Quantum Gravity Simulations with MIXMAX Generator

Andrzej Görlich

Institute of Physics, Jagiellonian University, Poland

Athens, July 4th, 2018

Introduction to **C**ausal **D**ynamical **T**riangulations

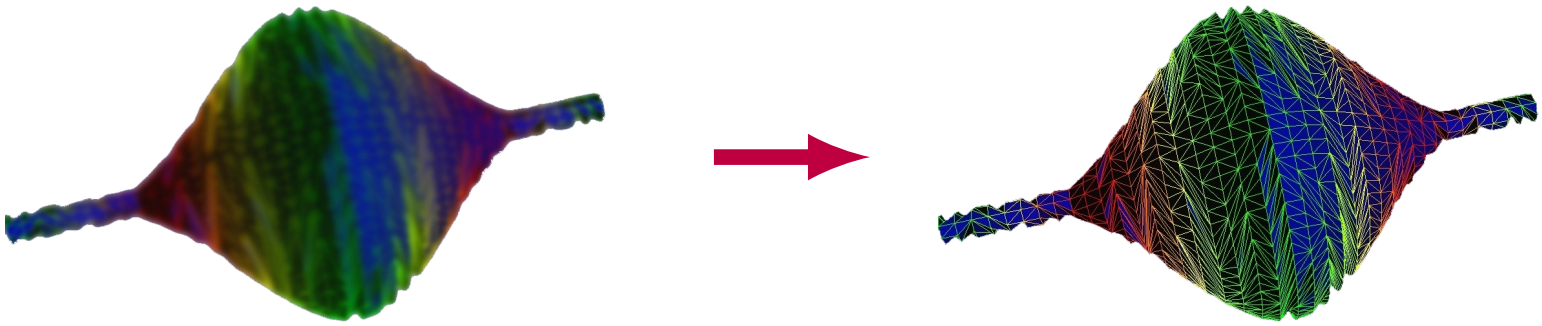
What is Causal Dynamical Triangulations?

Causal Dynamical Triangulations (CDT) is a background independent approach to quantum gravity. It provides a lattice regularization of the formal gravitational path integral via a sum over causal triangulations.

$$\int \mathrm{D}[g] e^{iS^{EH}[g]} \quad \rightarrow \quad \sum_{\mathcal{T}} e^{-S^{R}[\mathcal{T}]}$$

continuous

discrete



MIXMAX and CDT

For the last three years we have been using the standard version of the MIXMAX pseudorandom number generator with $N = 256$, $s = -1, m = 0$ (and discarding the first component) in our CDT code.
It shows many advantages:

- Great quality (passes Big Crush from *TestU01*)
- Very fast (although it's a small part, $< 0.5\%$ CPU time)
- Convenient vector size ($255 = 3 \times 85$)
- Convenient output size (61-bits, `double`)

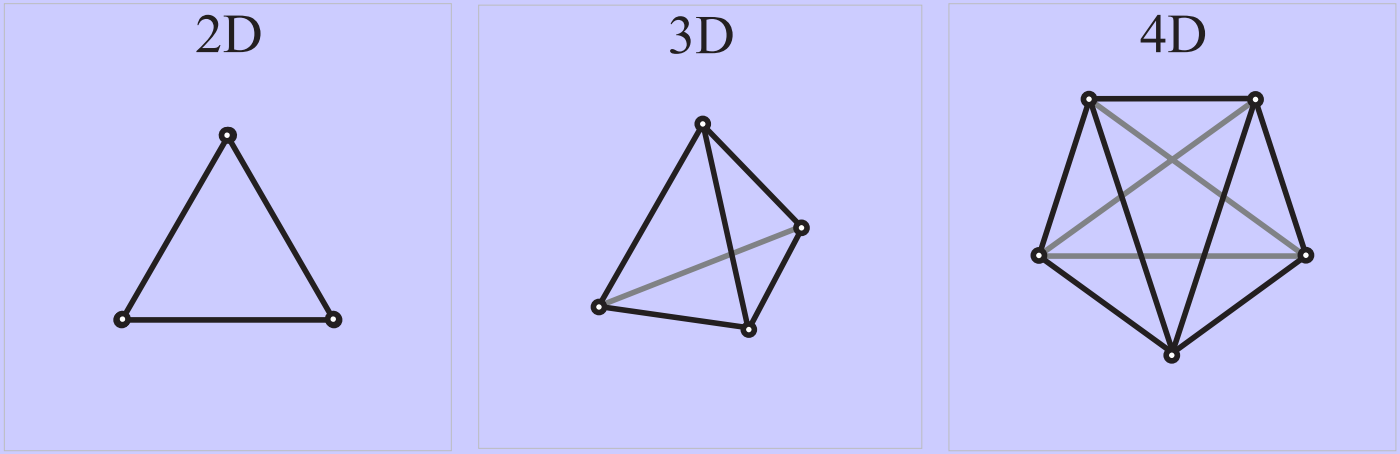
It just works!

We encountered no issues which could be related to the RNG.

Regularization via triangulation

- A *four-dimensional* simplicial manifold is obtained by gluing pairs of four-simplices along their three-faces (tetrahedra).
- The metric is **flat** inside each 4-simplex.
- Curvature is localized at triangles.
- Global proper-time foliation. Spatial states are *three-dimensional* geometries. Discretized states are build from equilateral **tetrahedra**.
- Length of time links a_t and space links a_s is constant.

Fundamental building blocks



Regge action

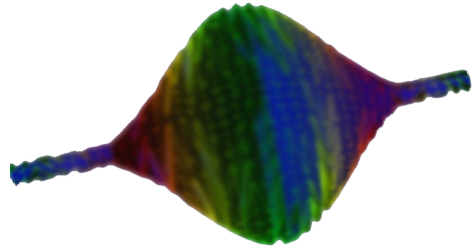
The Einstein-Hilbert action has a natural realization on piecewise linear geometries called Regge action

$$S^E[g] = -\frac{1}{G}\int \mathrm{d}t \int \mathrm{d}^D x \sqrt{g}(R - 2\Lambda)$$

N_0 number of vertices
N_4 number of simplices
N_{14} number of simplices of type $\{1,4\}$
K_0 K_4 Δ bare coupling constants $(G, \Lambda, a_t/a_s)$

The partition function

$$\int \mathrm{D}[g] e^{iS^{EH}[g]} \quad \rightarrow \quad \sum_{\mathcal{T}} e^{-S^R[\mathcal{T}]}$$



Regge action

The Einstein-Hilbert action has a natural realization on piecewise linear geometries called Regge action

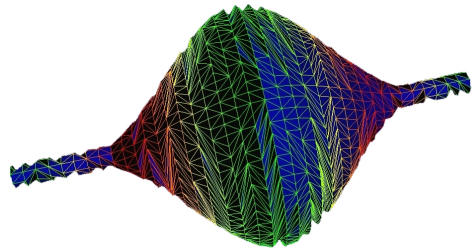
$$S^R[\mathcal{T}] = -K_0 N_0 + K_4 N_4 + \Delta(N_{14} - 6N_0)$$

N_0 number of vertices

N_4 number of simplices

N_{14} number of simplices of type $\{1, 4\}$

K_0 K_4 Δ bare coupling constants $(G, \Lambda, a_t/a_s)$



Causal Dynamical Triangulations

- The partition function of quantum gravity is defined as a formal integral over all geometries weighted by the Einstein-Hilbert action.

$$Z = \int \mathrm{D}[g] e^{iS^{EH}[g]}$$

- To make sense of the gravitational path integral one uses the standard method of regularization - discretization.
- The path integral is written as a nonperturbative sum over all causal triangulations $\mathcal{T}$.
- Wick rotation is well defined due to global proper-time foliation. $(a_t \to ia_t)$
- Using **Monte Carlo** techniques we can approximate expectation values of observables.

Causal Dynamical Triangulations

- The partition function of quantum gravity is defined as a formal integral over all geometries weighted by the Einstein-Hilbert action.

$$Z = \sum_{\mathcal{T}} e^{iS^R[g[\mathcal{T}]]}$$

- To make sense of the gravitational path integral one uses the standard method of regularization - discretization.
- The path integral is written as a nonperturbative sum over all causal triangulations $\mathcal{T}$.
- Wick rotation is well defined due to global proper-time foliation. $(a_t \to ia_t)$
- Using **Monte Carlo** techniques we can approximate expectation values of observables.

Causal Dynamical Triangulations

- The partition function of quantum gravity is defined as a formal integral over all geometries weighted by the Einstein-Hilbert action.

$$Z = \sum_{\mathcal{T}} e^{-S^R[\mathcal{T}]}$$

- To make sense of the gravitational path integral one uses the standard method of regularization - discretization.
- The path integral is written as a nonperturbative sum over all causal triangulations $\mathcal{T}$.
- Wick rotation is well defined due to global proper-time foliation. $(a_t \to i a_t)$
- Using **Monte Carlo** techniques we can approximate expectation values of observables.

Causal Dynamical Triangulations

- The partition function of quantum gravity is defined as a formal integral over all geometries weighted by the Einstein-Hilbert action.

$$Z = \sum_{\mathcal{T}} e^{-S^R[\mathcal{T}]}$$

- To make sense of the gravitational path integral one uses the standard method of regularization - discretization.
- The path integral is written as a nonperturbative sum over all causal triangulations $\mathcal{T}$.
- Wick rotation is well defined due to global proper-time foliation. $(a_t \to i a_t)$
- Using **Monte Carlo** techniques we can approximate expectation values of observables.

Monte Carlo simulations - initial and minimal configuration

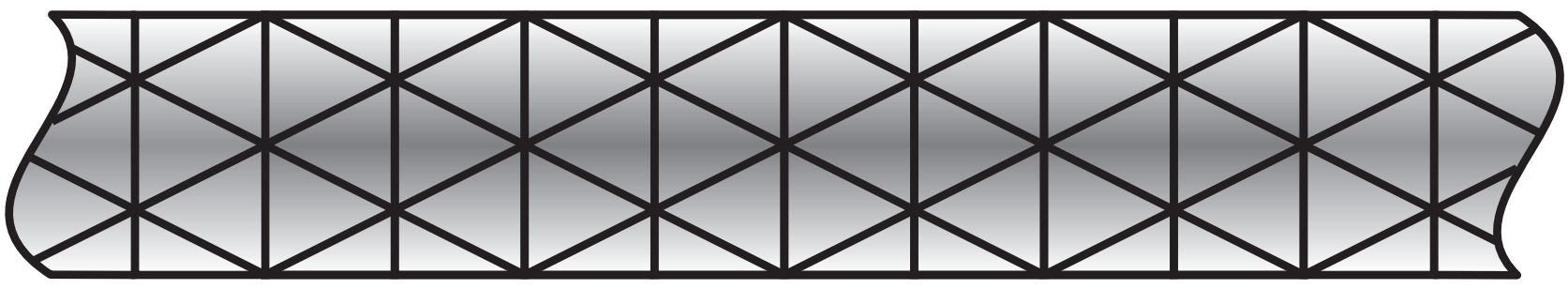
To start the simulations we need to construct an initial triangulation with the desired topology.

Spherical topology ($S^3 \times S^1$)

- **Minimal** triangulation: **five** tetrahedra.

Toroidal topology $T^3 \times S^1$

- Regular **initial** triangulation: **1024** four-simplices per slice
- **Minimal** triangulation: layered and interlaced structure, 15 vertices (not 3^3) and **90** tetrahedra.



Monte Carlo simulations - initial and minimal configuration

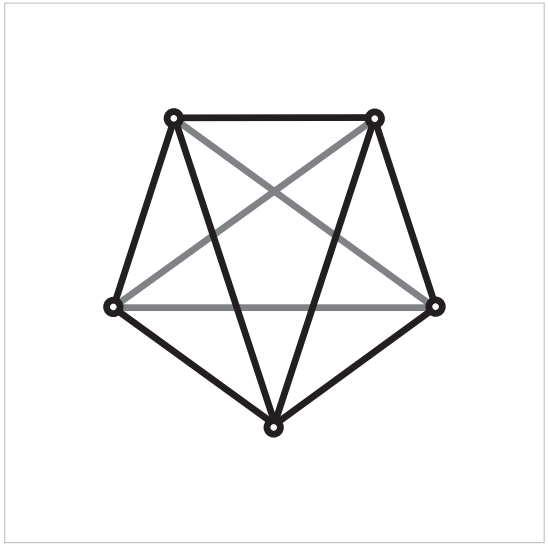
To start the simulations we need to construct an initial triangulation with the desired topology.

Spherical topology ($S^3 \times S^1$)

- **Minimal** triangulation: **five** tetrahedra.

Toroidal topology $T^3 \times S^1$

- Regular **initial** triangulation: **1024** four-simplices per slice
- **Minimal** triangulation: layered and interlaced structure, 15 vertices (not 3^3) and **90** tetrahedra.



Monte Carlo simulations - initial and minimal configuration

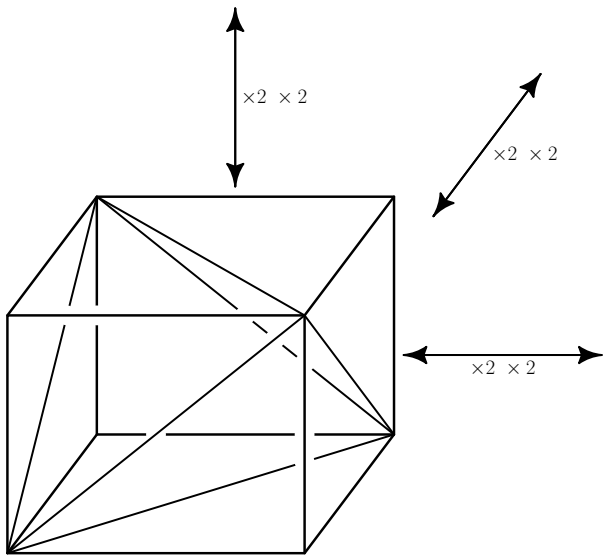
To start the simulations we need to construct an initial triangulation with the desired topology.

Spherical topology $(S^3 \times S^1)$

- **Minimal** triangulation: **five** tetrahedra.

Toroidal topology $T^3 \times S^1$

- Regular **initial** triangulation: **1024** four-simplices per slice
- **Minimal** triangulation: layered and interlaced structure, 15 vertices (not 3^3) and **90** tetrahedra.



Monte Carlo simulations - initial and minimal configuration

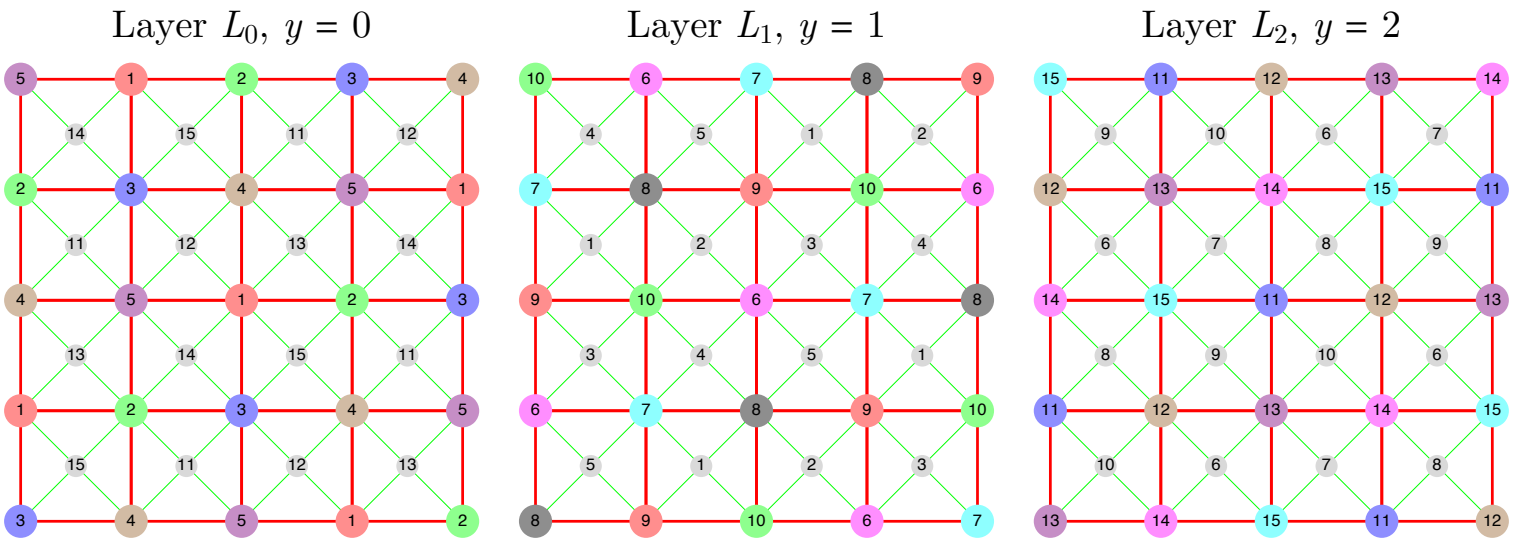
To start the simulations we need to construct an initial triangulation with the desired topology.

Spherical topology ($S^3 \times S^1$)

- **Minimal** triangulation: **five** tetrahedra.

Toroidal topology $T^3 \times S^1$

- Regular **initial** triangulation: **1024** four-simplices per slice
- **Minimal** triangulation: layered and interlaced structure, 15 vertices (not 3^3) and **90** tetrahedra.



Monte Carlo simulations - initial and minimal configuration

To start the simulations we need to construct an initial triangulation with the desired topology.

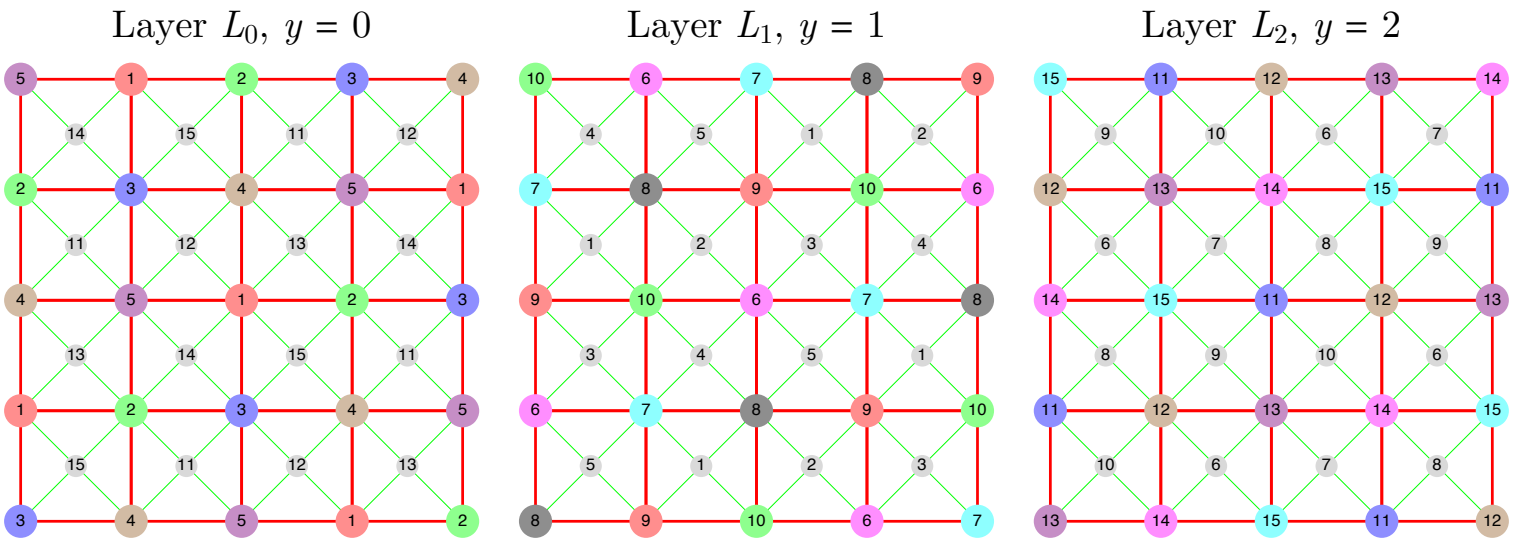
Spherical topology $(S^3 \times S^1)$

- **Minimal** triangulation: **five** tetrahedra.

Toroidal topology $T^3 \times S^1$

- Regular **initial** triangulation: **1024** four-simplices per slice
- **Minimal** triangulation, layered and interfaced structure, 15 vertices (not 3^3) and **90** tetrahedra.

Stronger finite size effects!

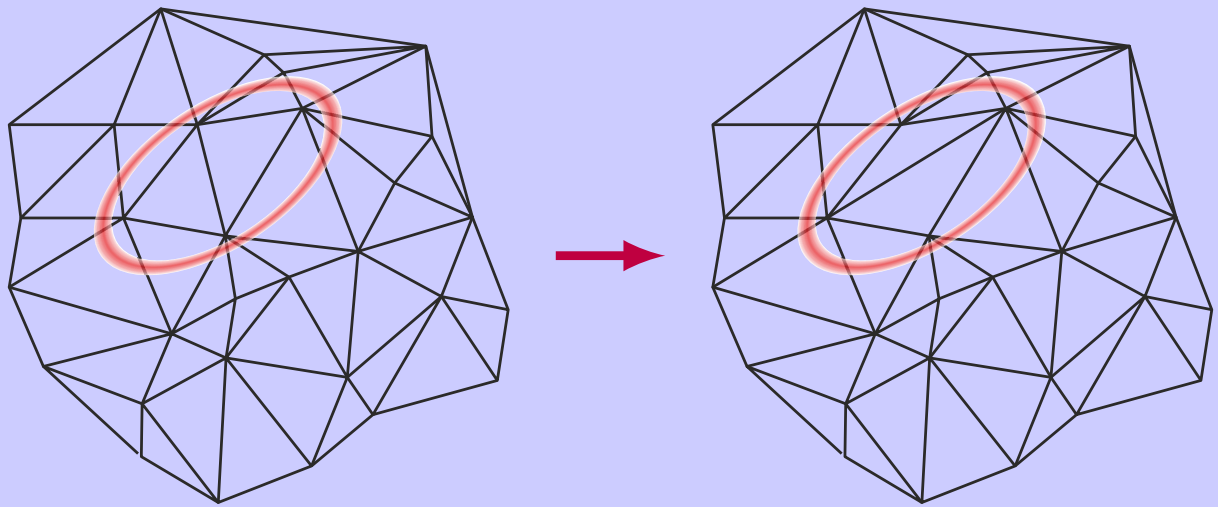


Monte Carlo simulations - Pachner moves

Random walk over configuration space:

- Ergodicity — all possible configurations can be generated by moves
- Fixed topology — moves don't change the topology
- Causality — moves preserve the foliation
- 4D CDT — set of 7 moves

Example of a 2D Monte Carlo move



Monte Carlo Markov Chain

- We perform a random walk in the phase-space of configurations (space of piecewise linear geometries).
- Each step is one of the 4D CDT moves.
- The weight (acceptance probability) $W(\mathcal{A} \to \mathcal{B})$ of a move from configuration $\mathcal{A}$ to $\mathcal{B}$ is determined (not uniquely) by the detailed balance condition:

$$P(\mathcal{A})W(\mathcal{A} \to \mathcal{B}) = P(\mathcal{B})W(\mathcal{B} \to \mathcal{A})$$

- The Monte Carlo algorithm ensures that we probe the configurations with the probability $P(\mathcal{A})$.
- After sufficiently long time, the configurations are independent.
- All we need, is the probability functional for configurations $P(\mathcal{A})$ up to the normalization (Regge action).

Dynamical lattice

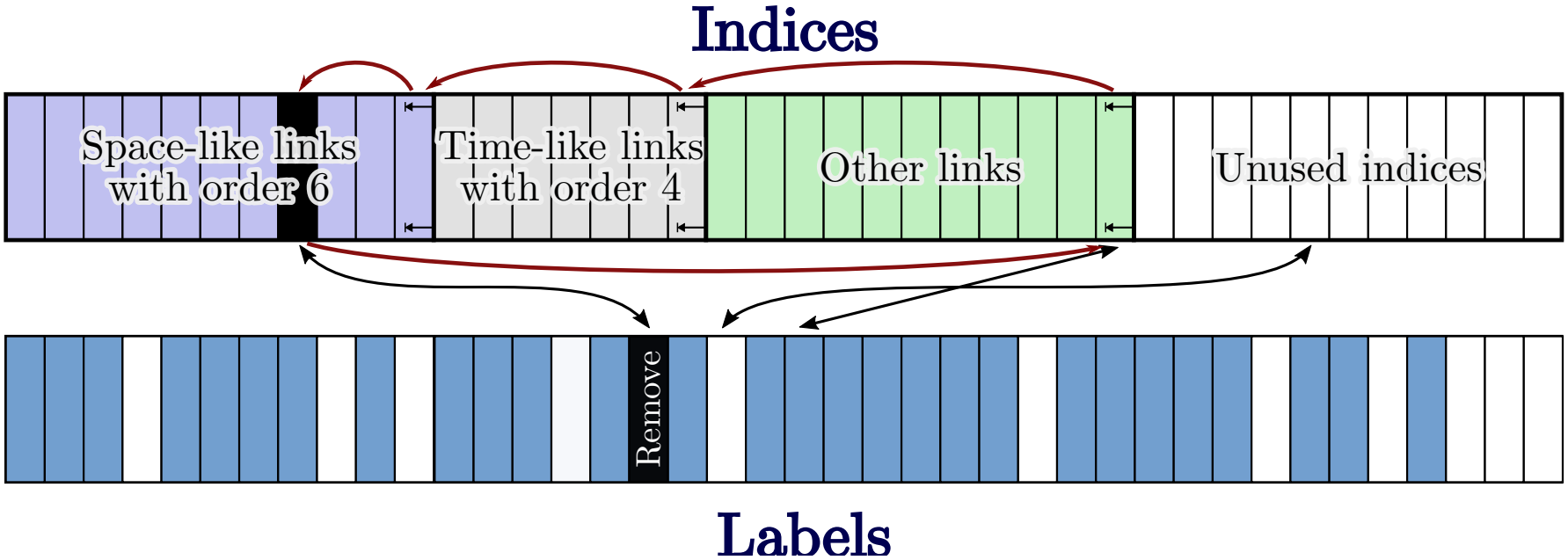
- The Monte Carlo moves alter the underlying spacetime lattice making it dynamical. The *(sub)-simplices* appear and disappear modifying the connectivity.
- To effectively deal with such system various techniques are used.

Basic stages of a single move attempt:

- Choose **which** move type (with adjusted probabilities).
- Choose **where** to make the move (index).
- Verify move validity (hash table) and calculate its weight (maximal information).
- Check **whether** to make the move (detailed balance). Reject the move, or accept the move.
- Update the triangulation if move was accepted.

Internals of CDT code: indices and labels

- Each object (point, link, triangle or simplex) is assigned two integer numbers: a **label** and an **index**.
- The label is fixed during the object lifetime allowing for its unique identification. Labels are not ordered continuously (removal).
- Indices are ordered continuously. They are split into few types to quickly find a valid move location. Index of a (sub)-simplex may change for various reasons (hidden randomness).

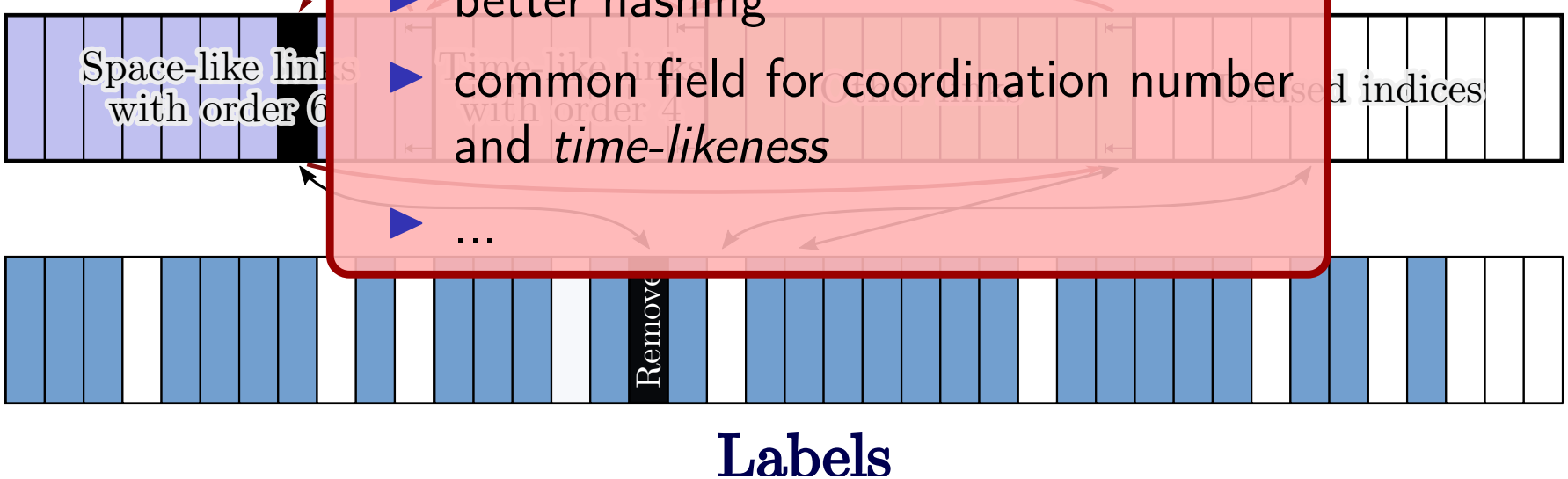


Internals of CDT code: indices and labels

- Each object (point, link, triangle or simplex) is assigned two integer numbers: a **label** and an **index**.
- The label is fixed during the object lifetime allowing for its unique identification. Labels are not ordered continuously (removal).
- Indices are ordered continuously. They are split into few types to quickly find a valid move location. Index of a (sub)-simplex may change for various reasons (hidden randomness).

New version ($>$ 30% faster)

- rewritten code
- simple, separate containers for sub-simplices
- tabelarization
- avoid `int` $\leftrightarrow$ f.p. conversions
- better hashing
- common field for coordination number and *time-likeness*
- ...



Internals of CDT code: data structures

- Simulations are limited by **CPU time** consumption, while **memory** usage is not a bottle-neck.
- To enhance the acceptance rate of the moves and gain fast access to necessary data **maximal information philosophy** is adopted.
- Various information about points, links, triangles and simplices (but not tetrahedra) are stored:

```
struct triangle
{
    int p[3];  /* Vertex labels */
    int n;     /* Coordination number */
    int s;     /* Simplex which owns this triangle */
    int i;     /* Index */
    int h;     /* Hash code of triangle */
    int next;  /* Next triangle in hash table list */
}
```

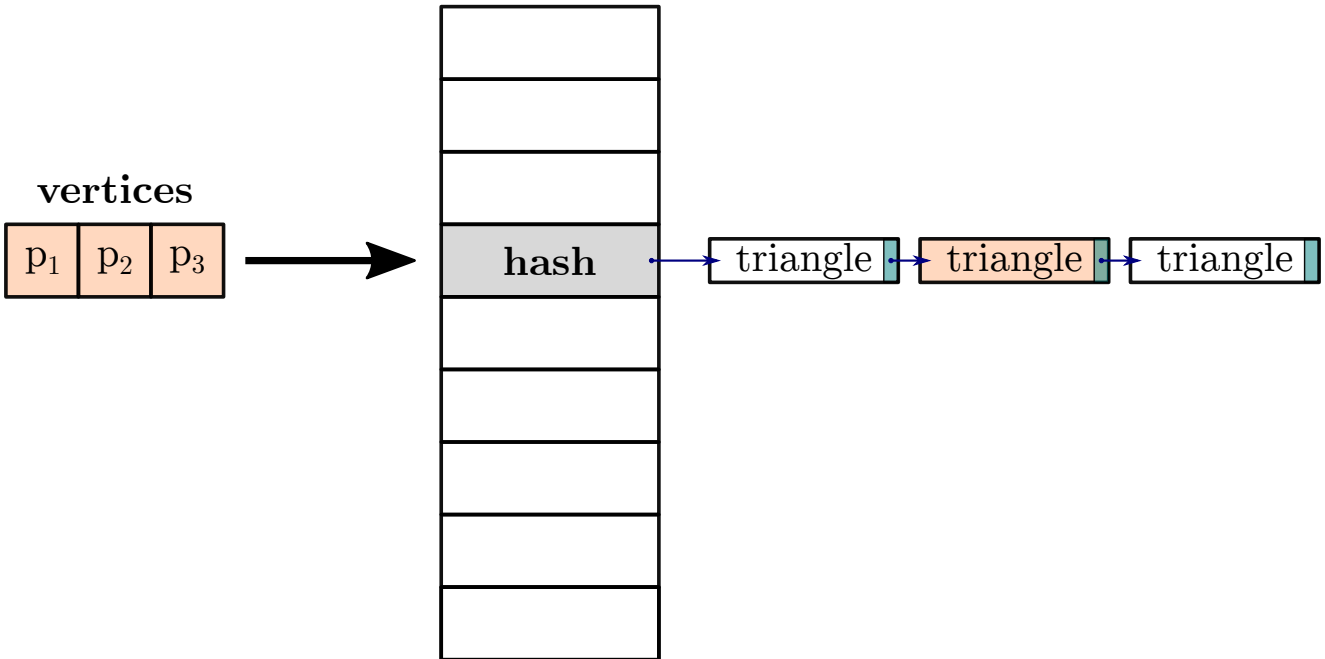
Internals of CDT code: hash tables

- To perform a fast check if a *link* or *triangle* with given vertices exists, a **hash table** is used.
- Verification of tetrahedron existence is more complicated and demands making a loop around a triangle.

Given labels of vertices `p[0] < p[1] < p[2]` we calculate the *hash code*,

```
h = (p[0] << 8) xor (p[1] << 4) xor p[2]
```

Triangles with the same *hash code* are stored in a singly-linked list.



Numerical setup

- **Monte Carlo** algorithm probes the space of configurations with the probability $P[\mathcal{T}] = \frac{1}{Z} e^{-S[\mathcal{T}]}$.
- To calculate the **expectation value of an observable**, the path integral is approximated by a sum over a finite set of Monte Carlo configurations.

$$
\begin{aligned}
\langle \mathcal{O}[g] \rangle &= \frac{1}{Z} \int \mathcal{D}[g] \mathcal{O}[g] e^{-S[g]} \\
&\downarrow \\
\langle \mathcal{O}[\mathcal{T}] \rangle &= \frac{1}{Z} \sum_{\mathcal{T}} \mathcal{O}[\mathcal{T}] e^{-S[\mathcal{T}]} \\
&\downarrow \\
\langle \mathcal{O}[\mathcal{T}] \rangle &\approx \frac{1}{K} \sum_{i=1}^{K} \mathcal{O}[\mathcal{T}^{(i)}]
\end{aligned}
$$

Numerical setup

- **Monte Carlo** algorithm probes the space of configurations with the probability $P[\mathcal{T}] = \frac{1}{Z} e^{-S[\mathcal{T}]}$.
- To calculate the **expectation value of an observable**, the path integral is approximated by a sum over a finite set of Monte Carlo configurations.

Example of an observable is n_t, the number of tetrahedra building slice t, $t = 1, \ldots, T$.

Configurations are generated with proper probability $P[\mathcal{T}] = \frac{1}{Z} e^{-S[\mathcal{T}]}$. Normalization Z is not needed.

$$\langle \mathcal{O}[g] \rangle = \frac{1}{Z} \int \mathcal{D}[g] \mathcal{O}[g] e^{-S[g]}$$

$$\downarrow$$

$$\langle \mathcal{O}[\mathcal{T}] \rangle = \frac{1}{Z} \sum_{\mathcal{T}} \mathcal{O}[\mathcal{T}] e^{-S[\mathcal{T}]}$$

$$\downarrow$$

$$\langle \mathcal{O}[\mathcal{T}] \rangle \approx \frac{1}{K} \sum_{i=1}^{K} \mathcal{O}[\mathcal{T}^{(i)}]$$

What can we measure?

From the computational point of view, the basic observables are the parameters which appear in the bare action:

N_0 number of vertices

N_{41} number of simplices of type $\{4, 1\}$

N_{32} number of simplices of type $\{3, 2\}$

The number of all remaining sub-simplices can be expressed as a linear combination of above parameters.

The simplest observable providing information about spacetime geometry is **spatial volume** n_t defined as a number of tetrahedra building a three-dimensional slice $t = 1 \ldots T$.

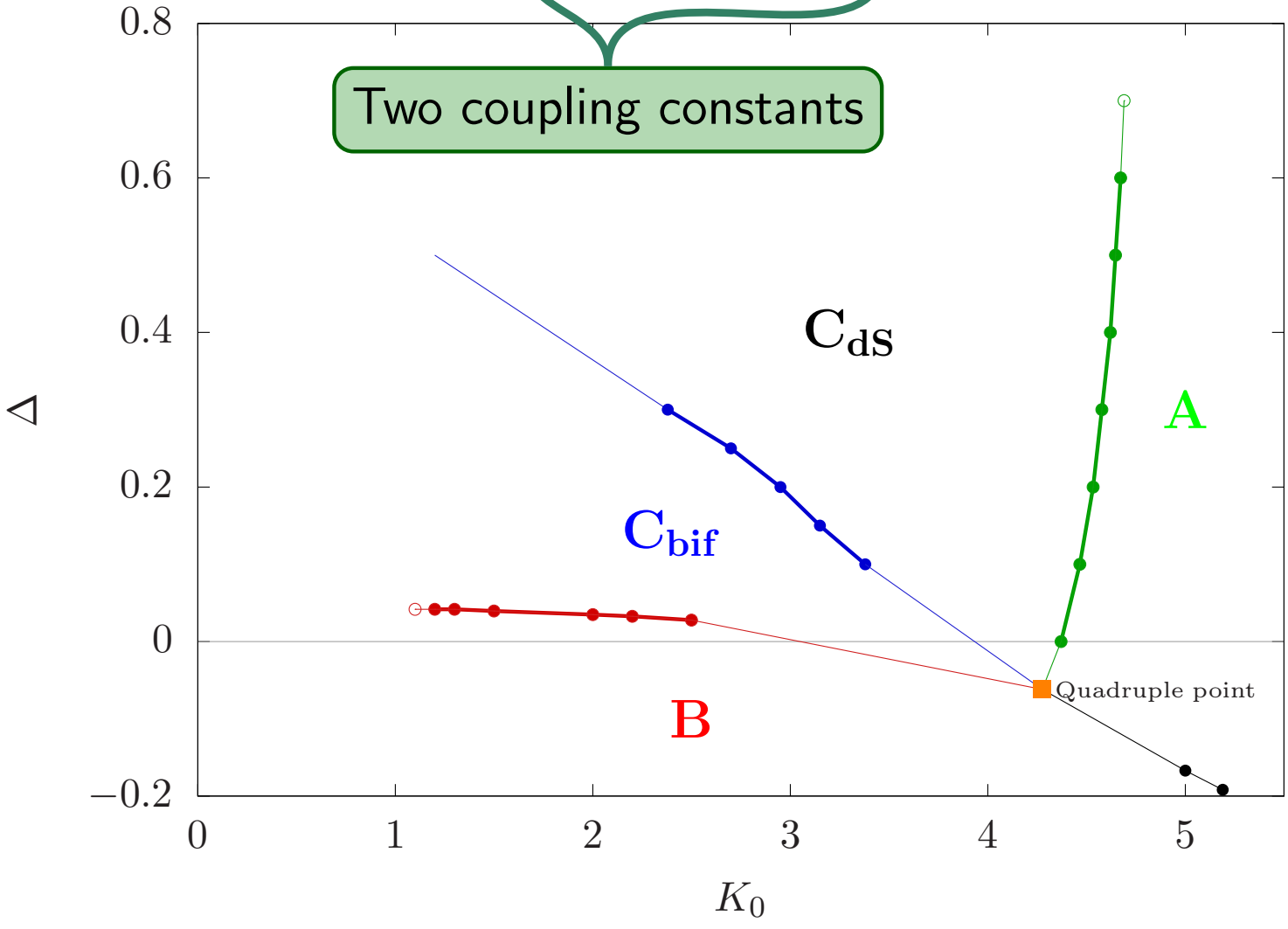
(Proper definition of e.g. curvature is non-trivial.)

Phase structure

Tuned cosmological constant

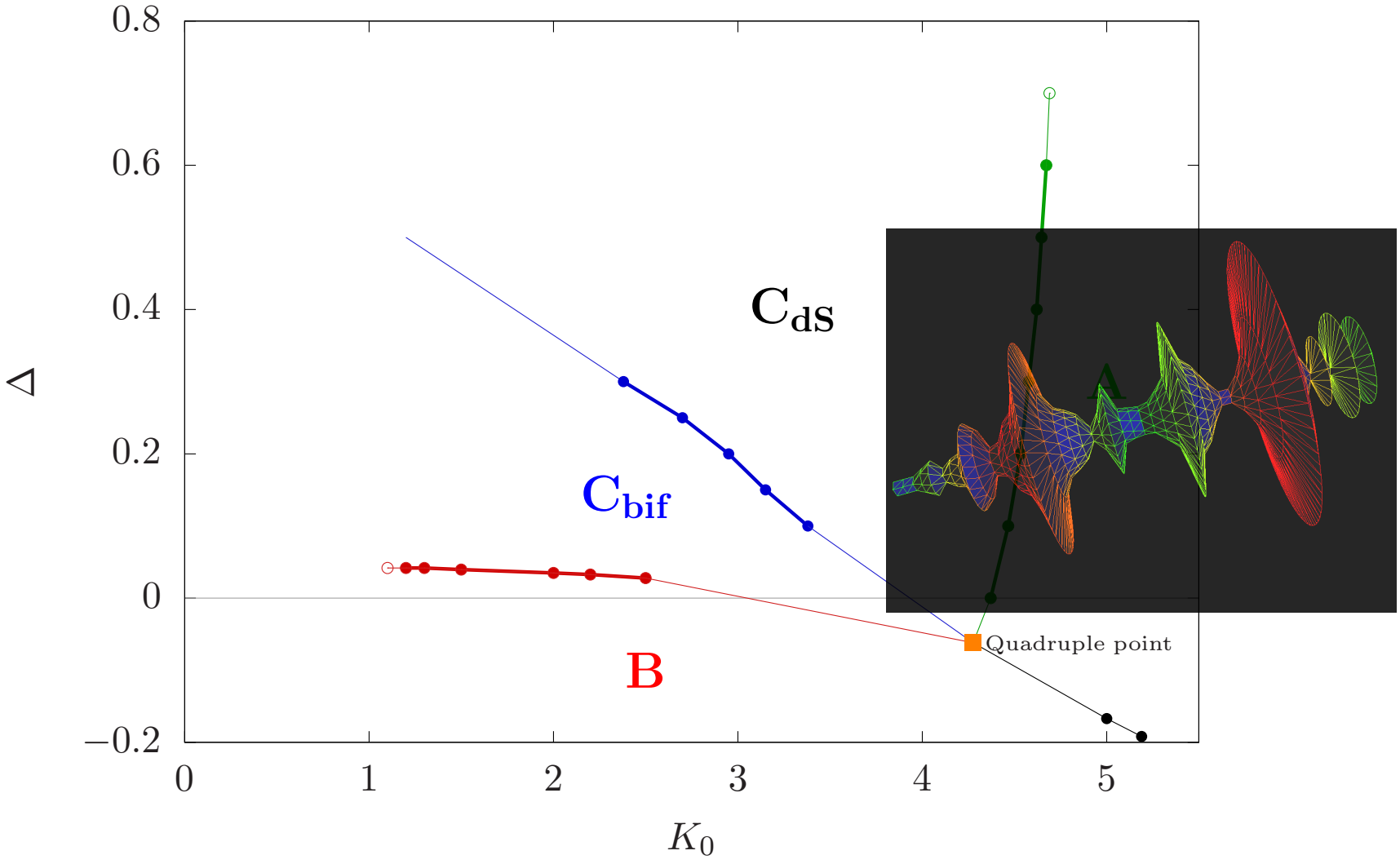
$$S[\mathcal{T}] = -K_0 N_0 + K_4 N_4 + \Delta(N_{14} - 6N_0)$$

Two coupling constants



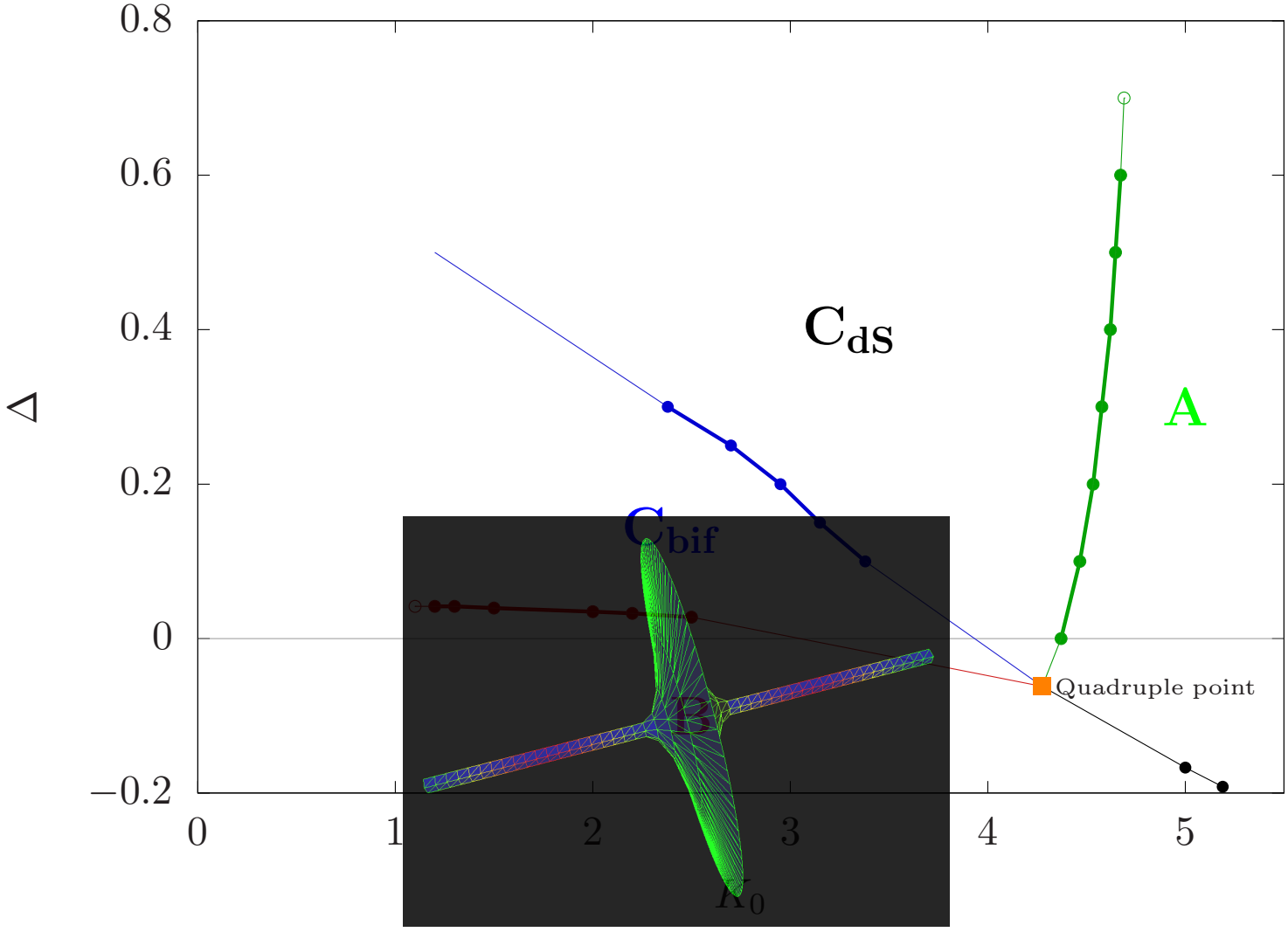
Phase structure

$$S[\mathcal{T}] = -K_0 N_0 + K_4 N_4 + \Delta(N_{14} - 6N_0)$$



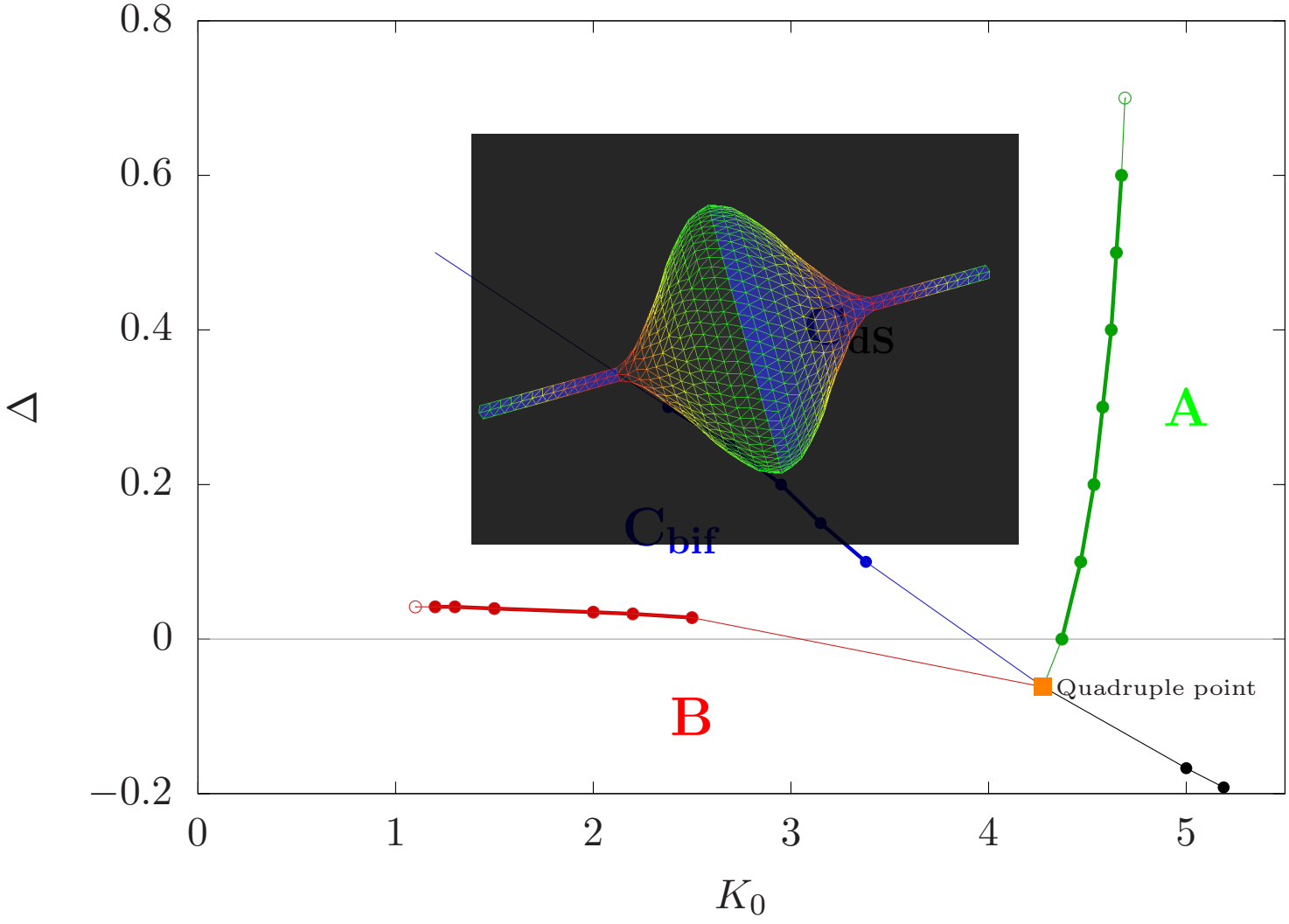
Phase structure

$$S[\mathcal{T}] = -K_0 N_0 + K_4 N_4 + \Delta(N_{14} - 6N_0)$$



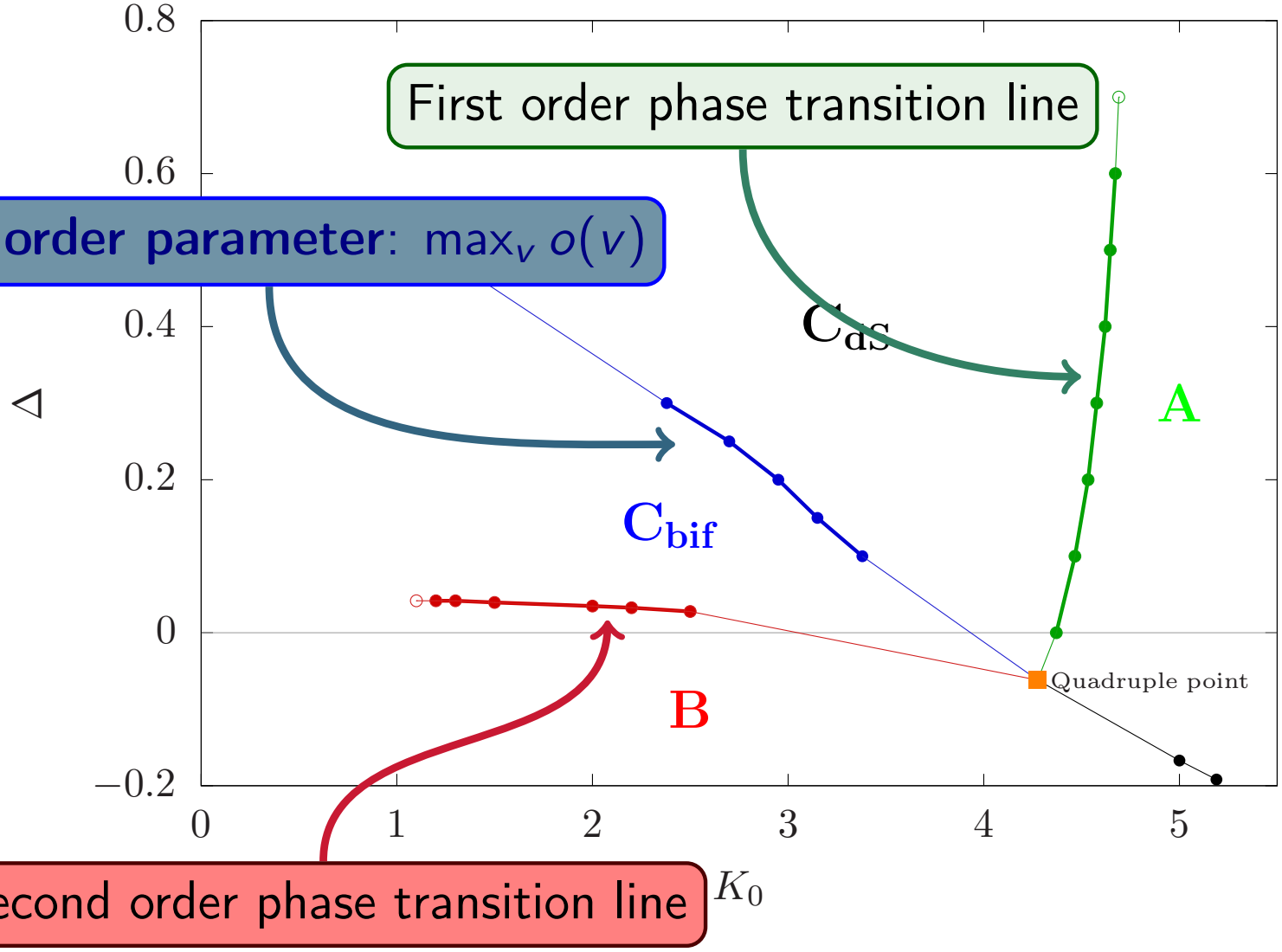
Phase structure

$$S[\mathcal{T}] = -K_0 N_0 + K_4 N_4 + \Delta(N_{14} - 6N_0)$$



Phase structure

$$S[\mathcal{T}] = -K_0 N_0 + K_4 N_4 + \Delta(N_{14} - 6N_0)$$

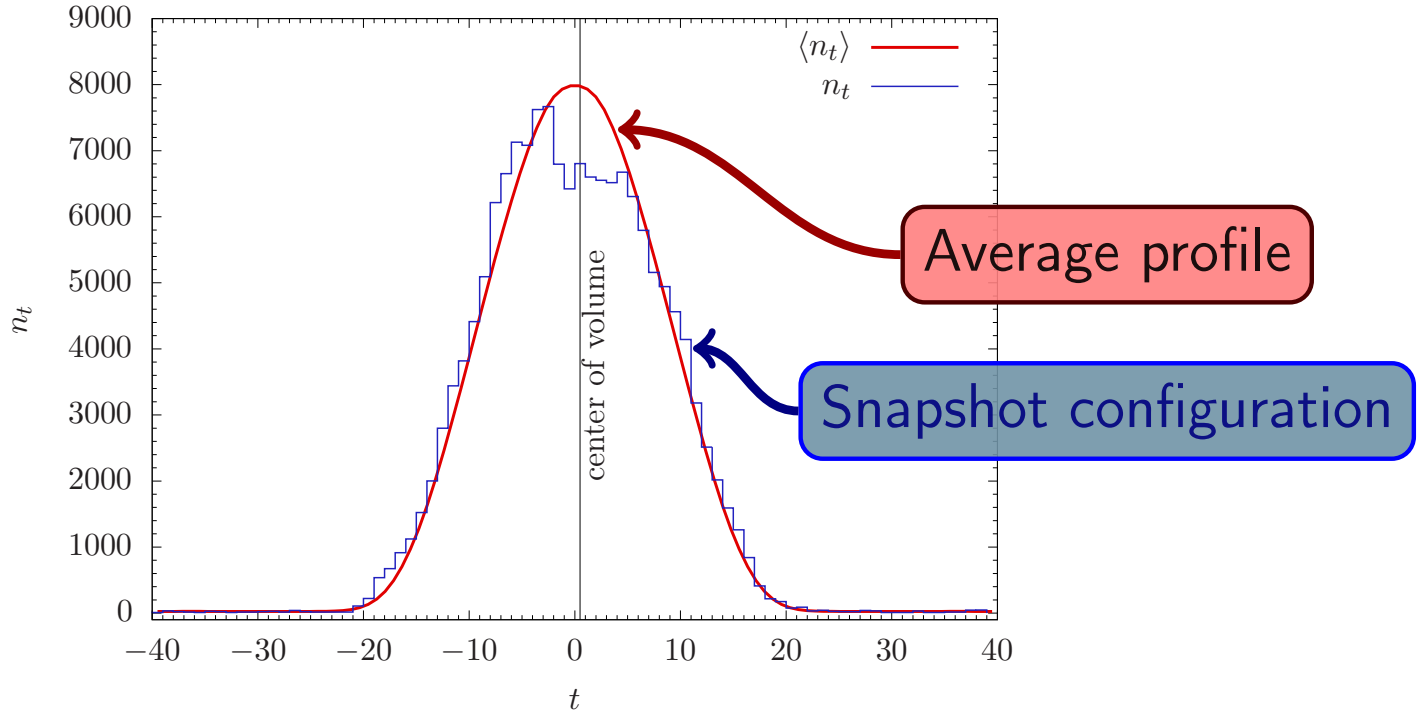


De Sitter phase - background geometry

- In phase **C** the time translation symmetry is spontaneously broken and the three-volume profile n_t is bell-shaped.
- The average volume $\langle n_t \rangle$ is with high accuracy given by formula

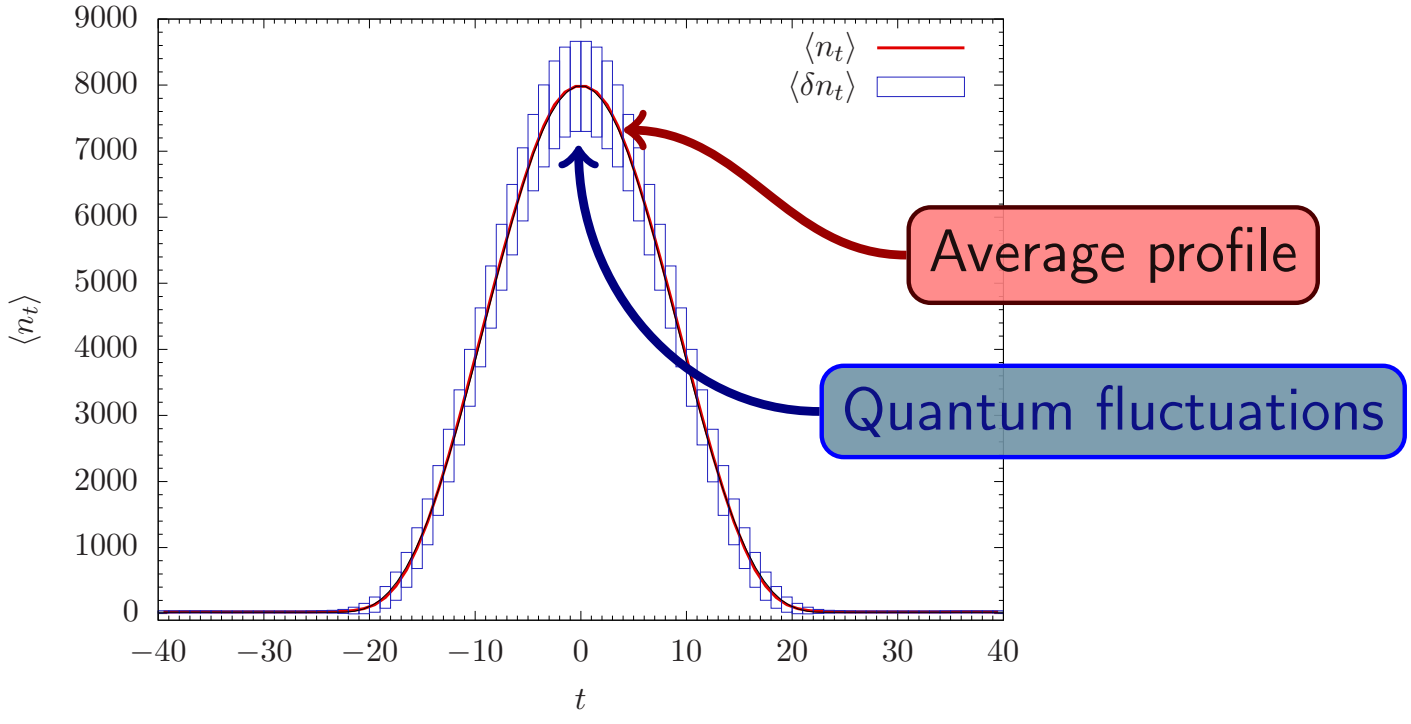
$$\langle n_t \rangle = H \cos^3\left(\frac{t}{W}\right)$$

a classical **vacuum solution**.



De Sitter phase - properties

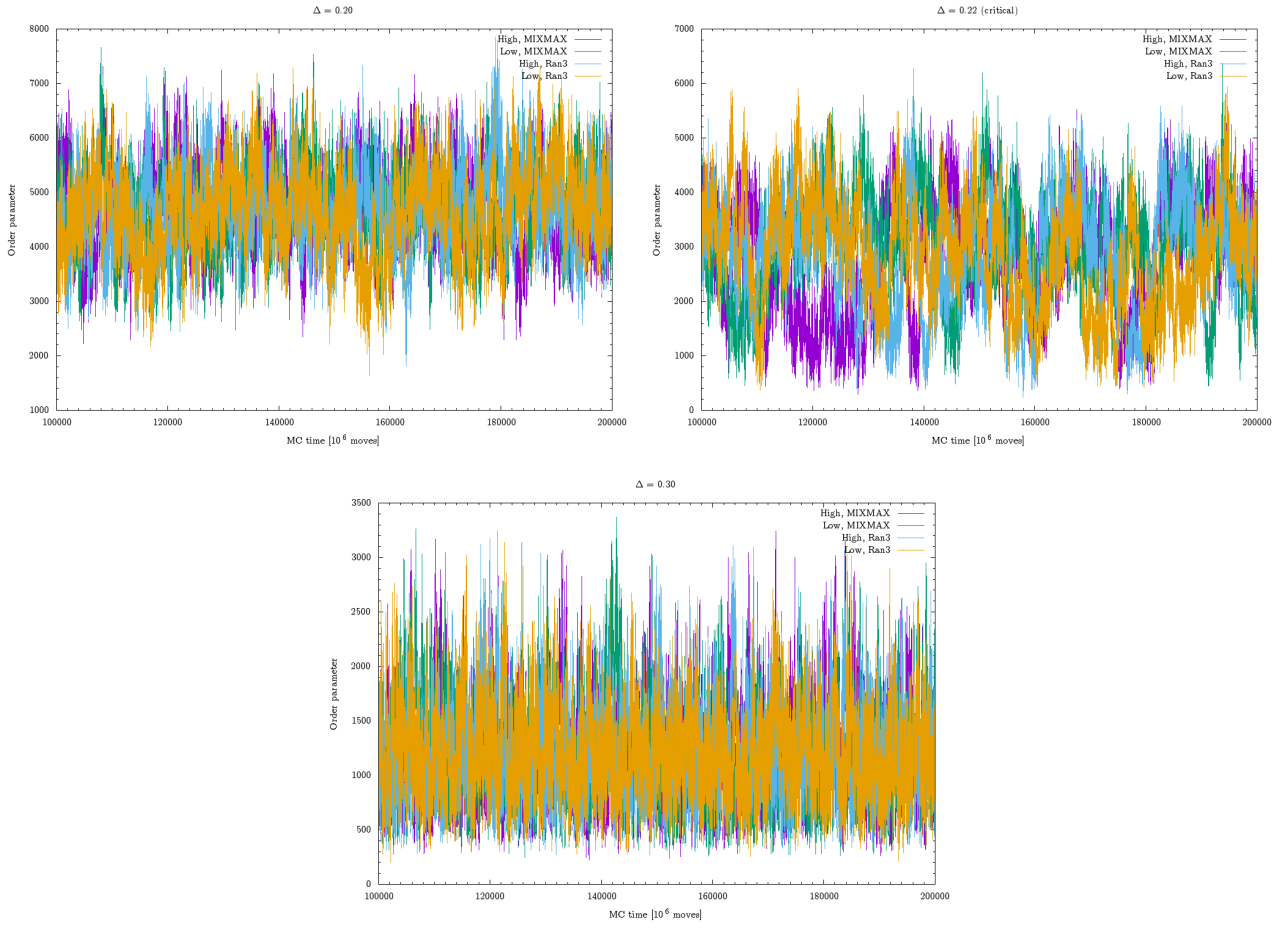
- Phase **C**, the so called de Sitter phase, is physically most interesting.
- A **four-dimensional** background geometry emerges dynamically. The background geometry corresponds to Euclidean **de Sitter** space (S^4), a classical **vacuum solution**.
- It is also possible to study **quantum fluctuations** around it.



Autocorrelation: MIXMAX vs Ran3

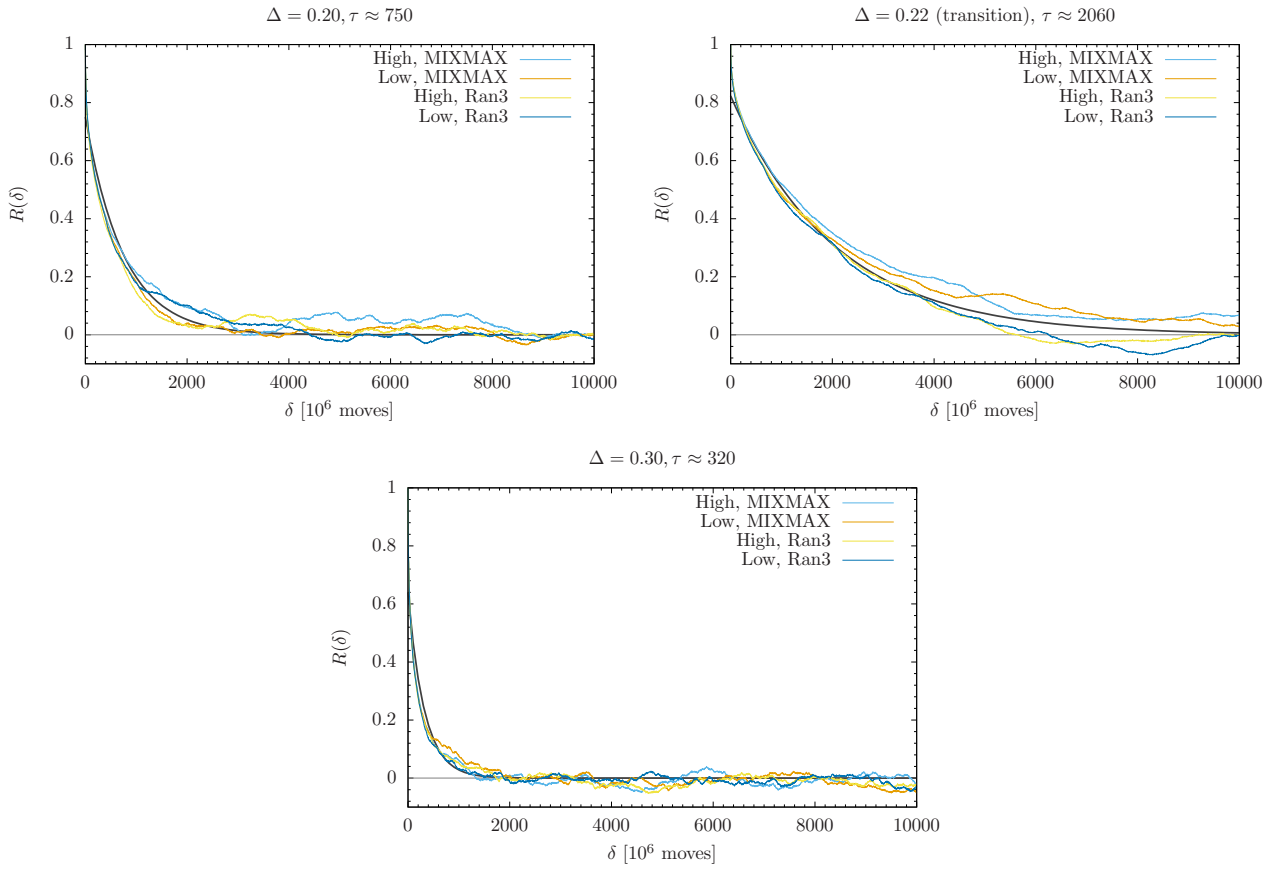
The MC moves introduce only small local changes and the configurations may be correlated for a long time.

$\Delta = 0.20$ $\Delta = 0.22$ (critical) $\Delta = 0.30$

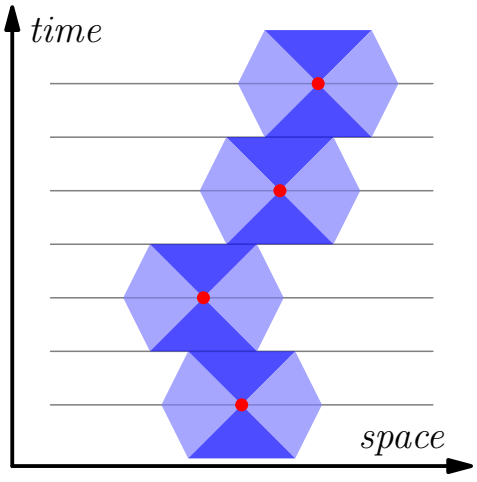


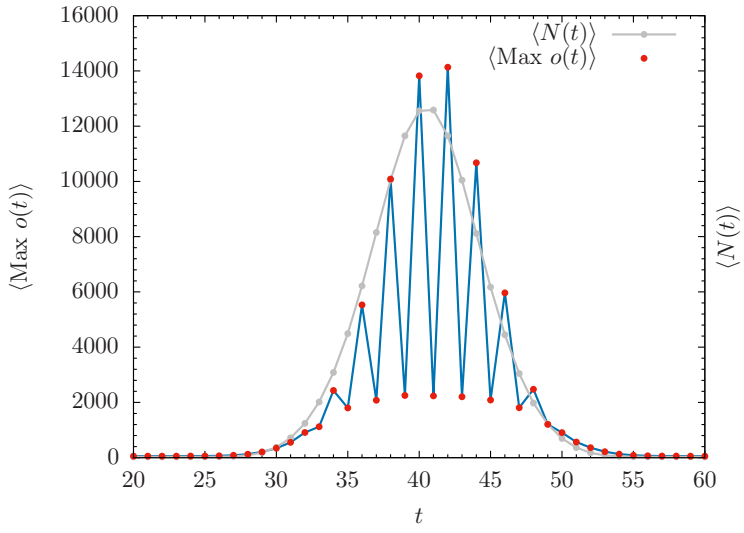
Autocorrelation time: MIXMAX vs Ran3

Often in the vicinity of phase transition, a phenomenon of critical slowing down is observed.
It takes longer for the system to thermalize, amplitude of fluctuations increases and the autocorrelation time grows.



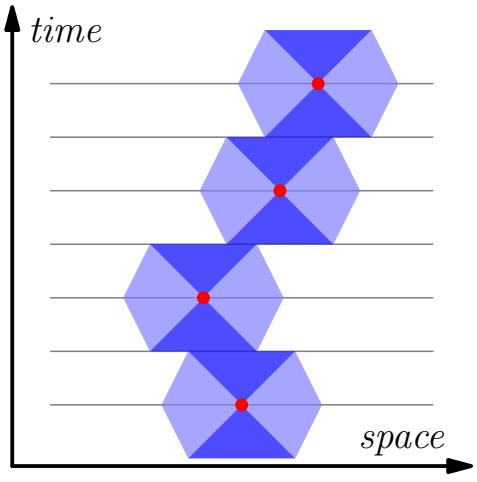
The new bifurcation phase

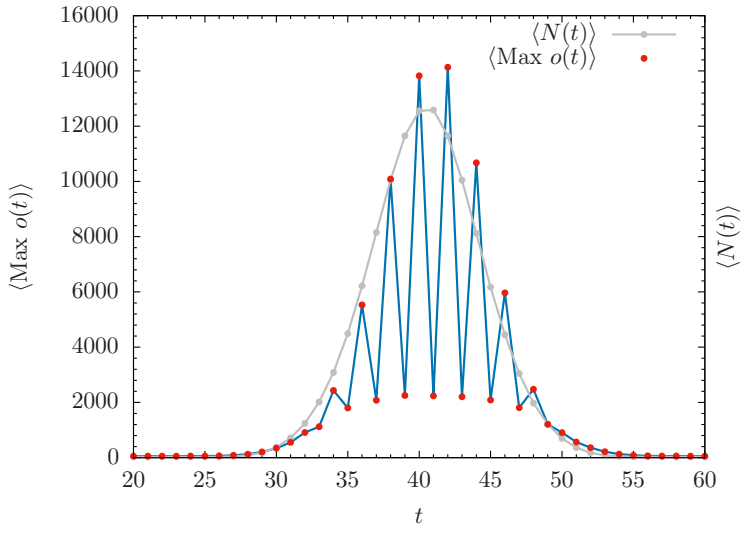




- The spatial volume profile $\langle n_t \rangle$ is similar as in phase C. But the agreement with $cos^3(t)$ is broken.
- The transfer matrix bifurcates into two branches. At some volume the kinetic term splits into a sum of two shifted Gausses.
- Every second slice **singular** vertices of very high order appear. New **order parameter**: $\max_v o(v)$.
- Not captured by global properties of the triangulation (N_0, N_{32}).

The new bifurcation phase





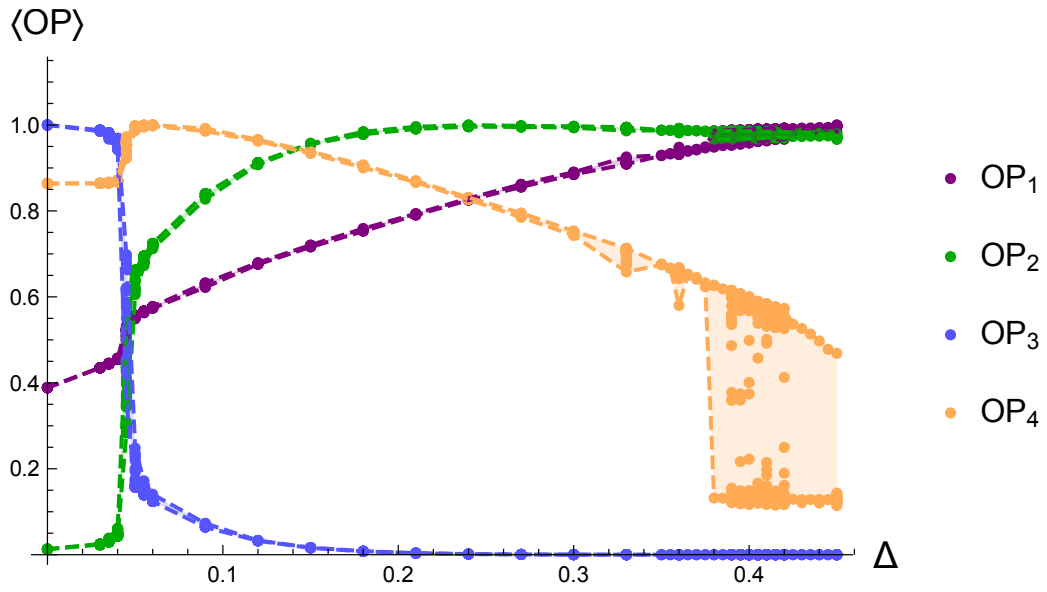
- The spatial volume profile $\langle n_t \rangle$ is similar as in phase C. But the agreement with $cos^3(t)$ is broken.
- The transfer matrix bifurcates into two branches. At some volume the kinetic term splits into a sum of two shifted Gausses.
- Every second slice **singular** vertices of very high order appear. New **order parameter**: $\max_v o(v)$.
- Not captured by global properties of the triangulation (N_0, N_{32}).

Difficult to dissolve

Toroidal topology - phase transitions

For the toroidal topology, the C_{bif}/C_{dS} transitions reveals a hysteresis and very long **thermalization time**.
Reason?

- Strong finite size effects
- Phase transition order depends on topology?



How to improve the simulations?

Motivation

- Limited *single-core* performance (gates, frequency, voltage, power)
- Cheap access to many cores

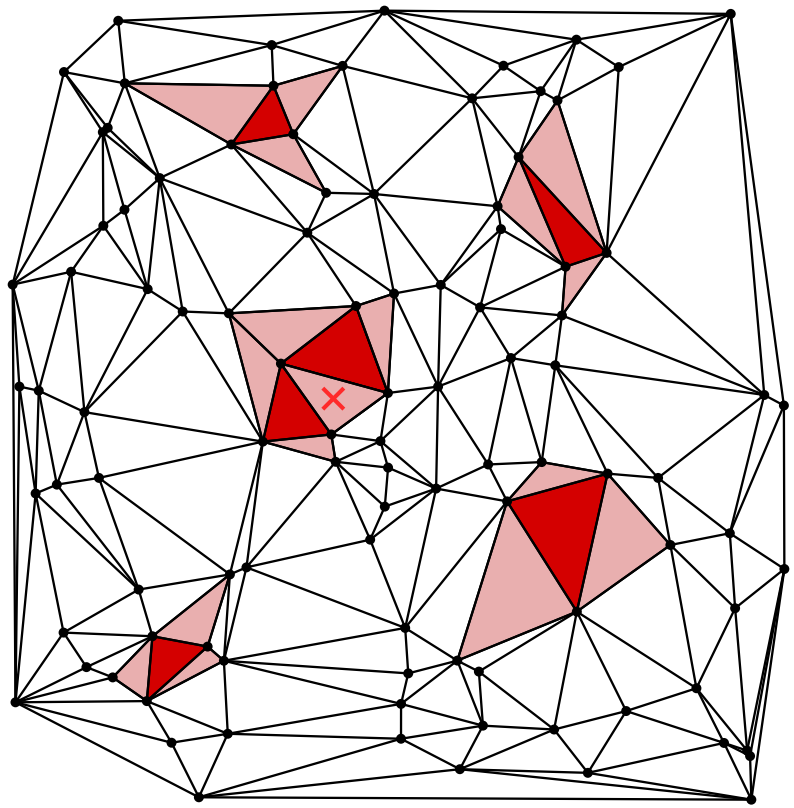
Naive parallelization is used to multiply the statistic, but ...

- ... a thermalized configuration is needed.
- Thermalization stage is a bottleneck
- Critical slowing down near phase transition
- Need to speed-up a single Monte Carlo Markov chain $\Rightarrow$ parallelization
- Large minimal triangulation $\Rightarrow$ strong finite size effects $\Rightarrow$ large triangulations
- Simple parallelization does not work ...

Simple parallelization

A simple parallel algorithm, where many threads modify the same triangulation, has a very large overhead

- Detect collisions
- Thread synchronization: withdraw a move
- Move weight depends on global parameters - threads are coupled

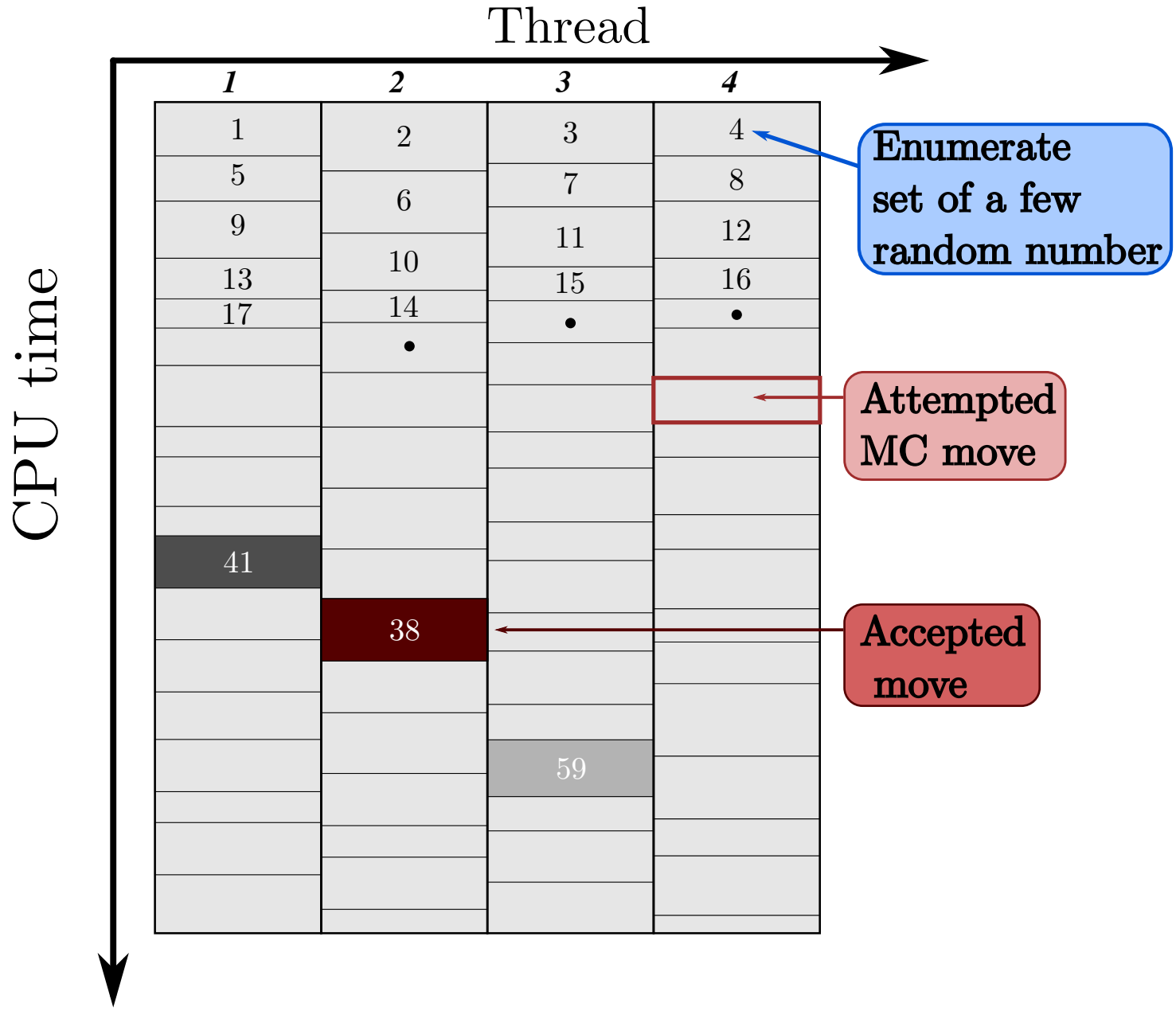


Parallel rejection

(by Jack Laiho, Phys. Rev. D **96** (2017) 064015)

- In some regions, the acceptance rate of MC moves is very low (critical slowing down)
- In such case a **parallel rejection** algorithm can be applied
- Multiple threads execute a series of **attempted moves**
- When one of them succeeds, rest of the threads are **stopped** and the triangulation is **updated**
- No collision detection is necessary
- To guarantee compatibility with the scalar version, *youngest* move has to be chosen (wait)
- Each attempted move is identified by three *random* numbers:
  - **which** move
  - **where** to make the move
  - **whether** to make the move (detailed balance)
- A block of random numbers is precomputed and split between threads. Synchronization with the RNG (separate thread).

Parallel rejection - diagram

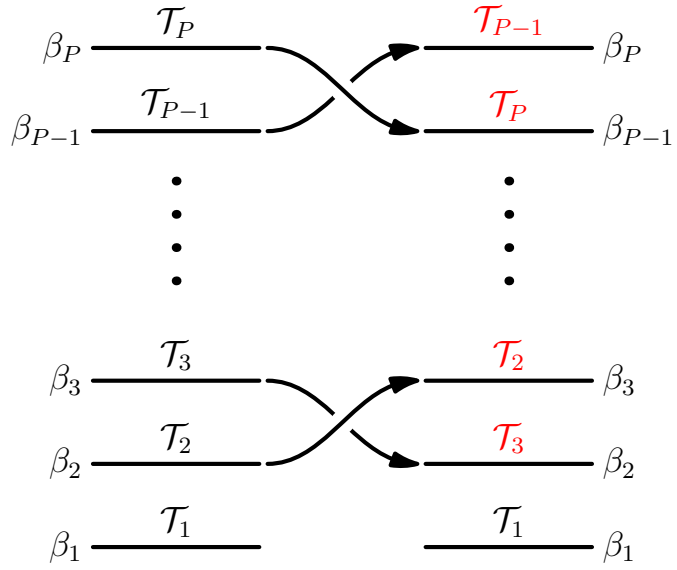


Parallel rejection - issues

- For three threads we gained 100% speed up for very small acceptance rate $\frac{1}{400}$
- Different move types significantly vary in acceptance rate
- Fixed number of random numbers (3) consumed per attempted move
- Replicability, compatibility with scalar version
- Thread synchronization is expensive
- Low level libraries, single barrier from `pthreads`
- Atomic operations (counter), busy waiting

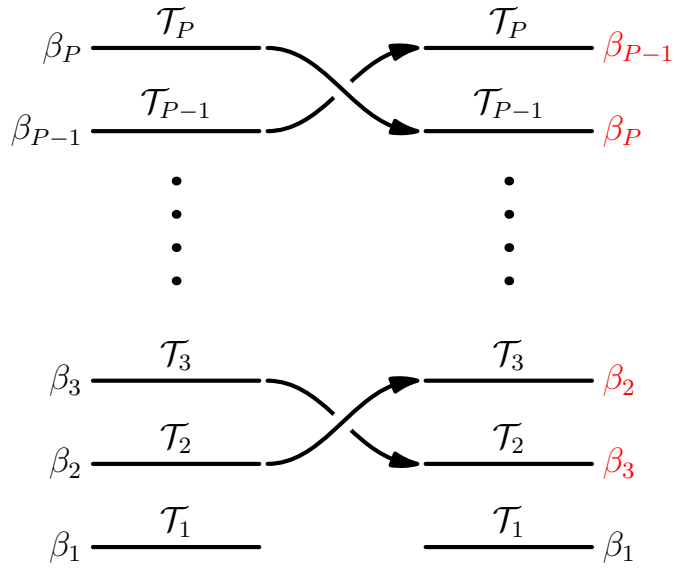
Replica exchange

- In this method we run a number of processes (replicas) with slightly different values of coupling constants.
- Periodically, the configurations are exchanged between processes with a certain probability (detailed balance).
- A *CDT configuration* is a heavy object (triangulation + additional data). Instead, the sets of a few coupling constants (denoted as β) are exchanged.



Replica exchange

- In this method we run a number of processes (replicas) with slightly different values of coupling constants.
- Periodically, the configurations are exchanged between processes with a certain probability (detailed balance).
- A *CDT configuration* is a heavy object (triangulation + additional data). Instead, the sets of a few coupling constants (denoted as β) are exchanged.



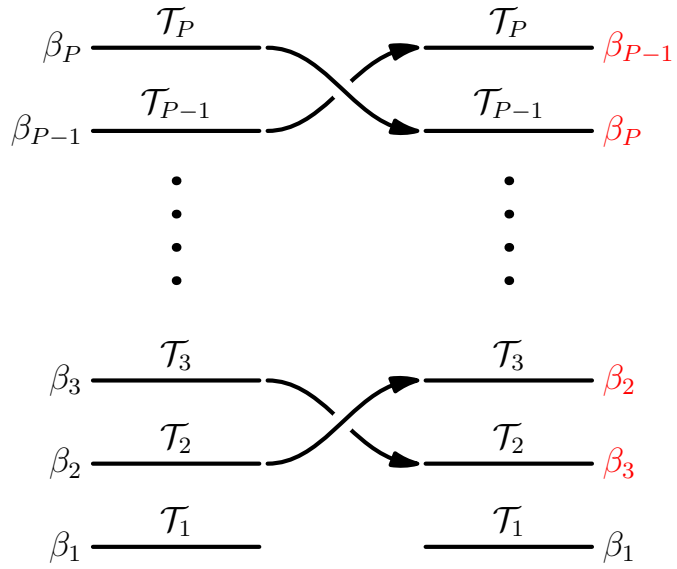
Replica exchange

- Exchange probability fulfills the *detailed balance condition*,

$$P(A) = P_1(\mathcal{T}_1)P_2(\mathcal{T}_2), \quad P(B) = P_2(\mathcal{T}_1)P_1(\mathcal{T}_2),$$

$$W(A \to B) = \min\{1, \frac{P_2(\mathcal{T}_1)P_1(\mathcal{T}_2)}{P_1(\mathcal{T}_1)P_2(\mathcal{T}_2)}\}$$

- **Implementation:** Managing process and *Linux message queues* to notify *working* processes when and with whom exchange



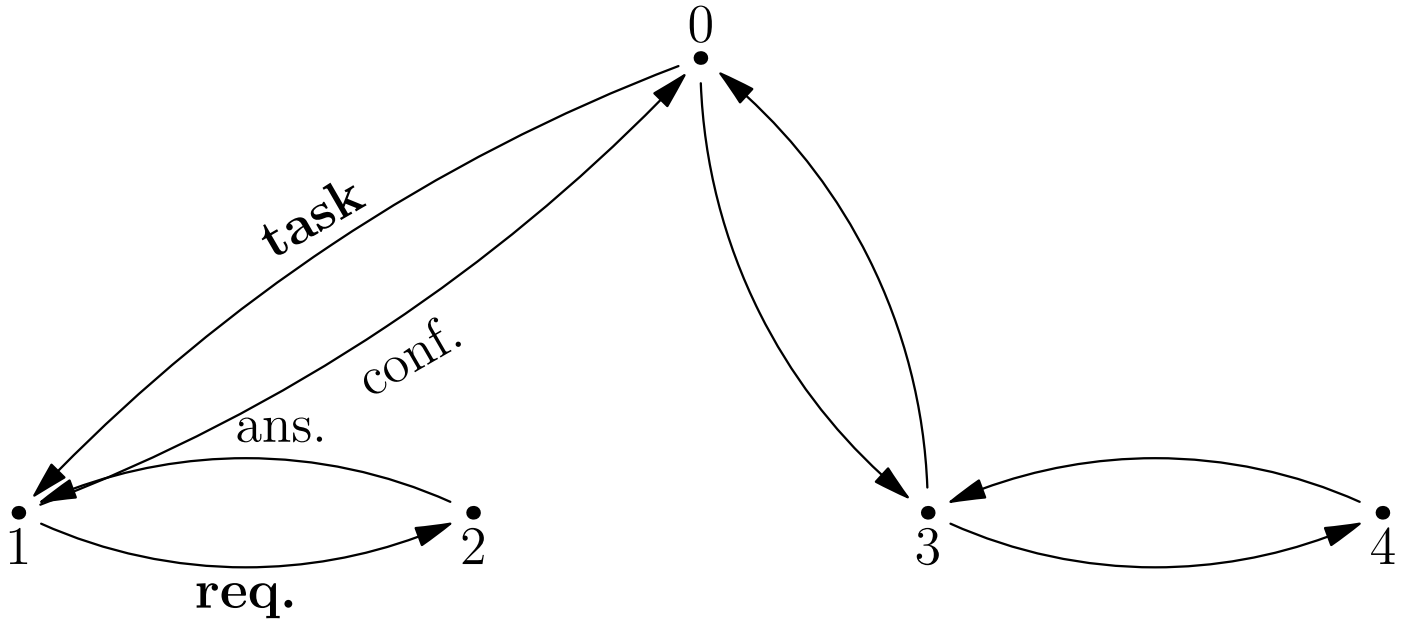
Replica exchange

- Exchange probability fulfills the *detailed balance condition*,

$$P(A) = P_1(\mathcal{T}_1)P_2(\mathcal{T}_2), \quad P(B) = P_2(\mathcal{T}_1)P_1(\mathcal{T}_2),$$

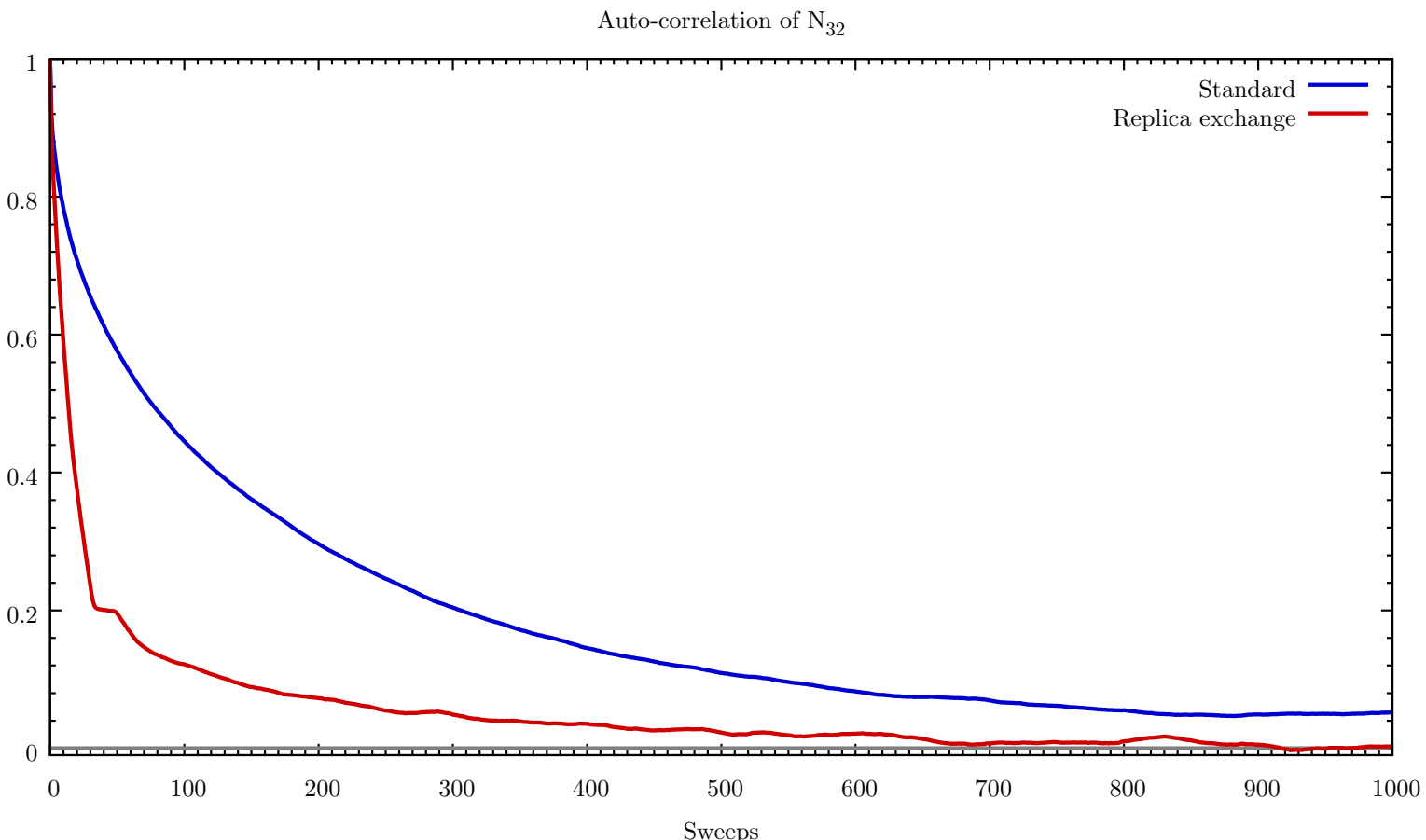
$$W(A \to B) = \min\{1, \frac{P_2(\mathcal{T}_1)P_1(\mathcal{T}_2)}{P_1(\mathcal{T}_1)P_2(\mathcal{T}_2)}\}$$

- **Implementation:** Managing process and *Linux message queues* to notify *working* processes when and with whom exchange



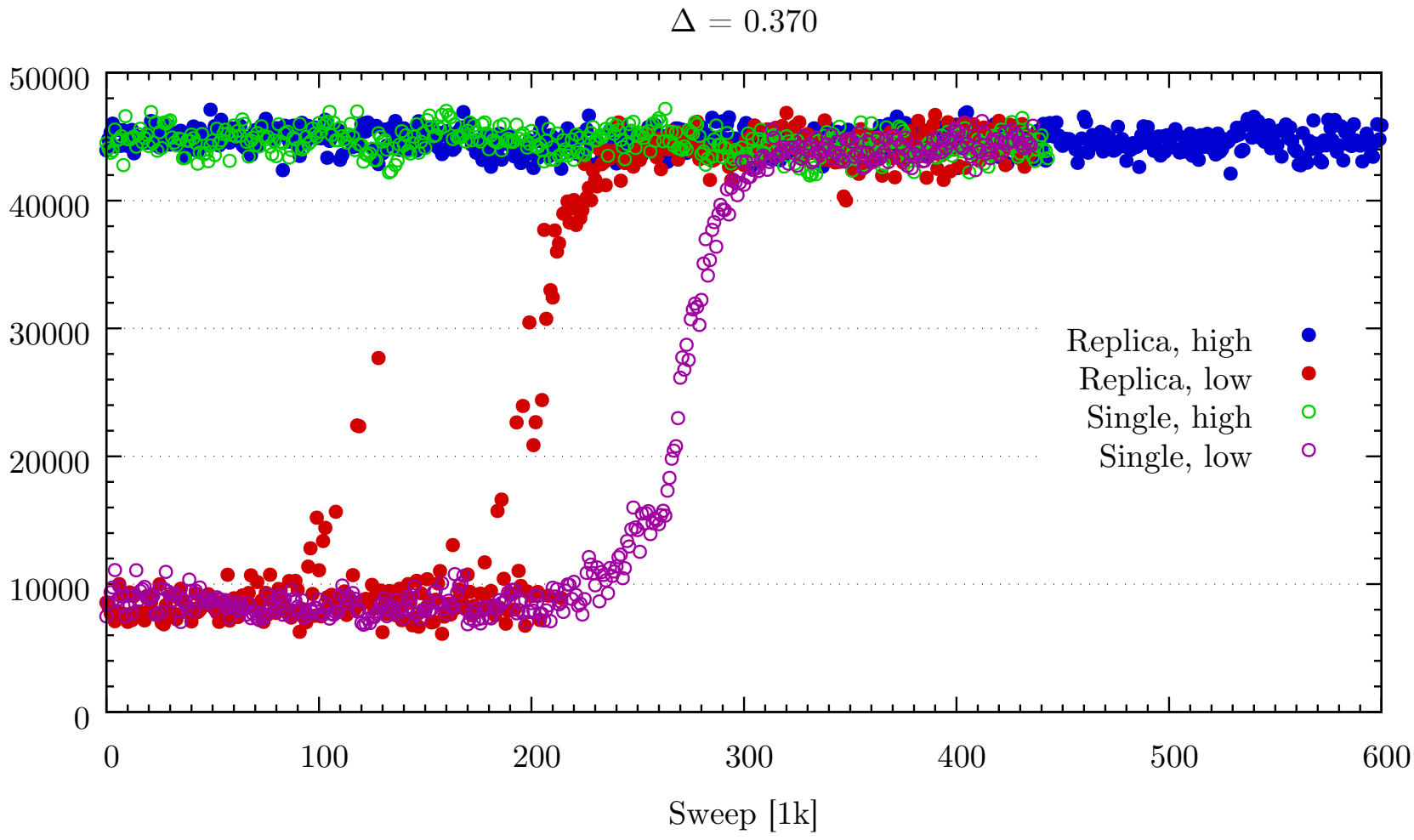
Autocorrelation time

- History of one configuration (Markov chain) contributes to many simulations and reduces the autocorrelation time.
- Exchange rate has to be higher than the autocorrelation time
- For 31 replicas a significant decrease of autocorrelation time is observed. A point near A/C transition (toroidal topology):



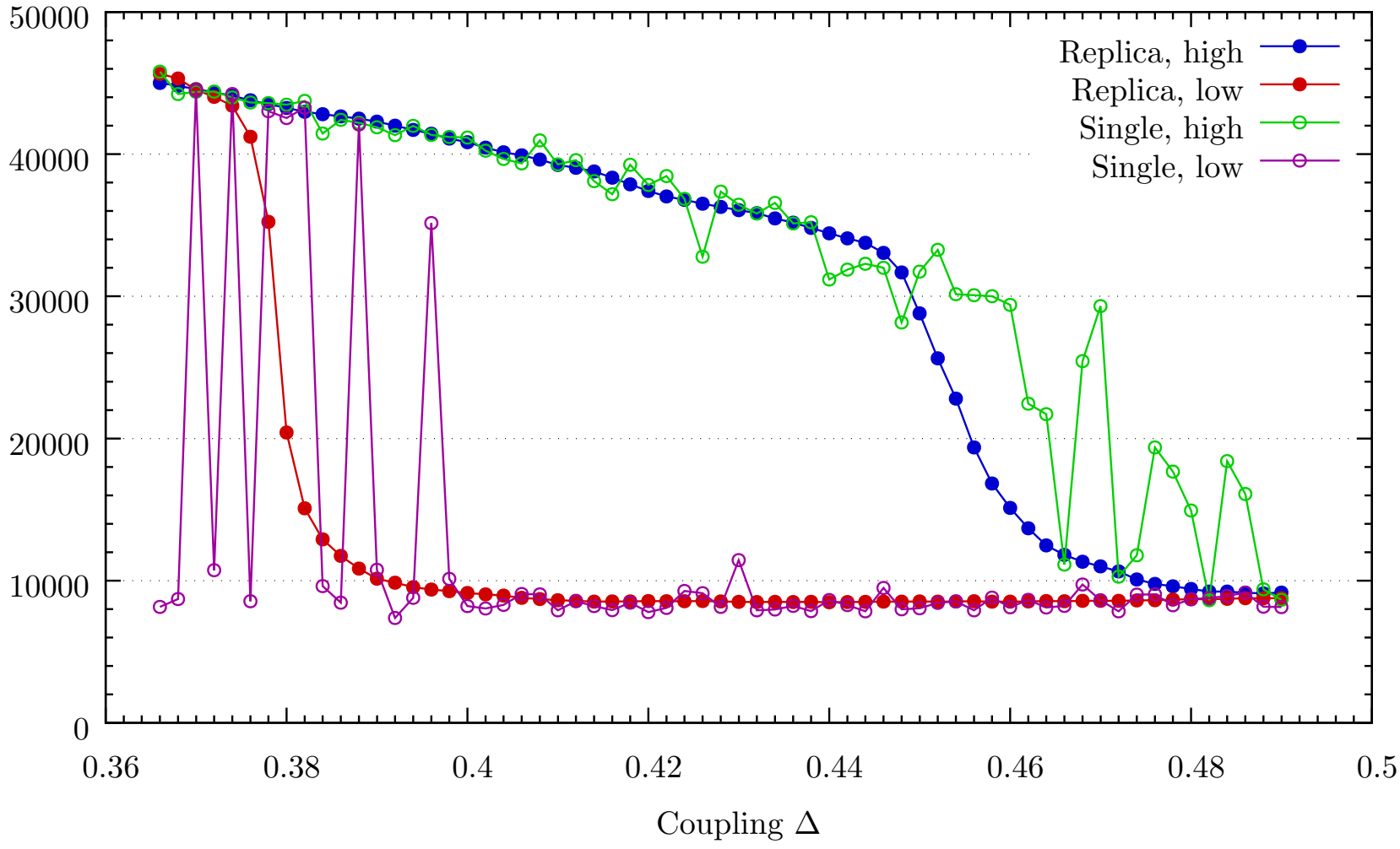
Replica exchange - hysteresis

Monte Carlo history of the relevant order parameter $\mathrm{Max}\, o(p)$ for single value of coupling constants. Visible jump and exchanges.



Replica exchange - hysteresis

Average value of the order parameter as a function of coupling constant. Comparison: replica exchange vs single threaded.



Thank You!