Vectorized and parallel simulations of the Kardar-Parisi-Zhang equation in 3+1 dimensions

1994 J. Phys. A: Math. Gen. 27 4049

(http://iopscience.iop.org/0305-4470/27/12/013)

J. Phys. A: Math. Gen. 27 (1994) 4049–4054. Printed in the UK

Vectorized and parallel simulations of the Kardar–Parisi–Zhang equation in 3 + 1 dimensions

Keye Moser† and Dietrich E Wolf‡

† Institut für Theoretische Physik, Universität zu Köln, Zülpicher, Strasse 77, D-50937 Köln, Germany

‡ Theoretische Physik, FB 10, Universität Duisburg, D-47048 Duisburg, Germany

Received 4 March 1994

Abstract. The Kardar–Parisi–Zhang (KPZ) equation describing kinetic roughening is solved numerically for (3 + 1)-dimensional systems in the strong-coupling phase. By massive use of supercomputing tools we calculate for the first time the exponent $\beta(= 0.181 \pm 0.007)$ with an accuracy comparable with that of lattice growth model simulations. A possible source of errors in parallel Monte Carlo calculations is pointed out.

1. Introduction

Algebraic roughening of growing surfaces can be found in several models [1], but here we concentrate on the case of the Kardar–Parisi–Zhang (KPZ) equation [2]

$$\partial_t h = \nu \Delta h + \tfrac{1}{2}\lambda(\nabla h)^2 + \eta \tag{1}$$

where $\nu > 0$ guarantees the stability of globally flat surfaces, λ describes lateral growth and η is a Gaussian noise with correlator $\langle \eta(\boldsymbol{x}, t)\eta(\boldsymbol{x}', t')\rangle = 2D\delta(\boldsymbol{x} - \boldsymbol{x}')\delta(t - t')$.

Roughening implies scale-invariant surface fluctuations. One possible quantity of interest is the surface width

$$\langle w \rangle = \langle (\overline{h^2} - \overline{h}^2)^{1/2} \rangle \tag{2}$$

where the bar denotes the spatial average over a (sub)system of linear size L and the angular brackets mean the disorder average over many realizations of the noise η.

Family and Vicsek [3] suggested for w the scaling form

$$\langle w \rangle \sim L^{\zeta} f(t/L^{z}) \tag{3}$$

where $f(x) \sim x^{\beta}$ with $\beta = \zeta/z$ for $x \ll 1$, and $f(x) \sim$ constant for $x \gg 1$. The roughness exponent ζ and the dynamical exponent z fulfil a scaling relation [4–6]

$$\zeta + z = 2. \tag{4}$$

As the one-loop dynamical renormalization group (RG) [7] shows, there exists only one effective coupling constant of physical relevance, namely

$$g' = \frac{\lambda^2 D}{\nu^3}. \tag{5}$$

0305-4470/94/124049+06$19.50

The RG [7] predicts a phase diagram for KPZ growth [8] that is dependent on the surface dimension d of the $(d+1)$-dimensional system. In $3+1$ dimensions two distinct phases exist: algebraic roughening is expected only in the rough phase at large g'. Unfortunately, the RG is not able to predict values for the exponents.

The exponent β has been measured in several lattice growth models (e.g. [9–11]). Direct numerical solutions of the KPZ equation [8, 12] are in agreement with these results for $d = 1$ and 2, but in the $(3+1)$-dimensional case no data of comparable precision have so far been published. We only found some indication of the phase transition, but due to the technical difficulties of the large-scale simulations required we were not able to obtain a reliable value for β. Here we extend this earlier work [8] and present results which were obtained using high-performance vector and parallel vector supercomputers, for which our program was specially adapted.

In the next section we discuss our improved algorithm. In section 3 we present our results and report on a problem arising when different sequences of the same random number generator (RNG) are used in domain decomposition on the parallel computer. Finally, we give a short summary.

2. Technique

We used the Euler algorithm to integrate equation (1) numerically. In [8] we presented a detailed discussion and test of this method. Thus we want to show here only the dimensionless discretized version of equation (1), which was implemented:

$$h_{\underline{n}}(t+\Delta t) = h_{\underline{n}}(t) + \frac{\Delta t}{\Delta x^2}\sum_{i=1}^{d}[(h_{\underline{n}+\underline{e}_i}(t) - 2h_{\underline{n}}(t) + h_{\underline{n}-\underline{e}_i}(t)) + \tfrac{1}{8}(h_{\underline{n}+\underline{e}_i}(t) - h_{\underline{n}-\underline{e}_i}(t))^2] + \sqrt{12\Delta t}R_{\underline{n}}(t). \tag{6}$$

Here $R_{\underline{n}}$ are uniformly distributed random numbers with $-\frac{1}{2} \leqslant R_{\underline{n}} \leqslant \frac{1}{2}$. All simulations start with a flat surface: $h_{\underline{n}}(0) = 0$. The spatial discretization parameter Δx is related [8] to the dimensionless coupling constant by $g = (\Delta x)^d/2$, and the time discretization Δt has to be small enough to guarantee the numerical stability.

We refined the existing algorithm in order to minimize memory requirements. It will be shown that this technique is also perfectly adapted to parallel supercomputers, with an additional advantage with respect to communication times.

Normally at least *two* arrays of system size are needed to store the field $h_{\underline{n}}$ and its update $h'_{\underline{n}}$ which after completion of one lattice sweep is copied into $h_{\underline{n}}$ (figure 1(*a*)). This is not necessary as only nearest-neighbour sites are needed to update $h_{\underline{n}}$ and as the updating is done in a typewriter mode starting with h_1 and ending with h_N. (Note that we use helical boundary conditions so that the site index $\underline{n}$ is replaced by a one-dimensional index.) Memory locations which are no longer needed can be overwritten: h'_n is stored in the location of $h_{n-\Delta}$, where $\Delta = L^{d-1}$ (figure 1(*b*)). The boundary conditions at the upper and lower ends of the lattice are implemented using $(d-1)$-dimensional buffer 'lines' with Δ sites each.

The system is shifted in each sweep with respect to the physical memory locations. (Even if no translational invariance is present, this drift may be easily corrected if necessary by protocolling the number of sweeps which were made.)

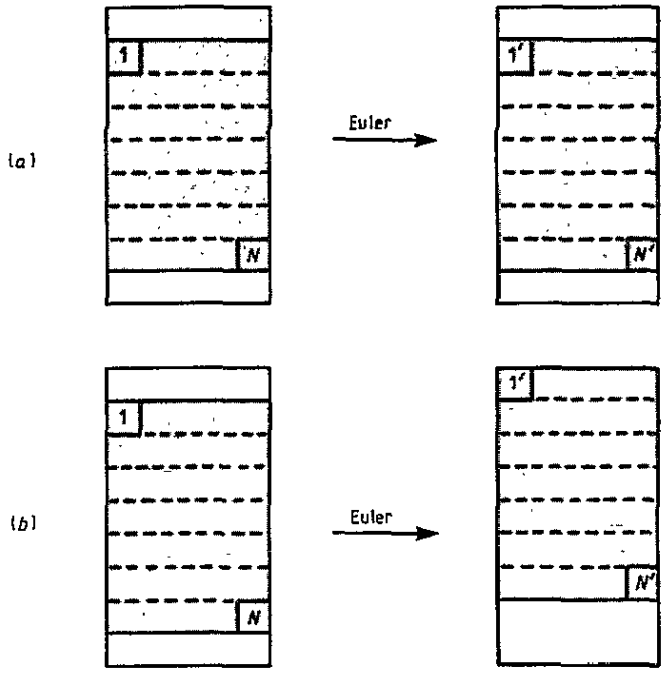

Figure 1. Helical boundary conditions with and without 'trick' (see text). Light: buffer lines.

On vector supercomputers this technique saves memory, which allows larger systems to be investigated. On parallel supercomputers this benefit exists as well but in addition there is a slight time advantage which will be described below.

To take advantage of the full computing performance of all processors ('nodes') and because of the large system size, we used domain decomposition for the parallelization of the existing program. This means slicing the whole system and assigning one slice to each of the nodes. After every sweep one has to exchange buffer lines between neighbouring nodes.

Such communication between the processors is *very* time consuming. On an IPSC/860 parallel computer, for instance, the setup of a connection between two nodes takes as much time as 1400 floating-point operations (at typically 8 MFlops/node). This is required every time a node has to send data to another one. On top of that the transmission of every byte takes the same time as approximately four floating-point operations.

According to figure 1(a) each node communicates one line to the upper and one to the lower neighbour to update the corresponding buffer lines: Two connections have to be set up and two messages (of typically 100 kbyte each) have to be sent. In our case, figure 1(b), only one connection (to the upper neighbour) is needed for transmitting a message of double size (the two uppermost lines of figure 1(b)). This reduces the communication overhead.

3. Results

We chose a system size of 160^3. The dimensionless coupling constant was $g = 357.7$, which is expected [8] to be deep in the rough strong-coupling phase. Δt was adjusted to

1.256 to achieve numerical stability. The random numbers were generated with an R250 Tausworthe RNG [13]. One run represents 15 923 timesteps.

On the vector computers (Cray Y-MP, $\sim$ 195 MFlops; NEC SX-3, $\sim$ 571 MFlops) no problems appeared. Figure 2 shows data from averaging over 25 runs. The resulting effective exponent β_{eff} in the asymptotic scaling regime ($t > 2000$) is

$$\beta_{\text{eff}} = 0.181 \pm 0.007 \tag{6}$$

with very conservative error bars. This is in good agreement with values obtained by other authors (Forrest and Tang [10], 0.180 ± 0.005; Ala-Nissila *et al* [11], 0.180 ± 0.002), which were obtained from lattice growth models.

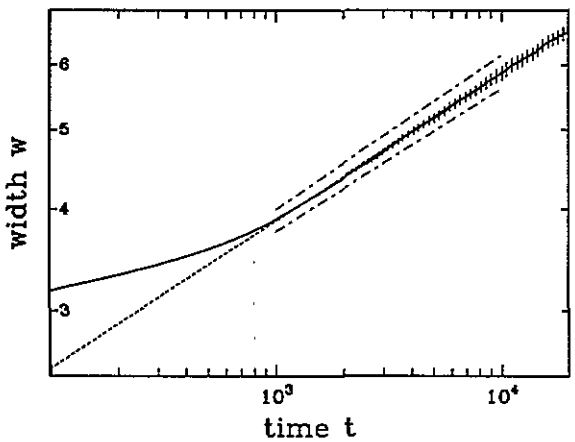

Figure 2. Log–log plot of the width w against time t as obtained from the numerical solution of the (3 + 1)-dimensional KPZ equation: $g = 357.7$; $\Delta t = 1.256$. The system size was $160 \times 160 \times 160$ and an average over 25 runs was taken. The broken curve represents the effective exponent $\beta_{\text{eff}} = 0.181$ in the asymptotic scaling regime ($t > 2000$). Statistical errors are indicated by the straight lines above and below the curve.

On the IPSC/860 parallel computer problems arose with the RNG. In order to keep communications small, independent RNGs have been started on all 32 nodes. Because of the large period (e.g. [14]), we decided to use the R250 Tausworthe RNG, initialized differently on all the nodes. We observed a strong dependence on the starting and setup procedure for this RNG.

Figure 3 shows data obtained if one determines the 250 start numbers for the R250 from a linear congruential RNG and uses consecutive odd numbers as seeds for the different nodes. Also shown are the data from figure 2, where only a single sequence of this RNG per run was used. The system parameters are equal in both cases, apart from different numbers of runs. The values of the surface width obtained from the parallel program are systematically too small, as if the system had a smaller size. We interpret this as an indication of correlations among the random number sequences on different nodes. By contrast, the recently reported correlations within one sequence produced by the Tausworthe RNG [15] have no measurable effect in the present simulation, as had already been checked in [8], where a linear congruential RNG and the R250 gave the same results.

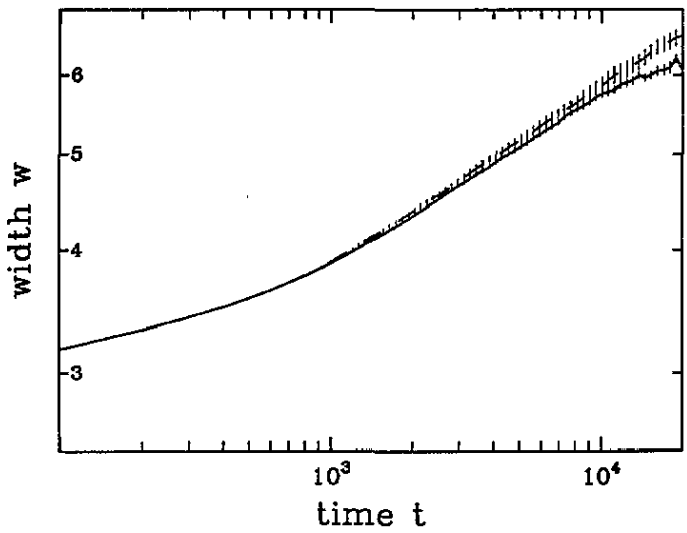

Figure 3. Comparison of the data from figure 2 (broken upper curve) to the data achieved on the parallel computer IPSC/860 (full lower curve). $g = 357.7$, $\Delta t = 1.256$ and system size $160 \times 160 \times 160$ in both cases. On the parallel computer an average over 55 runs was taken. The lower slope of the latter curve indicates problems with the RNG.

The problem of correlations among different random number sequences is not specific to the Tausworthe RNG. We also observed it in the linear congruential RNG used to start the R250

$$I_{n+1} = I_n * 16807$$

(32-bit machine). We started sequences $m = 1 \ldots M$ with seeds $I_0^{(m)} = I_0^{(1)} + 2m$. Plotting $I_n^{(1)}, \ldots, I_n^{(M)}$ for arbitrarily large fixed n reveals strong correlations among the numbers which form a type of saw-tooth curve.

Because of domain decomposition these correlations amount to a reduction in shot noise on scales larger than one domain and can lead to an effective reduction in the system size, as described above. This problem can presumably be avoided by warming the RNGs up over random time intervals.

4. Summary

We have presented new results for rough (3+1)-dimensional systems obtained by solving the KPZ equation numerically. In order to simulate the largest possible systems we optimized the usage of memory. As a side effect this also reduced the communication overhead on the parallel computer. This technique is not specific to the problem studied here but should work for many other models in statistical physics as well.

We analysed a problem which occurs on parallel computers using domain decomposition, if the RNGs on the different nodes are not *very* carefully initialized. An unexpected strong dependence on the setup procedure was observed, which indicates correlations among different sequences.

Finally, the resulting effective exponent $\beta_{\text{eff}} = 0.181 \pm 0.007$ is in good agreement with exponents obtained previously [10, 11] from lattice growth models.

Acknowledgments

We thank the supercomputer centres HLRZ (Jülich), ZAM/KFA (Jülich) and the RRZ (Köln) for providing us with computer time on the CRAY Y-MP, the IPSC/860 and the NEC SX-3. This work was carried out within SFB 341 of the Deutsche Forschungsgemeinschaft.

References

[1] Villain J 1991 *J. Physique I* **1** 19
Jullien R, Kertész J, Meakin P and Wolf D E (ed) 1992 *Surface Disordering: Growth, Roughening, and Phase Transitions* (New York: Nova Science, Commack)
[2] Kardar M, Parisi G and Zhang Y-C 1986 *Phys. Rev. Lett.* **56** 889
[3] Family F and Vicsek T 1985 *J. Phys. A: Math. Gen.* **18** L75
[4] Meakin P, Ramanlal P, Sanders L M and Ball R C 1986 *Phys. Rev.* A **34** 5091
[5] Krug J 1987 *Phys. Rev.* A **36** 5465
[6] Wolf D E and Kertész J 1989 *Phys. Rev. Lett.* **63** 1191
[7] Medina E, Hwa T, Kardar M and Zhang Y-C 1989 *Phys. Rev.* A **39** 3053
[8] Moser K, Kertész J and Wolf D E 1991 *Physica* **178A** 215
[9] Kim J M and Kosterlitz J M 1989 *Phys. Rev. Lett.* **62** 2289
[10] Forrest B and Tang L 1990 *Phys. Rev. Lett.* **64** 1405
[11] Ala-Nissila T, Hjelt T, Kosterlitz J M and Venäläinen O 1993 *J. Stat. Phys.* **72** 207
[12] Amar J G and Family F 1990 *Phys. Rev.* A **41** 3399
[13] Tausworthe R C 1965 *Math. Comput.* **19** 201
[14] Vattulainen I, Ala-Nissila T, Kankaala K and Saarinen J 1994 Nature of correlations in some random number generators *Preprint*
Kankaala K *et al* 1993 *Phys. Rev.* E **48** 4211
[15] Ziff R M private communication and footnote [15] of 1992 *Phys. Rev. Lett.* **69** 2670
Ferrenberg A M, Landau D P and Wong Y J 1992 *Phys. Rev. Lett.* **69** 3382
Selke W, Talapov A L and Shchur L N 1993 *JETP Lett.* **58** 8