the 3D case.

same for all momenta we can look again at figure 2. From these data it is already clear that the value 0.4 is excluded. The more sophisticated analysis which we have presented before is crucial in obtaining a controlled extrapolation to $L = \infty$, keeping the statistical errors under control. We believe that this is very strong evidence against the validity of the rational guess. We will see that for higher D values we obtain an even clearer discrepancy.

In 3D the same analysis of the three cumulants allows us to establish that

$$\chi_{D=3} = 0.3135 \pm 0.0015 \quad \omega_{D=3} = 0.98 \pm 0.08. \tag{12}$$

We include here sizes from $L = 11$ up to 103. We plot the rescaled w_2 versus the rescaled time in figure 5 and the numerical data for w_2 , w_3 and w_4 versus L and the best fit in figure 6. Here the rational guess would give $\chi_R = \frac{1}{3} \simeq 0.333$, while our value lies at more than ten standard deviations from this (Lässig gives here $\frac{2}{7}$, which is far from our estimate). The same check we have made for $D = 2$ leads to strong evidence: when fixing $\chi = \frac{1}{3}$ we again find a very small value $\omega \simeq 0.2$, and chi squared increases by a factor large than 20. Here the evidence against the validity of a rational guess is even stronger than in $D = 2$.

In 4D we use data for L starting from 10. Again the same three-cumulant analysis gives us

$$\chi_{D=4} = 0.255 \pm 0.003 \quad \omega_{D=4} = 0.98 \pm 0.09. \tag{13}$$

We plot the rescaled w_2 versus the rescaled time in figure 7 and the numerical data for w_2 , w_3 and w_4 versus L and the best fit in figure 8.

Here $\chi_R = \frac{2}{7} \simeq 0.286$, and, again, our value lies ten standard deviations away from the rational guess.

We summarize our findings in table 2, where we give all the best fit values of the parameters entering equation (7).

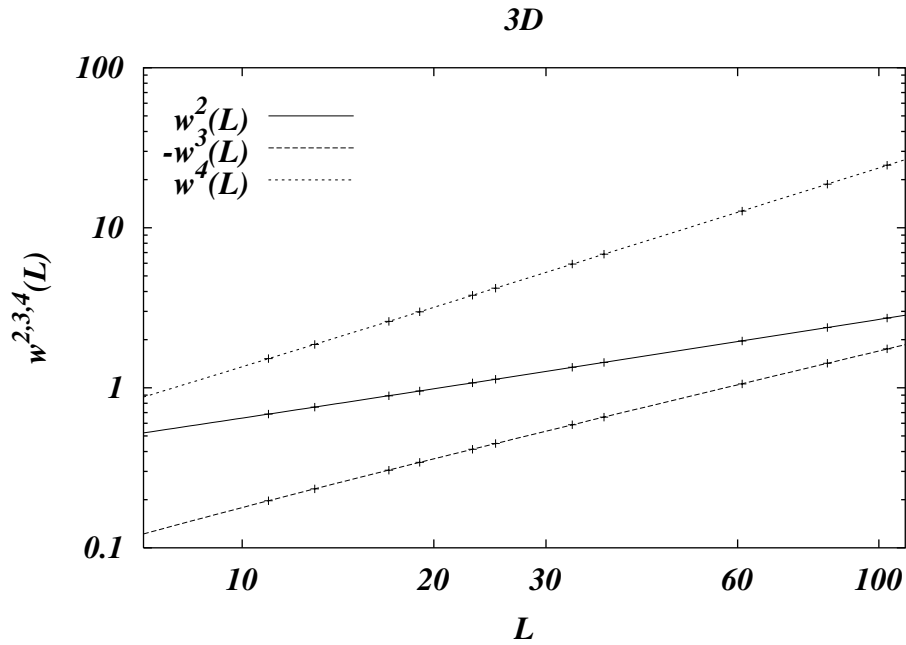


Figure 6. Numerical data for w_2 , w_3 and w_4 versus L and best fit for the 3D case.

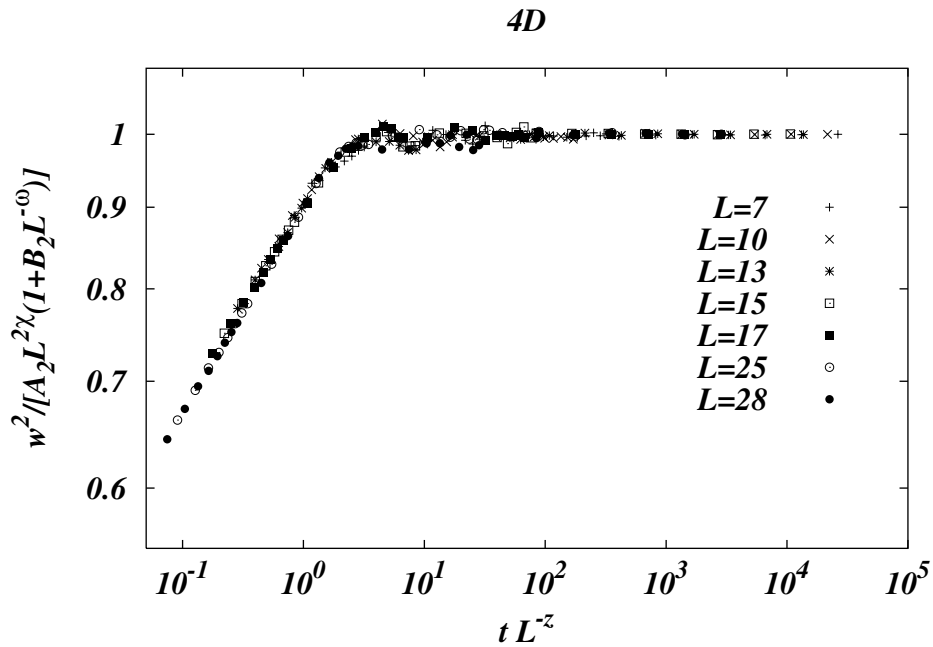


Figure 7. Scaling plot of the rescaled w_2 versus the rescaled time for the 4D case.

Two more observations are interesting. In the first we find $\omega \simeq 1$ for all the D values we investigate. In the second the pre-factor of the scaling corrections increases with D : we find $B_2(D = 2) \simeq 0.08$, $B_2(D = 3) \simeq 0.25$, $B_2(D = 4) \simeq 0.37$.

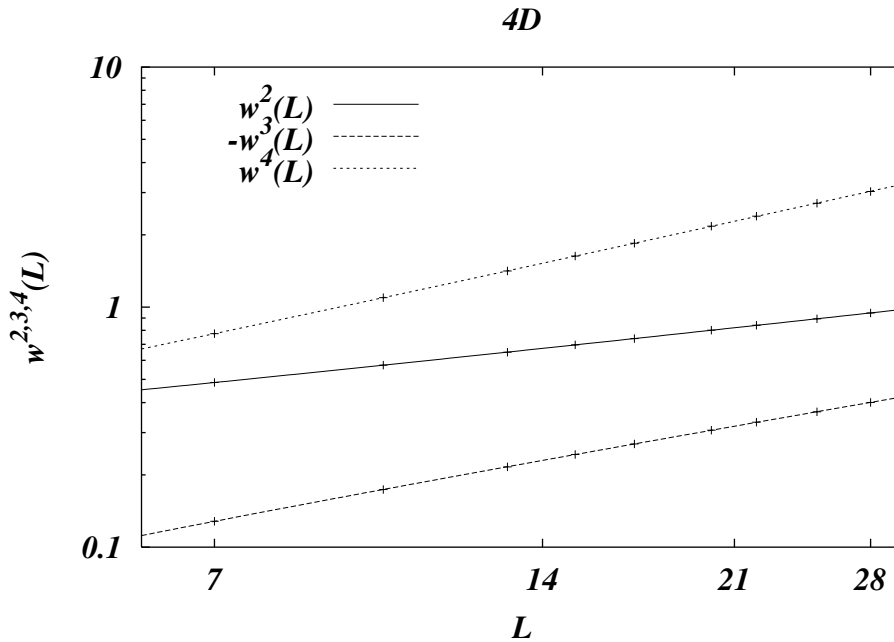


Figure 8. Numerical data for w_2 , w_3 and w_4 versus L and best fit for the 4D case.

Table 2. Best fit values of the parameters entering equation (7).

	A_2	B_2	A_3	B_3	A_4	B_4	ω	χ
2D	0.116(1)	0.08(1)	0.0105(1)	-0.8(1)	0.042(1)	-0.23(3)	1.1(3)	0.393(3)
3D	0.149(1)	0.25(2)	0.0226(1)	-0.9(1)	0.074(1)	0.28(1)	0.98(8)	0.3135(15)
4D	0.170(1)	0.37(3)	0.0321(2)	-0.7(1)	0.100(1)	0.46(4)	0.98(9)	0.255(3)

Our results are not incompatible with the recent ones of [17], but our small error bars allow us to reach precise conclusions. In addition, the comparison with the exponents found in [4] is fair: we have stressed the length of our runs, in order to be able to give a clean estimate of the asymptotic time behaviour, so that our result is hopefully unbiased.

5. Conclusions

The numerical technique we have introduced works well, and has allowed us to run very precise numerical simulations with a limited amount of computer time (a few months on a Pentium II processor).

We have been able to determine critical exponents of the KPZ universality class with high accuracy. We have falsified the guess that the exponents are simple rational numbers. It is also unambiguous from our data that the upper critical dimension is larger than four (as opposed to the claims of [7]).

Thanks to our precise measurements (and fitting together the first three non-trivial moments of h) we have also been able to determine the exponent ω of the first non-leading scaling corrections. It is interesting to notice that the estimated value of ω is always very close to unity, independent of the dimensionality of the system.

The next interesting step, following the approach of [6], would be to try and implement a systematic Monte Carlo renormalization group: the multi-surface coding technique we have discussed could be a very important ingredient of such a development.

Acknowledgments

For the numerical simulations described here we have used the *Kalix2* parallel computer (built from Pentium II chips), funded by Italian MURST 1998 COFIN. We thank Claudio Castellano and Matteo Marsili for interesting discussions, and Marcel den Nijs and Kay Wiese for relevant correspondence.

References

- [1] Kardar M, Parisi G and Zhang Yi-Cheng 1986 *Phys. Rev. Lett.* **56** 889
- [2] Halpin-Healy T J and Zhang C-Y 1995 *Phys. Rep.* **254** 215
Marsili M, Maritan A, Toigo F and Banavar J R 1996 *Rev. Mod. Phys.* **68** 963
- [3] Tang L H *et al* 1992 *Phys. Rev. A* **45** 7162
- [4] Ala-Nissila T *et al* 1993 *J. Stat. Phys.* **72** 207
- [5] Frey E and Tauber U C 1994 *Phys. Rev. E* **50** 1024
Lässig M 1995 *Nucl. Phys. B* **448** 559
Wiese K J 1997 *Phys. Rev. E* **56** 5013
Wiese K J 1998 *J. Stat. Phys.* **93** 143
- [6] Castellano C, Marsili M and Pietronero L 1998 *Phys. Rev. Lett.* **80** 3527
(Castellano C, Marsili M and Pietronero L 1998 *Preprint cond-mat/9802284*)
Castellano C, Gabrielli A, Marsili M, Muñoz M A and Pietronero L 1998 *Phys. Rev. E* **58** R5209
(Castellano C, Gabrielli A, Marsili M, Muñoz M A and Pietronero L 1998 *Preprint cond-mat/9809197*)
Castellano C, Marsili M, Muñoz M A and Pietronero L 1999 *Preprint cond-mat/9904434*
- [7] Lässig M and Kinzelbach H 1997 *Phys. Rev. Lett.* **78** 903
(Lässig M and Kinzelbach H 1996 *Preprint cond-mat/9608099*)
- [8] Tu Y 1994 *Phys. Rev. Lett.* **73** 3109
- [9] Kim J M and Kosterlitz J M 1989 *Phys. Rev. Lett.* **62** 2289
Kim J M, Kosterlitz J M and Ala-Nissila T 1991 *J. Phys. A: Math. Gen.* **24** 5569
- [10] Rieger H 1992 Fast vectorized algorithm for the Monte Carlo simulation of the random field Ising model *Preprint HLRZ 53/92*
(Rieger H 1992 *Preprint hep-lat/9208019*)
- [11] Liggett T M 1985 *Interacting Particles Systems* (Heidelberg: Springer) ch 3
- [12] Forrest B M and Tang L-H 1990 *J. Stat. Phys.* **60** 181
- [13] Ballesteros H G, Fernandez L A, Martin-Mayor V and Munoz-Sudupe A 1996 *Phys. Lett. B* **387** 125
Ballesteros H G, Fernandez L A, Martin-Mayor V, Munoz-Sudupe A, Parisi G and Ruiz-Lorenzo J J 1998 *Phys. Rev. B* **58** 2740
- [14] Flyvbjerg H 1998 Error estimates on averages of correlated data *Advances in Computer Simulation* ed J Kerstéz and I Kondor (Berlin: Springer) p 88
- [15] Chin C-S and den Nijs M 1999 *Phys. Rev. E* **59** 2633
- [16] Lässig M 1998 *Phys. Rev. Lett.* **80** 2366
(Lässig M 1997 *Preprint cond-mat/9711037*)
- [17] Kim J M 1999 *Physica A* **270** 335