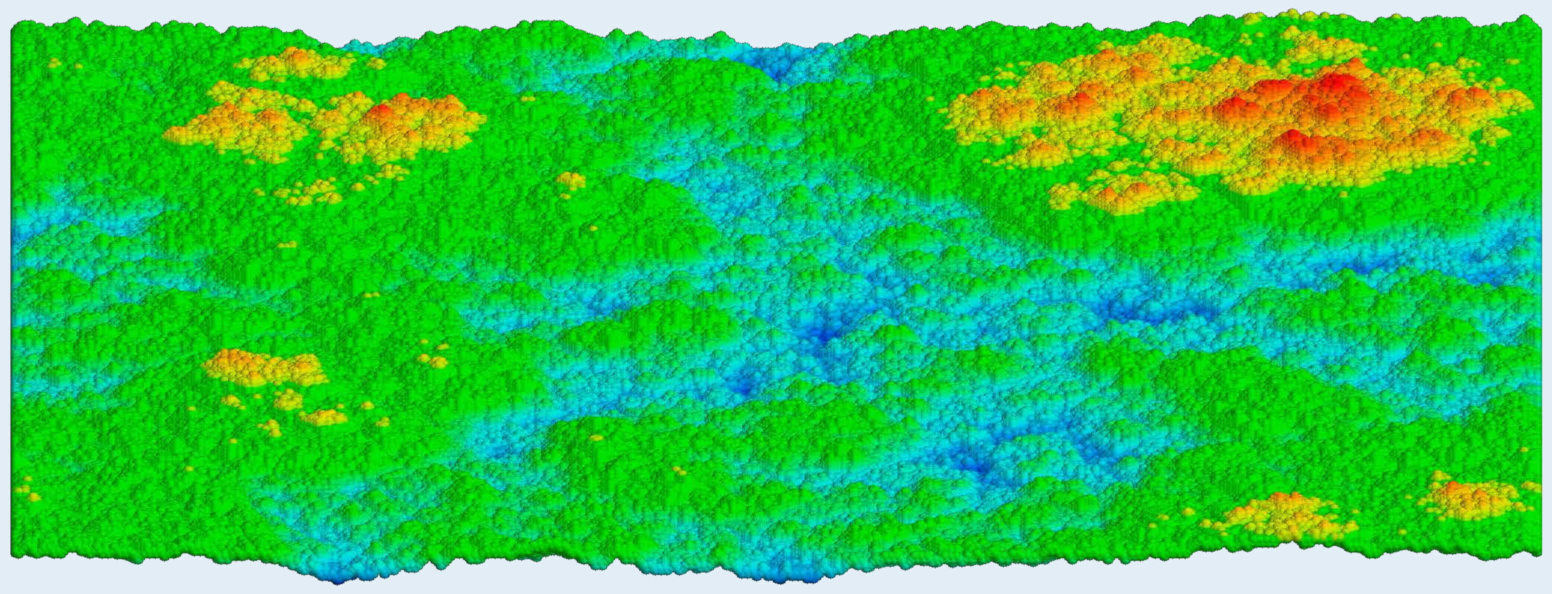
Non-Local Lattice Encoding for Bit-Vectorized Cellular Automata GPU Implementations

Jeffrey Kelling[1], Géza Ódor[2], Sibylle Gemming[1,3]
[1]Institute of Ion Beam Physics and Materials Research, Helmholtz-Zentrum Dresden-Rossendorf, Germany
[2]MTA-EK-MFA Institute of Technical Physics and Materials Science, Budapest, Hungary
[3]Institute of Physics, TU-Chemnitz, Germany

NVIDIA GPU CENTER OF EXCELLENCE · DRESDEN concept · HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

- stochastic cellular automata are widely used
  - in combination with machine learning to predict land use change [1]
  - in magnetism for the Ising model [2]
  - for surface growth models [3] like restricted solid-on-solid model
- large scale atomistic simulations required to understand self-organization and evolution of various systems



growing surface in the steady state

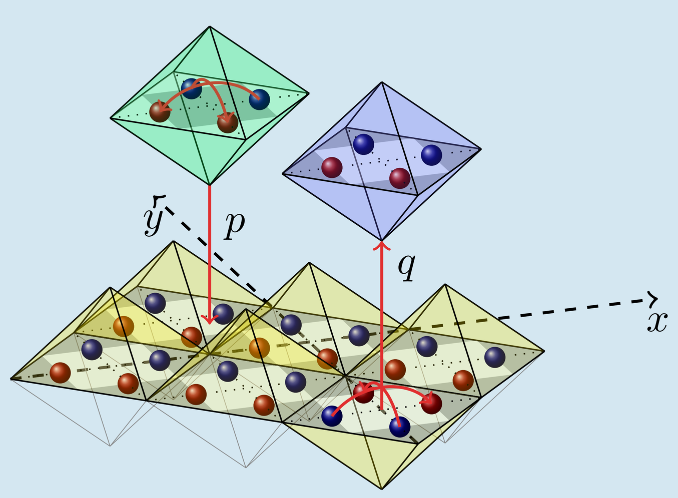
- simulating randomly growing surface with stochastic cellular automaton (SCA)
- Kardar–Parisi–Zhang (KPZ) or Edwards-Wilkinson (EW) universality class [4]
- can describe *growth processes*, directed polymers in random media, randomly stirred fluids, dissipative transport and magnetic flux lines in superconductors.

$$d_t h(\mathbf{x}, t) = \underbrace{v}_{\text{mean growth vel.}} + \underbrace{\sigma_2 \nabla^2 h(\mathbf{x}, t)}_{\text{surface tension}} + \underbrace{\lambda \left[\nabla h(\mathbf{x}, t)\right]^2}_{\text{local growth vel.}} + \underbrace{\eta(\mathbf{x}, t)}_{\text{noise}}$$

KPZ stochastic differential equation

- GPGPU enables simulations of micron-sized volumes, billions of atoms and studies of the long-time evolution of systems

Stochastic CA for Roof-Top Model



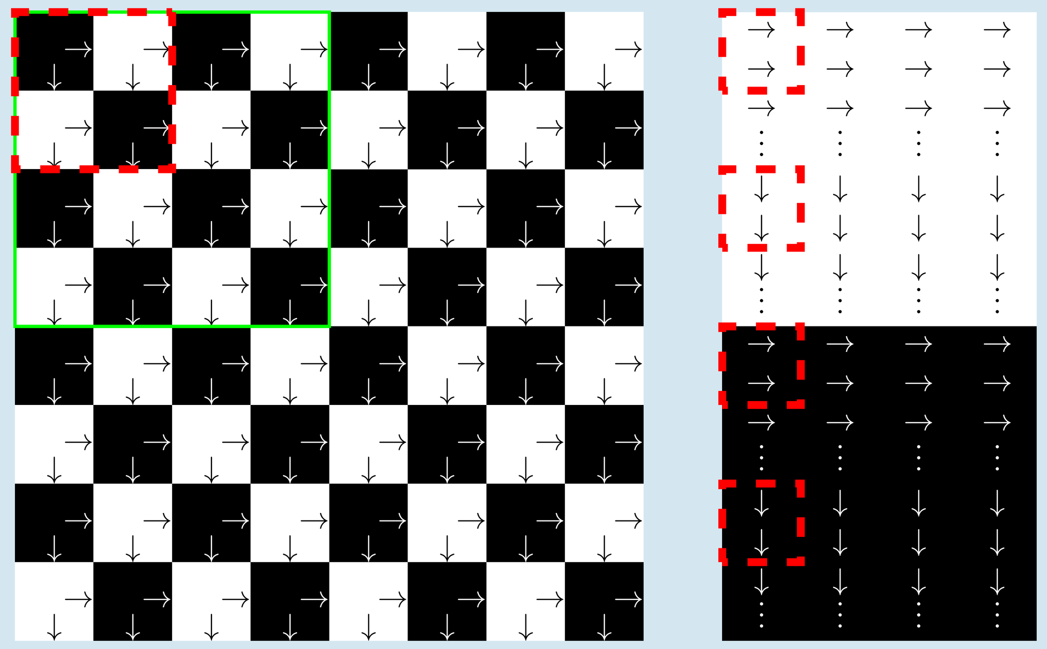
2 + 1D roof-top model—octahedron model

- stochastic model for surface growth [5, 6]
- 2D square lattice (octahedron model)
- carry out *random* depositions (probability p) or removals (probability q)
- time t measured in sweeps of the lattice: Monte-Carlo-Steps (MCS)
- KPZ universality for $q = 0$
- EW universality for $p = q$

Checkerboard Update Algorithm (SCA)

- bit-coded up- or down-slopes in between lattice-sites, two bits per site
- surface decomposed into checkerboard of even (white) and odd (black) lattice
- sites on the even or odd lattice, respectively, do not interact with each other during single lattice sweeps
  - update of sites on one sub-lattice can be carried out in any order
  - sub-lattices are updated in alternation

Bit-Vectorized Implementation using Non-Local Encoding



local encoding: 4×4 sites per dword — non-local encoding

- arrows (→, ↓) represent slopes connecting sites
- red frames indicate corresponding data in the respective encodings

Local → Non-Local Encoding

- in *local encoding* attributes of a lattice site are encoded *on site*, sites are ordered by physical coordinates
  - good for random site-selection, neighbors can be accessed in single memory transaction
- in *non-local encoding* same attributes of all sites are stored at consecutive addresses, sites are ordered by sub-lattice first (white: even, black: odd)
  - required for bit-level vectorization
  - reduced bandwidth overhead when updating one sublattice only

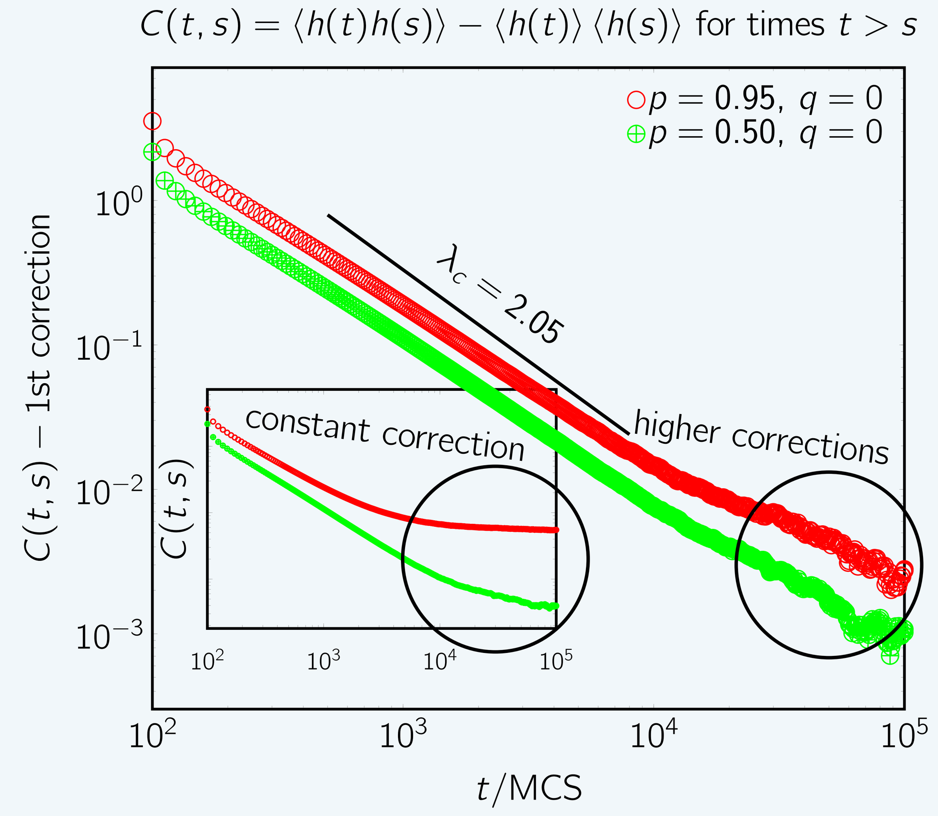
GPU Implementation

- each thread processes chunks of 32 lattice sites (dword), neighboring in x direction on one sublattice
  - requires only one random number for 32 updates, relies on all bits being random
- block used as virtually larger word-size (larger SIMD-size)
  - data exchange between threads via shared memory, no warp shuffle.

⇒ manual caching improves memory access efficiency (100 %, except fur huge systems)

Future Application Towards Surface Aging

Auto-correlation of surface heights (h) during growth
$C(t, s) = \langle h(t)h(s)\rangle - \langle h(t)\rangle \langle h(s)\rangle$ for times $t > s$



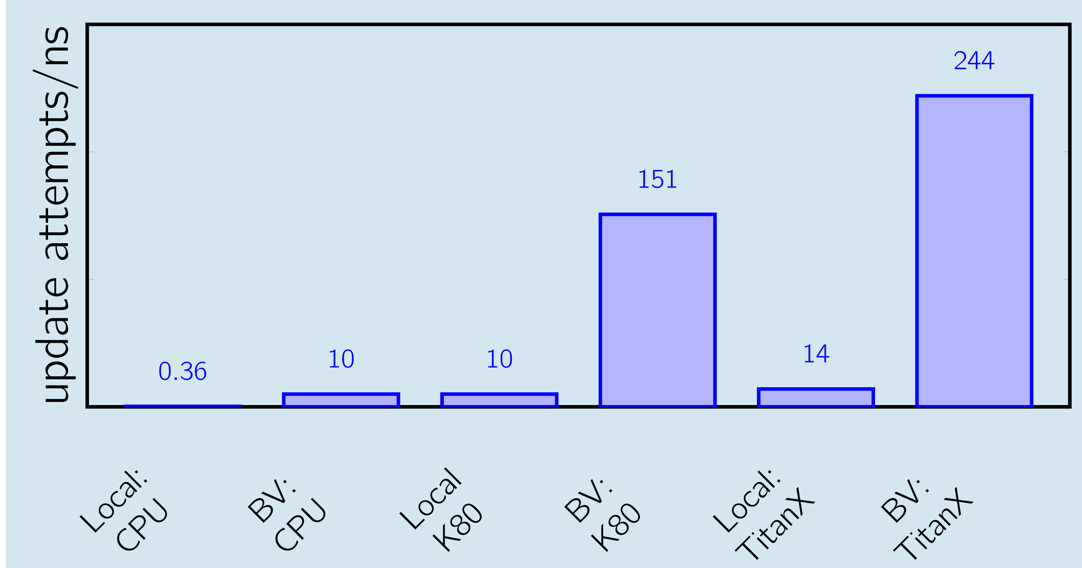
- stochastic surface growth can only be modelled exactly by sequential random site-selection
- SCA is widely used and more efficient

\+ SCA gives correct asymptotic scaling properties

– SCA introduces strong corrections to autocorrelation properties

! corrections have not been studied before
  - constant correction can be extrapolated, but requires long-time runs for small p —decreases exponentially with p
  - higher corrections require much statistics and long-time runs

Benchmarks



- Bit-vectorized (BV) CPU code uses 64-bit vectors
  - 2× Xeon E5-2603 (6 Cores @1.6 GHz)
  - SSE and AVX via gcc vector extensions did not result in speedup

Conclusions

- ∼ 20× speedup for bit-level vectorization over non-vectorized SCA on GPU, due more efficient use of random numbers and global memory access
- approach transferable to other lattice-based SCAs (and CAs)

⇒ will be able to compute higher-order correlation corrections

Acknowledgements

We thank the HZDR computing center and M. Bussmann for providing computational resources. We acknowledge the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for providing us with access to the Taurus cluster and NIIF for awarding us access to clusters in Budapest and Debrecen (Hungary), This work has been partially funded by the Initiative and Networking Fund of the German Helmholtz Association via the W2/W3 Programm für exzellente Wissenschaftlerinnen (W2/W3-026) and the Helmholtz International Research School for Nanoelectronic Networks NanoNet (VH-KO-606). We gratefully acknowledge further funding by the GCoE Dresden.

[1] Charif, O., Basse, R.-M., Omrani, H. & Trigano, P. In *Proceedings of the Winter Simulation Conference*, WSC '12, 163:1–163:12.
[2] Newman, M. E. J. & Barkema, G. T. *Monte Carlo Methods in Statistical Physics* (Oxford University Press, 1999).
[3] Forrest, B. M. & Tang, L.-H. *Journal of Statistical Physics* **60**, 181–202 (1990).
[4] Kardar, M., Parisi, G. & Zhang, Y.-C. *Phys. Rev. Lett.* **56**, 889–892 (1986).
[5] Plischke, M., Rácz, Z. & Liu, D. *Phys. Rev. B* **35**, 3485–3495 (1987).
[6] Ódor, G., Liedke, B. & Heinig, K.-H. *Phys. Rev. E* **79**, 021125 (2009).