Efficient Implementation of Sweeny's Algorithm for Simulations of the Random Cluster Model

E. M. Elçi and M. Weigel

Applied Mathematics Research Centre
Coventry University
United Kingdom

13th International NTZ-Workshop on New Developments in Computational Physics, Leipzig 2012

Random-Cluster model (RCM)

- "On the Random-Cluster Model" (Fortuin & Kasteleyn, Physica Vol. 57/Issue 4, 1972)
- Defined on a graph $G = (V, E)$ with fixed number of vertices N; here square lattice
- Has two parameters $0 \leq p \leq 1$ and $q > 0$ with partition function:

$$\mathcal{Z}_{RC}(p, q) = \sum_{A \subseteq G} p^{b(A)} \, (1-p)^{|E|-b(A)} \, q^{k(A)} \tag{1}$$

- Summation over all sub-graphs with $A = (V, E')$ where $E' \subseteq E$.
- $k(A)$: number of components/clusters; $b(A)$: number of edges in A
- Restricting to $q \in \{2, 3, ...\} \subseteq \mathbb{N}$ and setting $p = 1 - e^{-\beta J}$ is equivalent to the Potts model ($v = e^{\beta J} - 1$)

$$\mathcal{Z}_{\text{Potts}}(\beta J, q) = \sum_{A \subseteq G} v^{b(A)} q^{k(A)} = \mathcal{Z}_{\text{RC}}(\frac{v}{1+v}, q) \tag{2}$$

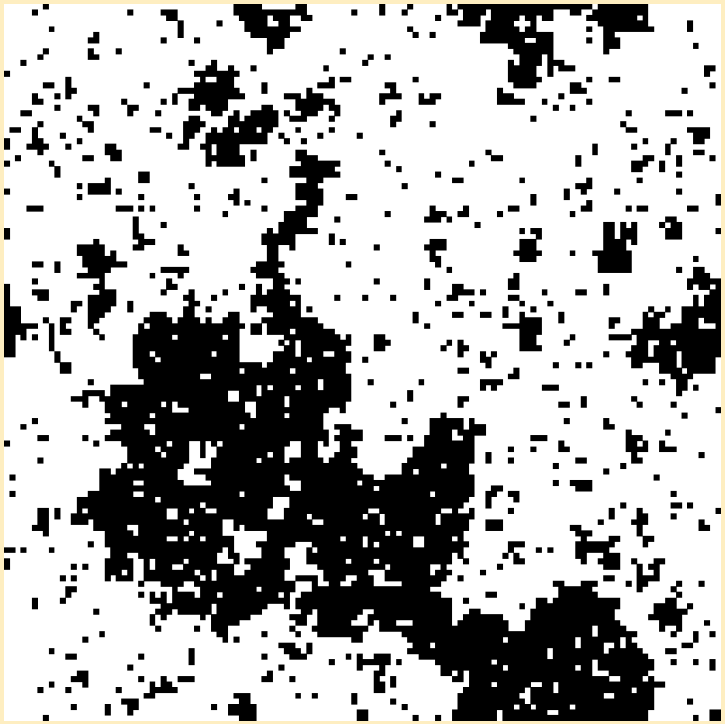
RCM: Unification of a variety of stochastic processes

- $q \in \{2, 3, ...\} \subseteq \mathbb{N}$: q-state **Potts model** with critical point $v_c = \sqrt{q}$
  - $q = 2$: Ising model
- $q \to 1$: **Bond percolation**
- $q \downarrow 0$ with fixed v: "**Maximal connected spanning sub-graphs**"
  - Dominant terms are those with $k(A) = 1$ for connected G
  - Bond percolation with $p = v/(1 + v)$ conditioned that resulting graph is connected
- $q \downarrow 0$ with $v \sim q^{\alpha}$ and $0 < \alpha < 1$: "**Maximal spanning forest**"; for connected G:
  - "Uniform spanning trees"; linked to theory of **electrical networks** (Kirchoff's theorem)
  - Bond percolation with $p = 0.5$ conditioned that resulting graph is a spanning tree

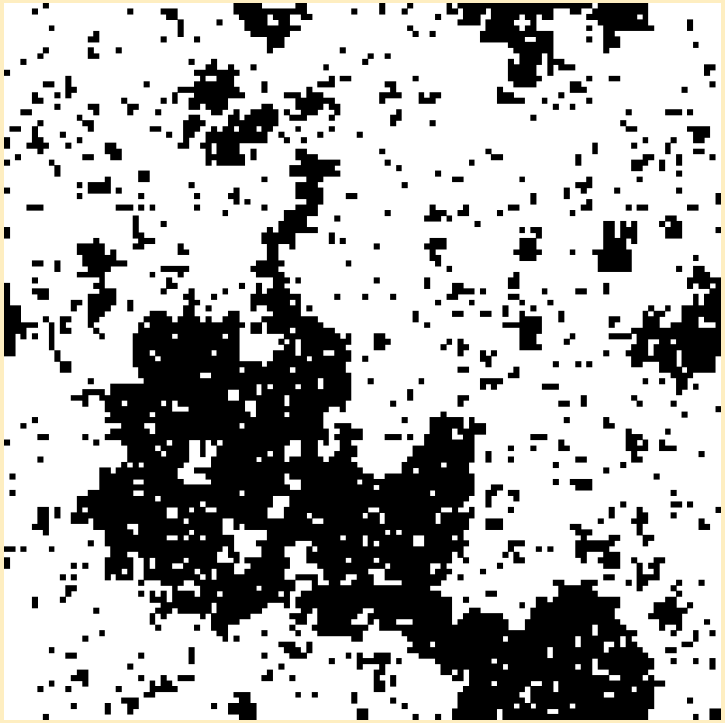
Swendsen-Wang (SW) Algorithm

- "Nonuniversal critical dynamics in Monte Carlo simulations", (Swendsen and Wang, PRE 58 86-88, 1987)
- Works for integer $q \geq 1$: Potts Model
- Based on a joint spin-/bond- measure whose marginal measure on the spins/bonds equals Potts/RCM measure
- SW switches between both representations:
  - Given a spin configuration create a bond between two vertices if they have the same spin value, otherwise create a bond with probability $p = 1 - e^{-\beta J}$
  - Based on the bond-configuration assign a unique randomly chosen spin to every component
- Generalized to non-integer $q \geq 1$ by Chayes and Machta (CM) in 1997 (Physica A, Vol. 239/Issue 4)

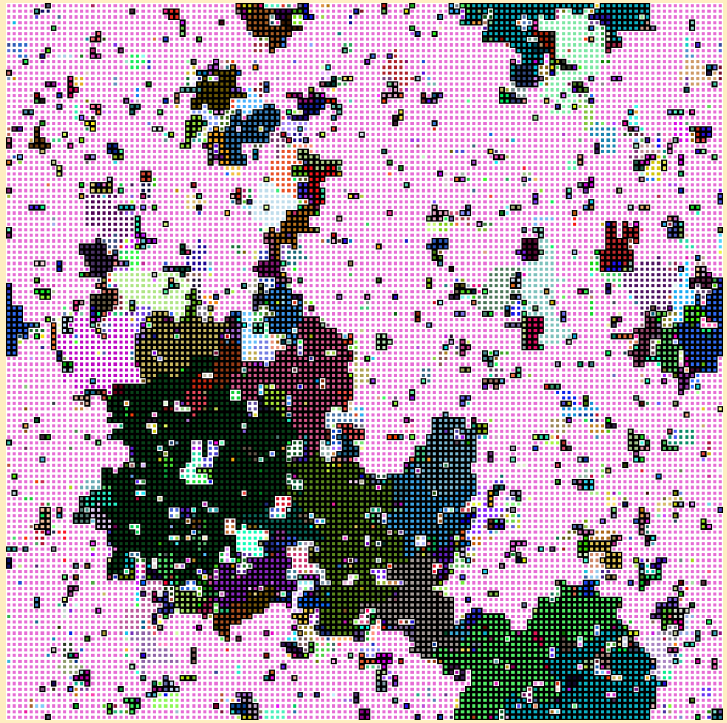
Ising spin configuration



Ising spin configuration



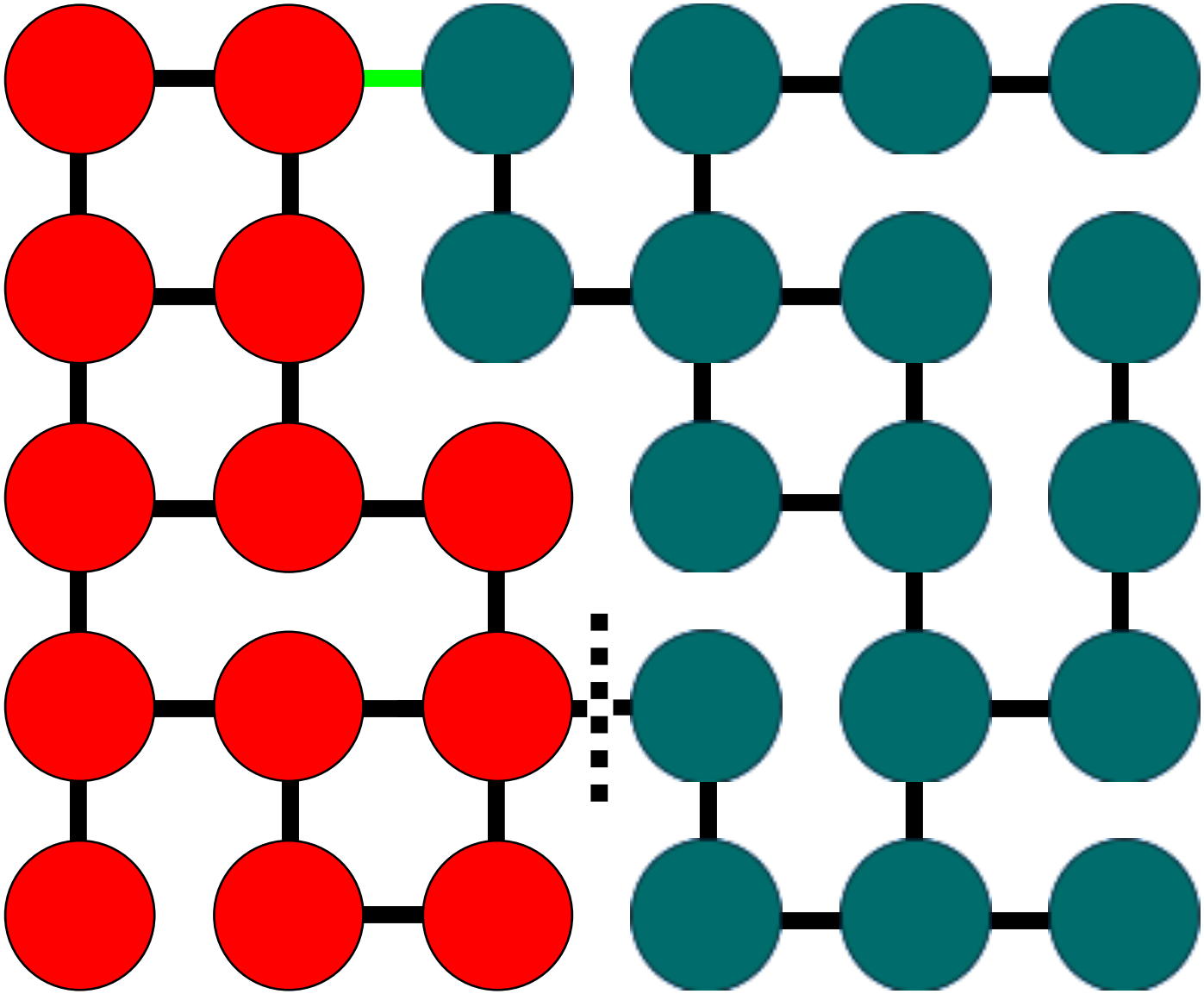
SW Cluster decomposition



Markov Chain Monte Carlo Simulation for the RCM

- Is it possible to create new configurations by doing **only one** edge operation?
- **Yes**: "Monte Carlo study of weighted percolation clusters relevant to the Potts model", (M. Sweeny, 1983, PRB Vol. 27/Issue 7)
- Insertion/Deletion of bonds as update operation
- Acceptance ratio (Metropolis) of trial move: $a = \min\left(1, v^{\Delta b} q^{\Delta k}\right)$
  - Non-local quantity Δk (Δb is trivial)
- Dynamical critical exponents z (integrated) for some observables smaller then for SW/CM algorithm
- Observables like the "sum of squared cluster sizes" $\mathcal{S}_2$ or the "size of the giant component" C_1 have $z \leq 0$ for $q \lesssim 2$ at $v_c = \sqrt{q}$: "**Critical Speeding Up**", (Deng et al., PRL 98 230602, 2007)

Connectivity problem ($\Delta b = -1, \Delta k = 0$)



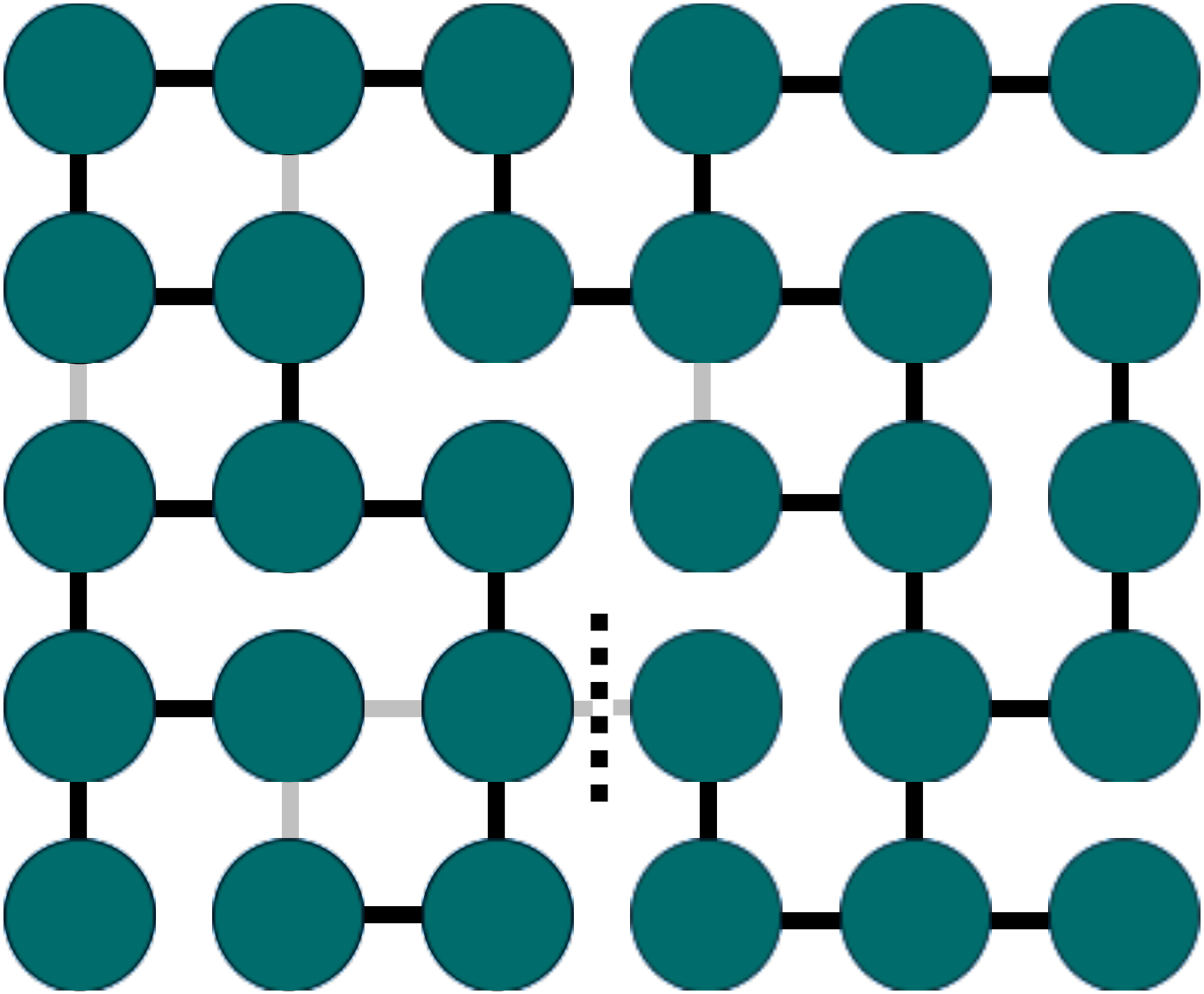
Connectivity

- Original implementation answers connectivity questions by traversal
  - $\mathcal{O}(N)$ worst-case (run-)time complexity for a bond operation
  - In contrast: Swendsen-Wang has $\mathcal{O}(1)$
- We used a "Poly-logarithmic fully-dynamic connectivity algorithm" (Holm et al., J. ACM Vol.48/Issue 4, 2001) which has $\mathcal{O}(\log(N)^2)$
  - In combination with smaller z: Sweeny's algorithm is asymptotically more efficient in terms of the runtime to create a statistically independent sample: $T/N \sim \tau\bar{t}$
- In practise also constants have to be considered, because they determine crossover

Poly-logarithmic fully-dynamic connectivity algorithm

- Maintain spanning forest (minimal set of edges with the same connectivity information), i.e., separate edges into:
  - Tree-edges (TE) / Bridges
  - Non-tree edges (NTE) (Insertion into spanning tree creates a simple/fundamental cycle)
- Manipulations on spanning trees; augment to hold information about incident non-tree edges
- Separation allows "cheap" $\Delta k = 0$ determination for insertion/deletion of NTE's
- In case of deletion of TE's systematic search for possible reconnecting NTE's necessary
  - Achieved by introduction of edge levels $l(e) = 0, ..., l_{\max} = \lfloor log(N) \rfloor$
  - A spanning tree at level i includes edges with level $l(e) \geq i$ thus $\mathcal{F} \equiv \mathcal{F}_0 \supseteq \cdots \supseteq \mathcal{F}_{l_{max}}$

Edge separation ($\Delta b = -1, \Delta k = 0$)



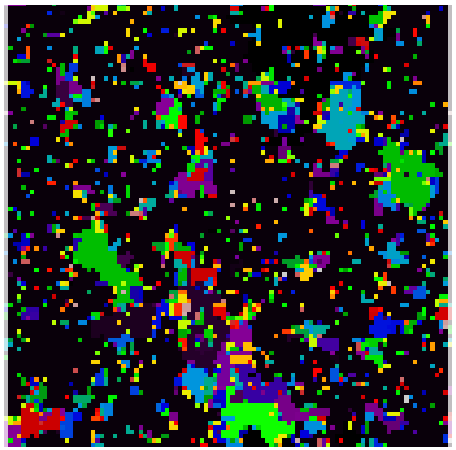
Poly-logarithmic fully-dynamic connectivity algorithm

- Edge levels are not static; (partial) modifications after every update operation with respect to two invariants which assure runtime bounds:
  - $\mathcal{F}$ is a maximal spanning tree (with respect to edge level): After deletion of e search for replacement edge only necessary at levels $i \leq l(e)$
  - A spanning tree at level i has size $\leq \lfloor \frac{N}{2^i} \rfloor$
- Intuition:
  - The higher the level of an edge, the denser the component in which it is contained
  - Keep important edges at low levels and "push" unimportant to higher levels
- Spanning trees stored in ET-Trees, which are linearised Eulerian paths/tours of $\mathcal{F}$

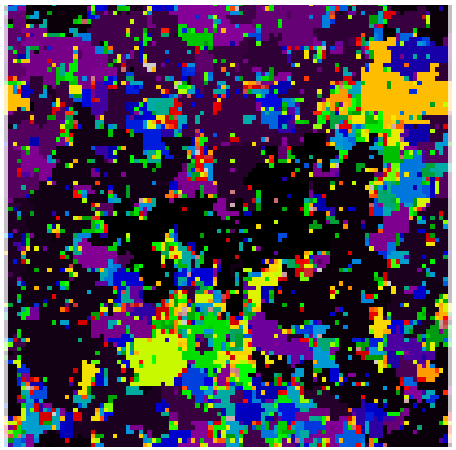
Illustration of edge level

(Simulation performed at $q = 2$ and $v_c = \sqrt{2}$)

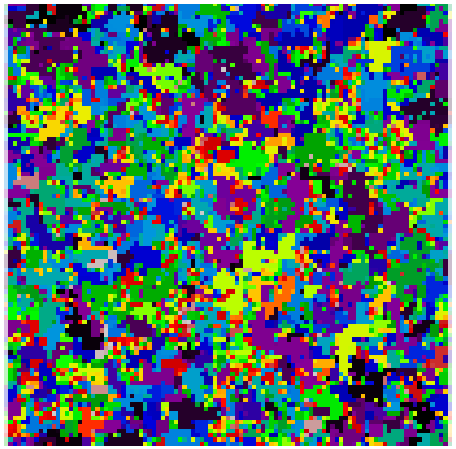
Level 0



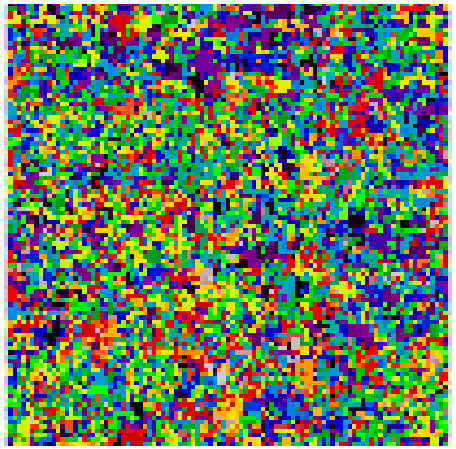
Level 2



Level 4

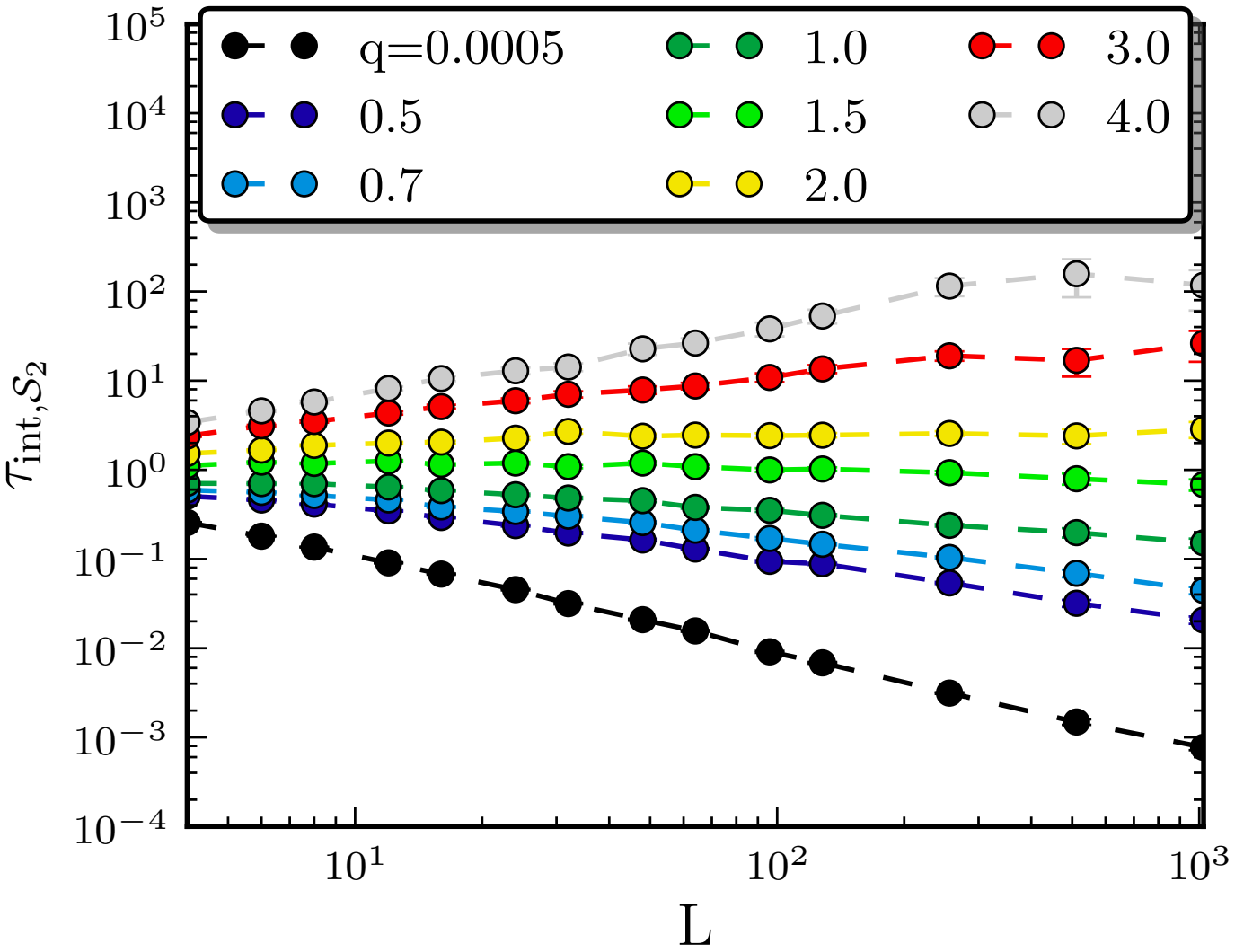


Level 6

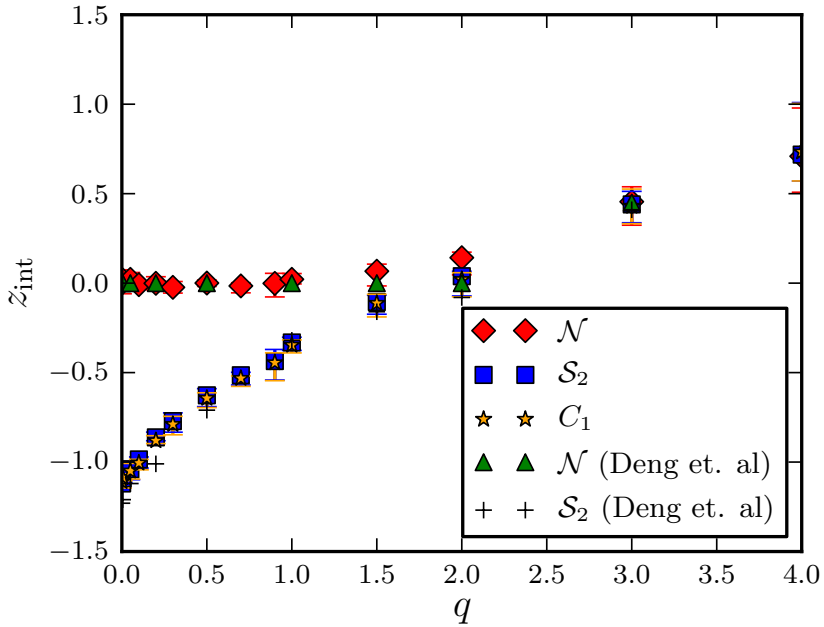


Integrated autocorrelation time for $\mathcal{S}_2$ observable

(All simulations performed at $v_c = \sqrt{q}$)



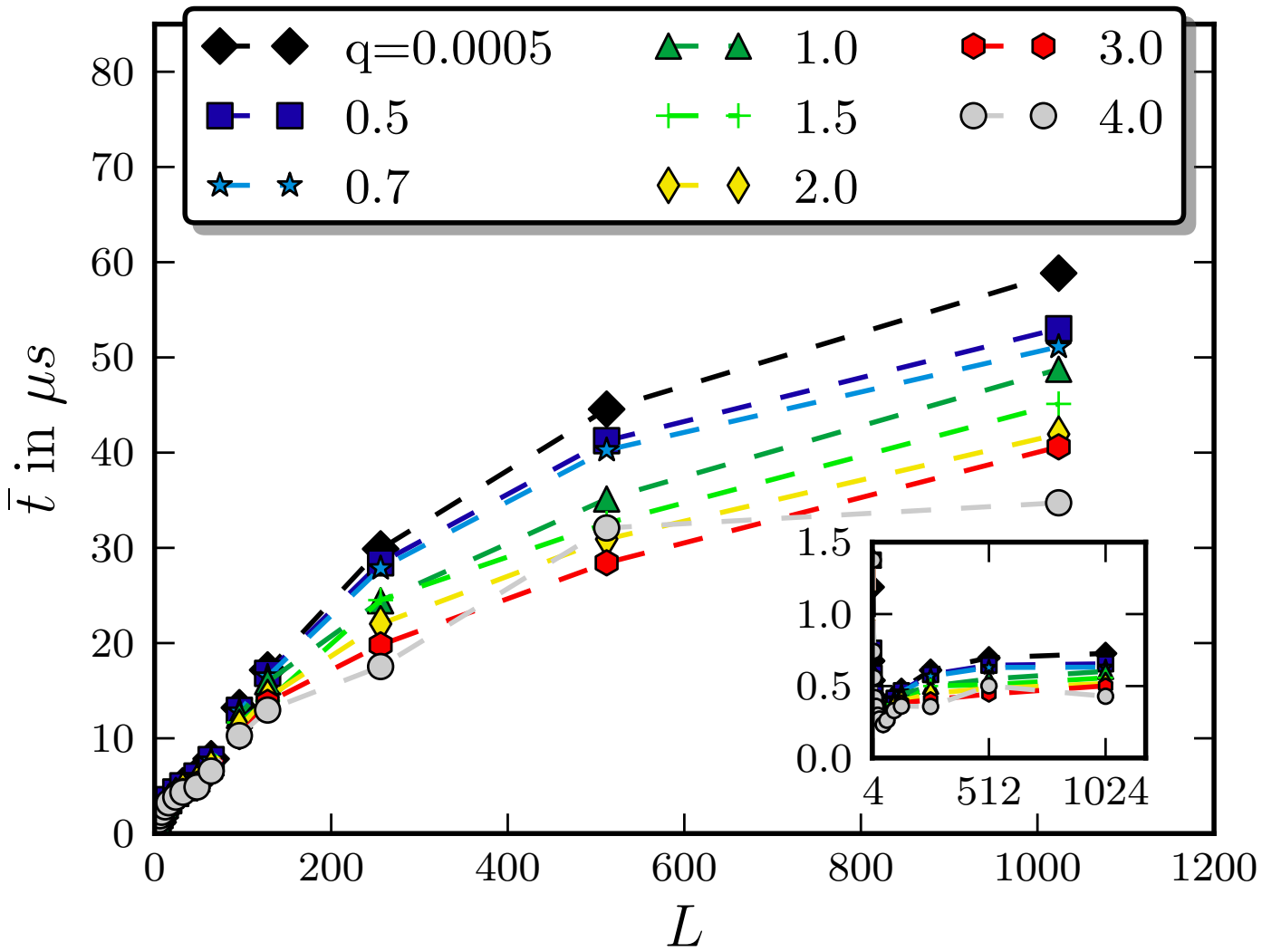
Results: $z_{int}(q)$ at $v_c = \sqrt{q}$



- Two-time scale-ansatz for normalized auto-correlation function $\rho_{\mathcal{S}_2}(t) = f(t/L^w)g(t/L^{d+z_{exp}})$
- $w = d - d_{\rm red}$, $d_{\rm red}$ "red"-bond fractal dimension
- Decorrelation strongest for fragile clusters, namely for largest $d_{\rm red}$ which is the case for small q

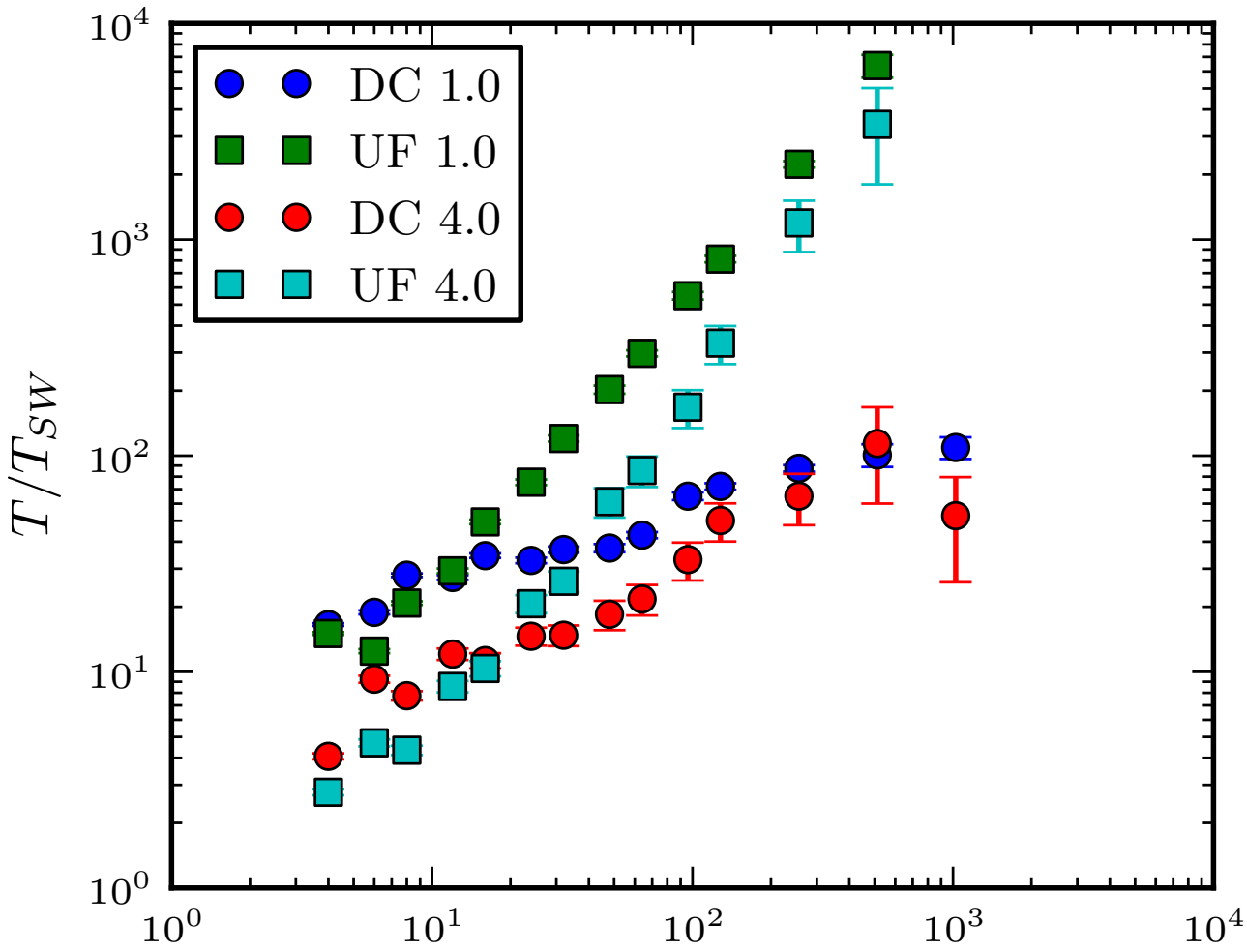
Results: Average runtime per edge operation

Inset shows $\frac{\bar{t}}{\log(L)^2}$ vs L



Results: Effective runtime for $\mathcal{S}_2$ samples

(UF stands for Union-Find implementation with interleaved Breadth First Search); Measured relative to Swendsen-Wang



Conclusions

- In an asymptotic sense most efficient (in terms of statistical independent samples) way of simulating the Potts model
- Space complexity is $\mathcal{O}(N \log(N))$ with large constants; Simulation of $L = 1024$ system has approx 2.5 GB memory requirement
- Only MCMC algorithm which allows for simulations $q < 1$
- Easily adapted to different dimensions and lattices (in general graphs)
- Potential for heuristics, e.g. to reduce space complexity ($l_{max} \to \lfloor l_{max}/2 \rfloor$)
- Further interesting applications:
  - Generalized ensemble simulations of the RCM
  - Droplet simulations
  - Identification of loop configurations

Thank you for your attention!