A flat histogram stochastic growth algorithm

(with application to polymers and proteins)

Thomas Prellberg

thomas.prellberg@tu-clausthal.de

Technical University of Clausthal

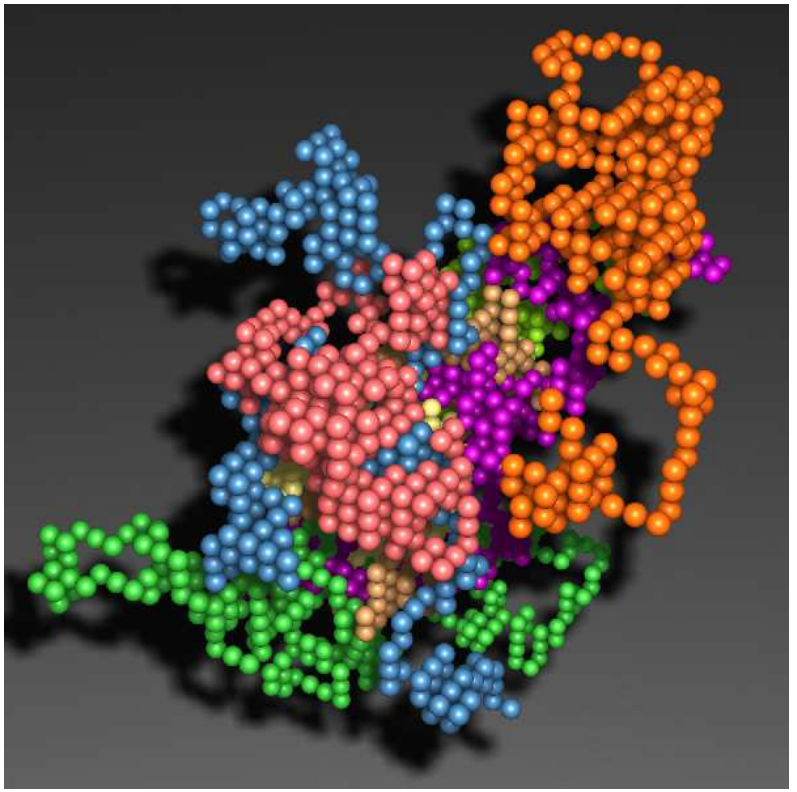
Joint work with

Jaroslaw Krawczyk, TU Clausthal, and A. Rechnitzer, Uni Melbourne

Introduction

Modelling of Polymers in Solution

- Polymers:
  long chains of monomers
- “Coarse-Graining”:
  beads on a chain
- “Excluded Volume”:
  minimal distance
- Contact with solvent:
  effective short-range interaction
- Good/bad solvent:
  repelling/attracting interaction
- Consequence:
  chains clump together



Eight polymers in a bad solvent
(Grassberger, FZ Jülich)

Lattice Model

Self-Avoiding Walks with Interactions

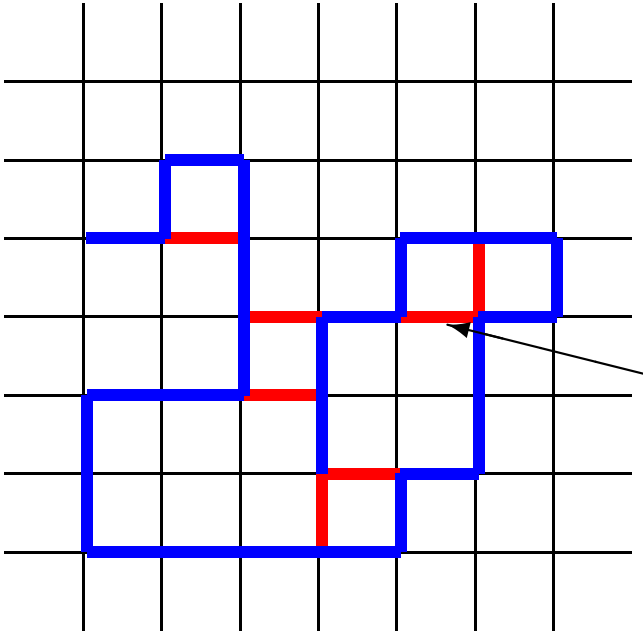
- Physical space $\rightarrow$ lattice $\mathbb{Z}^3$ (or $\mathbb{Z}^d$)
- Polymer in solution $\rightarrow$ random walk with self-avoidance
- Quality of solvent $\rightarrow$ short-range interaction ϵ
- Properties of monomers $i, j \rightarrow$ interaction $\epsilon = \epsilon_{i,j}$

Lattice Model

Self-Avoiding Walks with Interactions

- Physical space $\rightarrow$ lattice $\mathbb{Z}^3$ (or $\mathbb{Z}^d$)
- Polymer in solution $\rightarrow$ random walk with self-avoidance
- Quality of solvent $\rightarrow$ short-range interaction ϵ
- Properties of monomers $i, j \rightarrow$ interaction $\epsilon = \epsilon_{i,j}$
- Three examples:
  - SAW in a strip: interaction $\epsilon_{i,j} = 0$, restricted geometry
  - HP model: two types of monomers H and P: $\epsilon_{HH}, \epsilon_{HP}, \epsilon_{PP}$
    of interest: fixed finite sequence, density of states
  - ISAW model: interaction $\epsilon_{i,j} = -1$
    of interest: thermodynamic limit ($V = \infty$ and $n \rightarrow \infty$)

Lattice Model: (I)SAW



$n = 26$

$m = 7$

$e^{-\beta\epsilon_{i,j}}$

Boltzmann weight

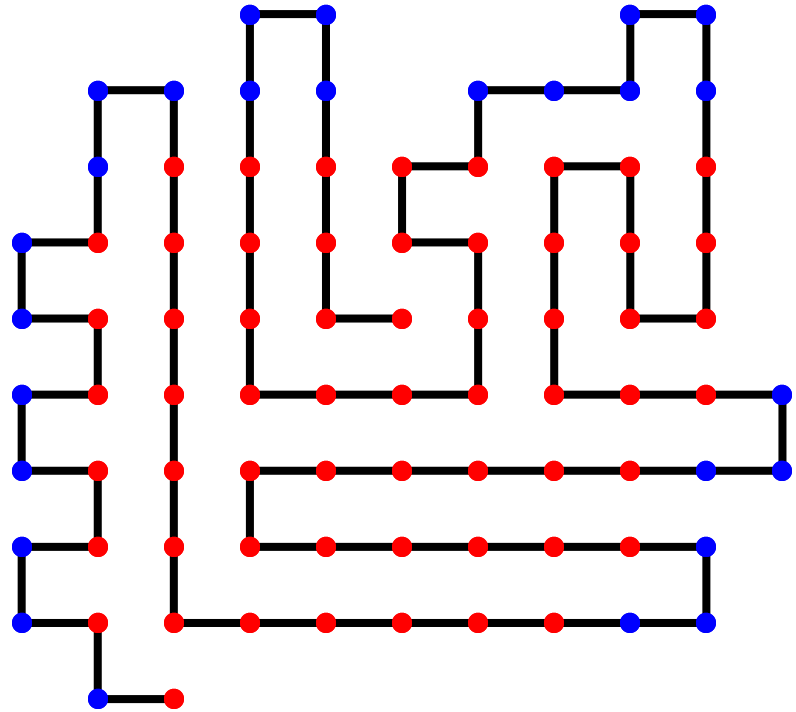
- ISAW model: interaction $\epsilon_{i,j} = -1$
- Partition function: $Z_n(\beta) = \sum_m C_{n,m} e^{\beta m}$
- $C_{n,m}$ number of SAW with length n and m interactions
- SAW: $C_n = \sum_m C_{n,m}$ number of SAW with length n ($\beta = 0$)

Lattice Model: HP model

- HP model: interaction $\epsilon_{HH} = -1, \epsilon_{HP} = \epsilon_{PH} = \epsilon_{PP} = 0$

Hydrophobic

Polar



- Groundstate of sequence with 85 monomers ($d = 2$)

Why Simulations?

- ISAW model:
  - Tricritical phase transition, $d_u = 3$
  - In principle understood, however surprising details e.g. “pseudo-first-order transition” for $d > 3$
  - No good understanding of collapsed regime

Why Simulations?

- ISAW model:
  - Tricritical phase transition, $d_u = 3$
  - In principle understood, however surprising details
    e.g. “pseudo-first-order transition” for $d > 3$
  - No good understanding of collapsed regime
- HP model:
  - Toy model for proteins
  - Design of sequences with specific ground state structure
  - Density of states – folding dynamics

Why Simulations?

- ISAW model:
  - Tricritical phase transition, $d_u = 3$
  - In principle understood, however surprising details
    e.g. “pseudo-first-order transition” for $d > 3$
  - No good understanding of collapsed regime
- HP model:
  - Toy model for proteins
  - Design of sequences with specific ground state structure
  - Density of states – folding dynamics
- Most interesting open questions in collapsed regime

Collapsed regime is notoriously difficult to simulate

Stochastic Growth Algorithms

PERM: “Go With The Winners”

PERM = Pruned and Enriched Rosenbluth Method

Grassberger, Phys Rev E 56 (1997) 3682

- Rosenbluth Method: kinetic growth

PERM: “Go With The Winners”

PERM = Pruned and Enriched Rosenbluth Method

Grassberger, Phys Rev E 56 (1997) 3682

- Rosenbluth Method: kinetic growth

- Enrichment: weight too large $\rightarrow$ make copies of configuration
- Pruning: weight too small $\rightarrow$ remove configuration occasionally

PERM: "Go With The Winners"

PERM = Pruned and Enriched Rosenbluth Method

Grassberger, Phys Rev E 56 (1997) 3682

- Rosenbluth Method: kinetic growth

- Enrichment: weight too large $\rightarrow$ make copies of configuration
- Pruning: weight too small $\rightarrow$ remove configuration occasionally

Observation: kinetic growth weights and interactions balance each other at suitable temperatures (in the collapse region)

PERM – further developments

- A significant improvement: nPERM = “new” PERM

  Hsu et al, J Chem Phys 118 (2003) 444

  - Enforce distinct enrichment steps
  - Crossover to exact enumeration

PERM – further developments

- A significant improvement: nPERM = “new” PERM

  Hsu et al, J Chem Phys 118 (2003) 444

  - Enforce distinct enrichment steps
  - Crossover to exact enumeration
- Current work: flatPERM = “flat histogram” PERM

  TP and JK, cond-mat/0312253, PRL (2004)

  - flatPERM samples a generalised multicanonical ensemble
  - Covers the whole temperature range in *one* simulation!

  Related: multicanonical chain growth algorithm

  Bachmann and Janke, PRL 91 208105 (2003)

PERM – further developments

- A significant improvement: nPERM = “new” PERM

  Hsu et al, J Chem Phys 118 (2003) 444

  - Enforce distinct enrichment steps
  - Crossover to exact enumeration

- Current work: flatPERM = “flat histogram” PERM

  TP and JK, cond-mat/0312253, PRL (2004)

  - flatPERM samples a generalised multicanonical ensemble
  - Covers the whole temperature range in *one* simulation!

  Related: multicanonical chain growth algorithm

  Bachmann and Janke, PRL 91 208105 (2003)

Applications:

- linear and branched polymers, proteins, percolation, . . .

Algorithm details - kinetic growth

View kinetic growth as | approximate enumeration |

Algorithm details - kinetic growth

View kinetic growth as approximate enumeration

- Exact enumeration: choose *all* a continuations with equal weight
- Kinetic growth: chose *one* continuation with a-fold weight (a may be zero).

Algorithm details - kinetic growth

View kinetic growth as approximate enumeration

- Exact enumeration: choose *all* a continuations with equal weight
- Kinetic growth: chose *one* continuation with a-fold weight (a may be zero).
  - An n step configuration gets assigned a weight

$$W = \prod_{k=0}^{n-1} a_k$$

  - S growth chains with weights $W_n^{(i)}$ give estimate

$$C_n^{est} = \langle W \rangle_n = \frac{1}{S} \sum_i W_n^{(i)}$$

Why does this work?

- The sum is normalized with the number S of *started* growth chains

$$C_n^{est} = \langle W \rangle_n = \frac{1}{S} \sum_i W_n^{(i)}$$

(not all of these growth chains reach length n)

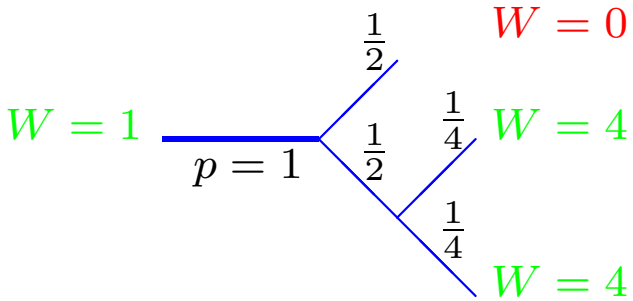
Why does this work?

- The sum is normalized with the number S of *started* growth chains

$$C_n^{est} = \langle W \rangle_n = \frac{1}{S} \sum_i W_n^{(i)}$$

(not all of these growth chains reach length n)

- Example:



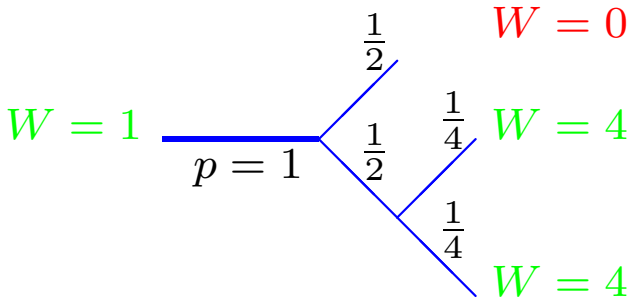
Why does this work?

- The sum is normalized with the number S of *started* growth chains

$$C_n^{est} = \langle W \rangle_n = \frac{1}{S} \sum_i W_n^{(i)}$$

(not all of these growth chains reach length n)

- Example:



$$\langle W \rangle_2 = \tfrac{1}{2}0 + \tfrac{1}{4}4 + \tfrac{1}{4}4 = 2$$

Algorithm details - pruning/enrichment

- $W_n^{(i)}$ is estimate of C_n
- Consider ratio $r = W_n^{(i)}/C_n^{est}$
  - $r > 1 \rightarrow$ enrichment step:
    make $c = \min(\lfloor r \rfloor, a_n)$ distinct copies with weight $\frac{1}{c} W_n^{(i)}$
  - $r < 1 \rightarrow$ pruning step:
    continue growing with probability r and weight C_n^{est}

Algorithm details - pruning/enrichment

- $W_n^{(i)}$ is estimate of C_n
- Consider ratio $r = W_n^{(i)}/C_n^{est}$
  - $r > 1 \rightarrow$ enrichment step:
    make $c = \min(\lfloor r \rfloor, a_n)$ distinct copies with weight $\frac{1}{c} W_n^{(i)}$
  - $r < 1 \rightarrow$ pruning step:
    continue growing with probability r and weight C_n^{est}
- Consequences
  - Number of samples generated for each n is roughly constant
  - Ideally, unbiased random walk in configuration size
  - We have a flat histogram algorithm

From PERM to flatPERM

- PERM: estimate number of configurations C_n
  - $C_n^{est} = \langle W \rangle_n$
  - $r = W_n^{(i)} / C_n^{est}$

From PERM to flatPERM

- PERM: estimate number of configurations C_n
  - $C_n^{est} = \langle W \rangle_n$
  - $r = W_n^{(i)} / C_n^{est}$

Consider energy $E_m = \epsilon m$, temperature $\beta = 1/k_B T$

- thermal PERM: estimate partition function $Z_n(\beta)$
  - $Z_n^{est}(\beta) = \langle W \exp(-\beta E) \rangle_n$
  - $r = W_n^{(i)} \exp(-\beta E_m^{(i)}) / Z_n^{est}(\beta)$

From PERM to flatPERM

- PERM: estimate number of configurations C_n
  - $C_n^{est} = \langle W \rangle_n$
  - $r = W_n^{(i)} / C_n^{est}$

Consider energy $E_m = \epsilon m$, temperature $\beta = 1/k_B T$

- thermal PERM: estimate partition function $Z_n(\beta)$
  - $Z_n^{est}(\beta) = \langle W \exp(-\beta E) \rangle_n$
  - $r = W_n^{(i)} \exp(-\beta E_m^{(i)}) / Z_n^{est}(\beta)$

- flatPERM: estimate density of states $C_{n,m}$
  - $C_{n,m}^{est} = \langle W \rangle_{n,m}$
  - $r = W_{n,m}^{(i)} / C_{n,m}^{est}$

From PERM to flatPERM

- PERM: estimate number of configurations C_n
  - $C_n^{est} = \langle W \rangle_n$
  - $r = W_n^{(i)} / C_n^{est}$

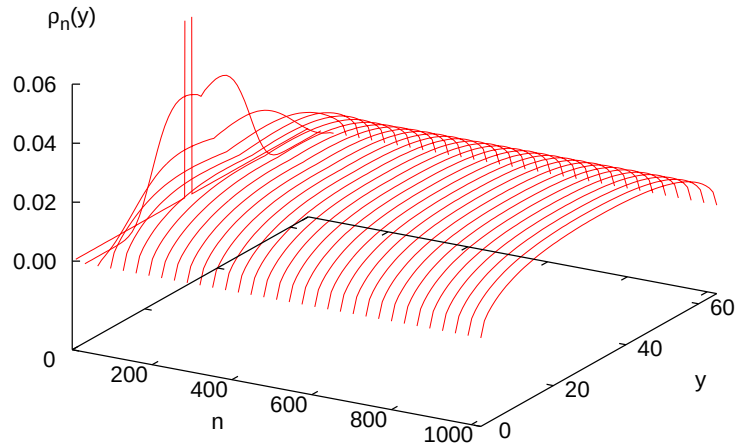
Consider energy $E_m = \epsilon m$, temperature $\beta = 1/k_B T$

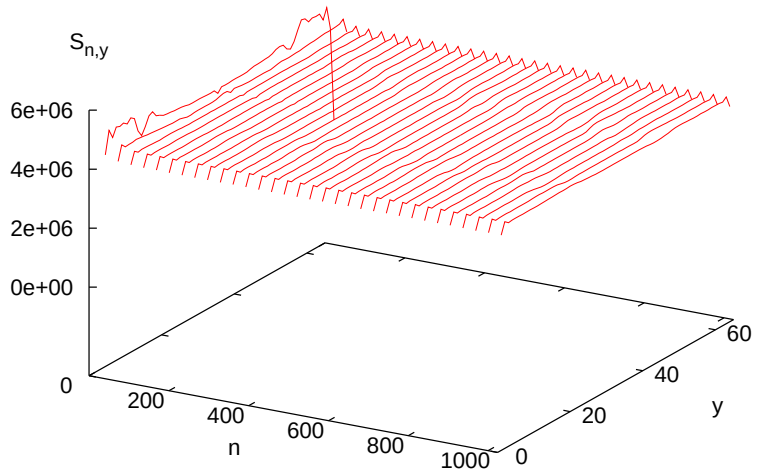
- thermal PERM: estimate partition function $Z_n(\beta)$
  - $Z_n^{est}(\beta) = \langle W \exp(-\beta E) \rangle_n$
  - $r = W_n^{(i)} \exp(-\beta E_m^{(i)}) / Z_n^{est}(\beta)$
- flatPERM: estimate density of states $C_{n,m}$
  - $C_{n,m}^{est} = \langle W \rangle_{n,m}$
  - $r = W_{n,m}^{(i)} / C_{n,m}^{est}$
- Generalization to more microcanonical parameters possible

Simulation Results

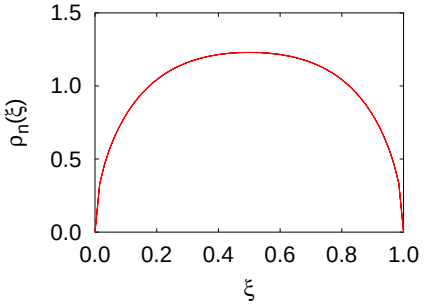
Simulational results: SAW in a strip

- 2d SAW in a strip: strip width 64, up to $n = 1024$





- Endpoint density

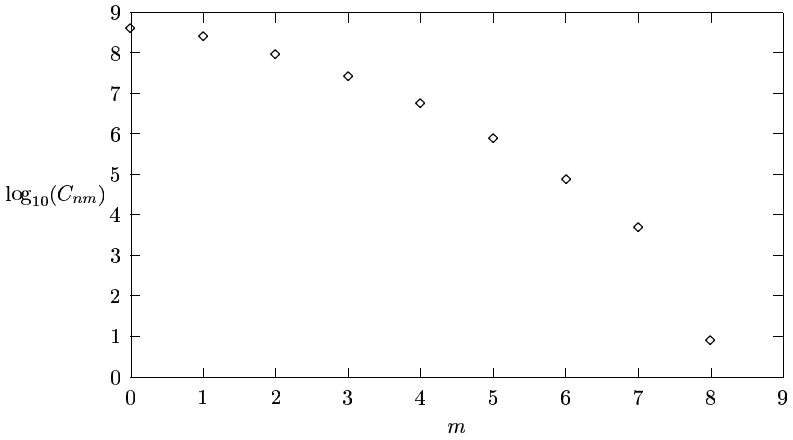


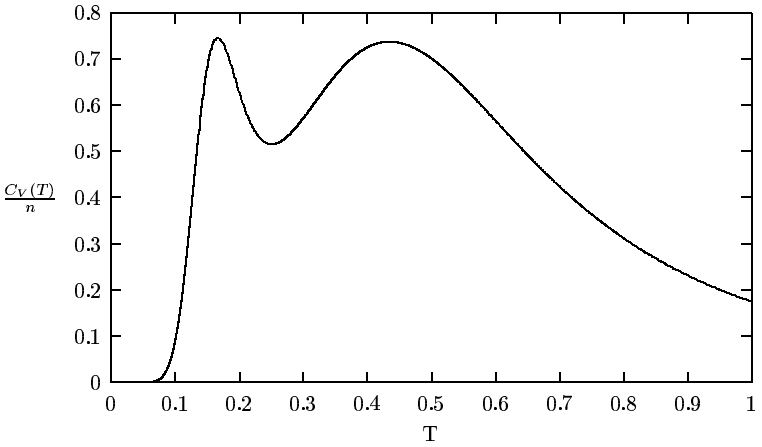
- Compares favorably with “Markovian anticipation”–PERM

Hsu and Grassberger, Eur Phys J 36 (2003) 209

Simulation results: HP model

- Sequence I (14 Monomers, HPHPHHPHPHHPPH, $d = 3$):

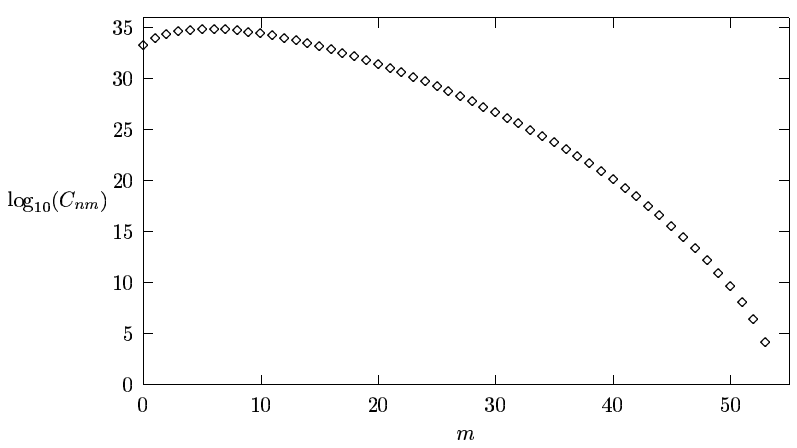


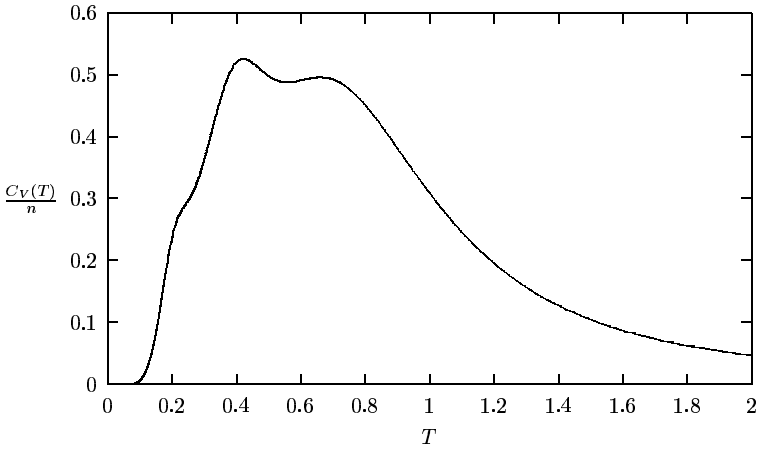


- Pedagogical example, engineered for native ground state
- Perfect agreement with exact enumeration

Simulation results: HP model

- Sequence II (85 Monomers, $d = 2$):





- Investigated several other sequences in $d = 2$ and $d = 3$
- Collapsed regime accessible
- Reproduced known ground state energies
- Obtained $C_{n,m}$ over large range

Simulation: dynamics of the algorithm

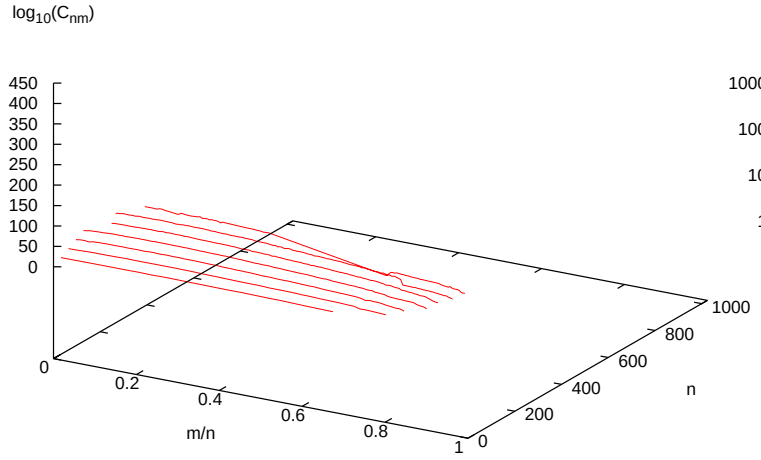
2d ISAW simulation up to $n = 1024$

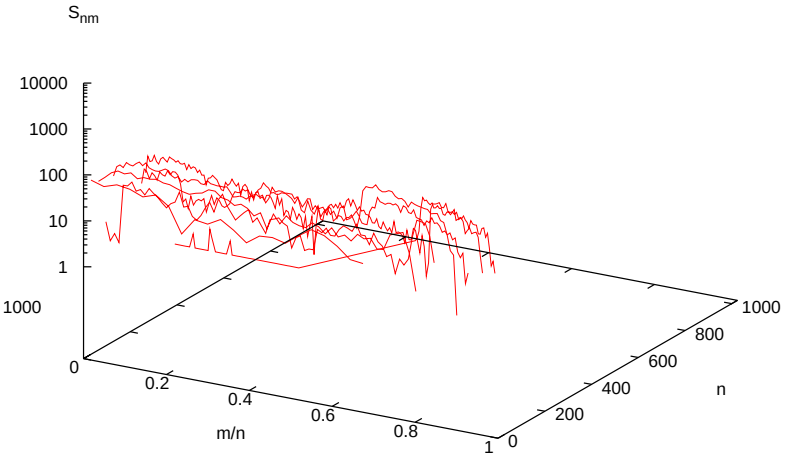
- To stabilize algorithm (avoid initial overflow/underflow): Delay growth of large configurations
- Here: after t tours growth up to length $10t$

Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $1,000,000$

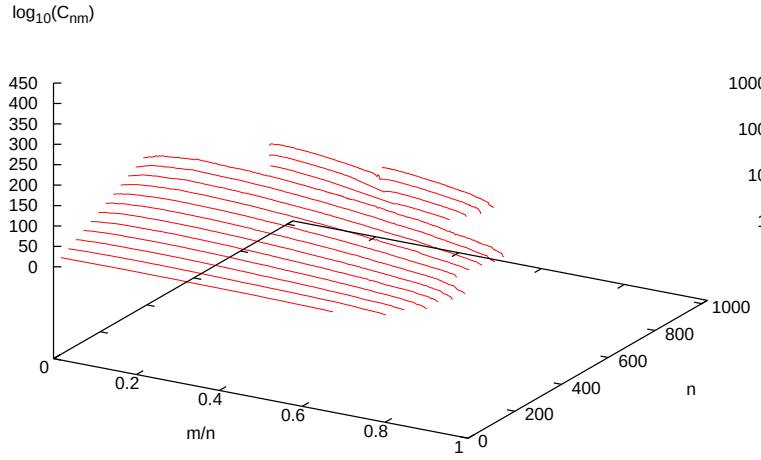


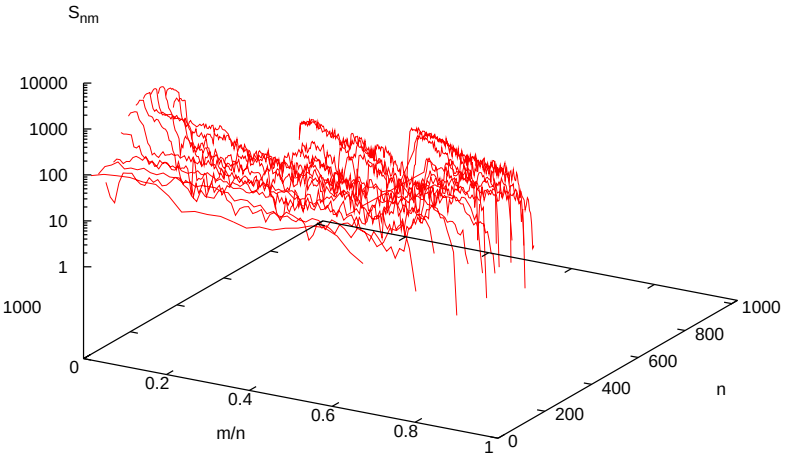


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $10,000,000$

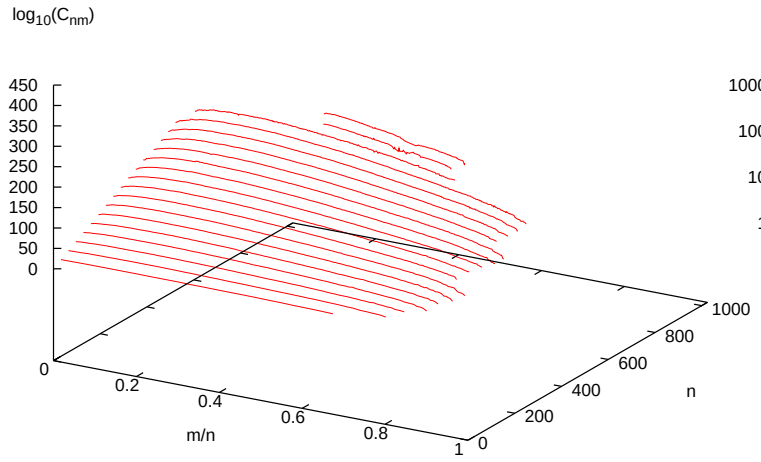


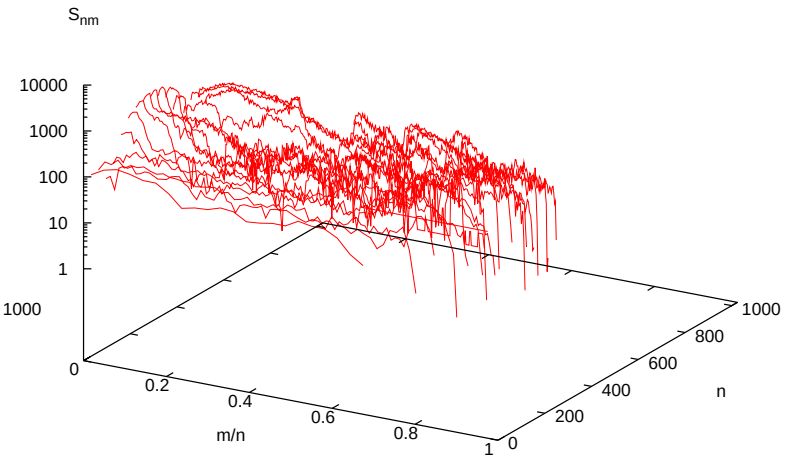


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $20,000,000$

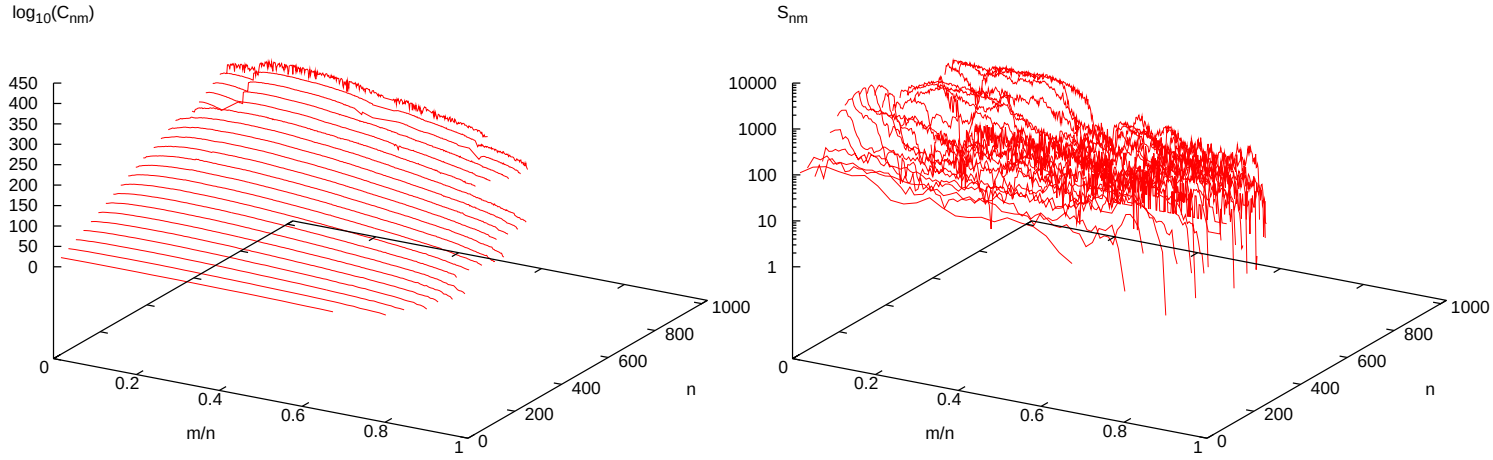




Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

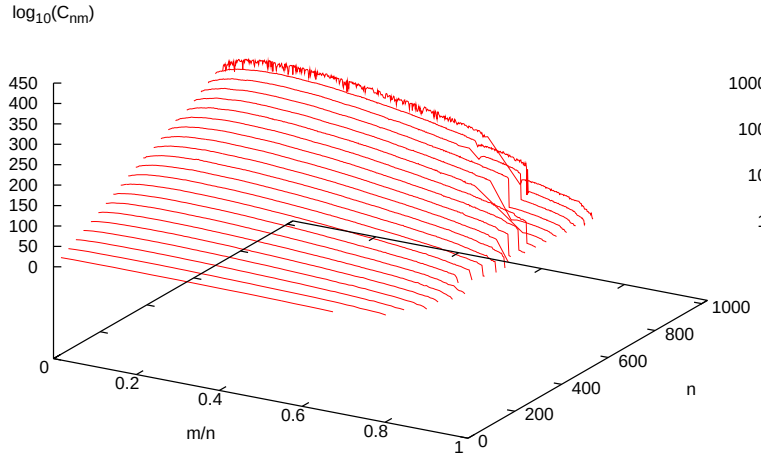
- Total sample size: $30,000,000$

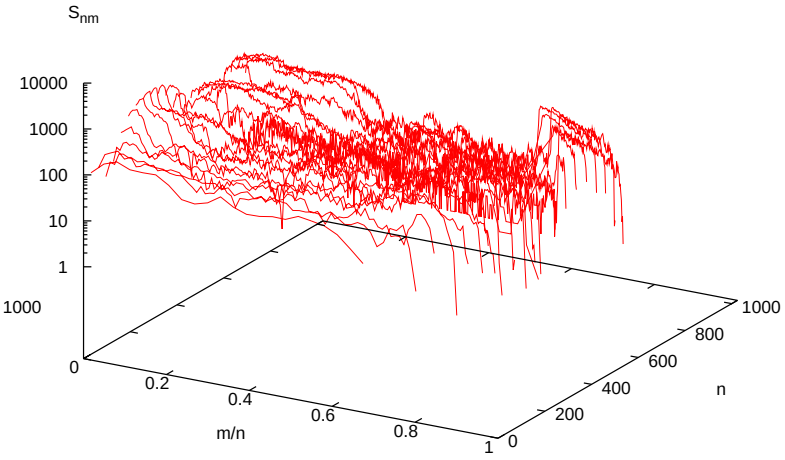


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $40,000,000$

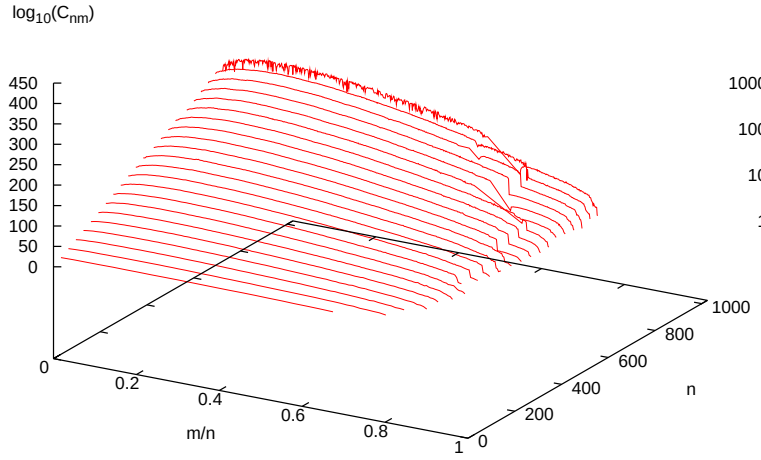


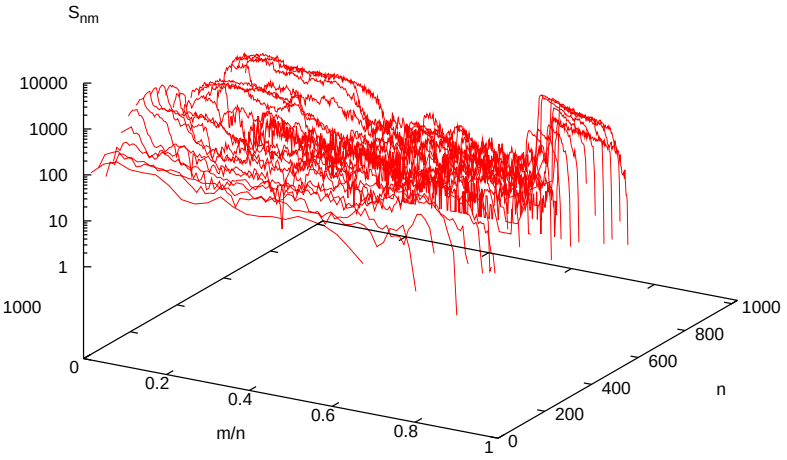


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $50,000,000$

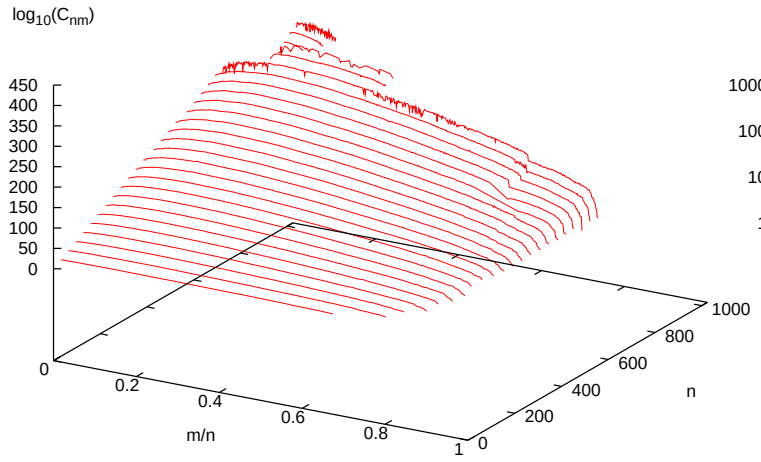


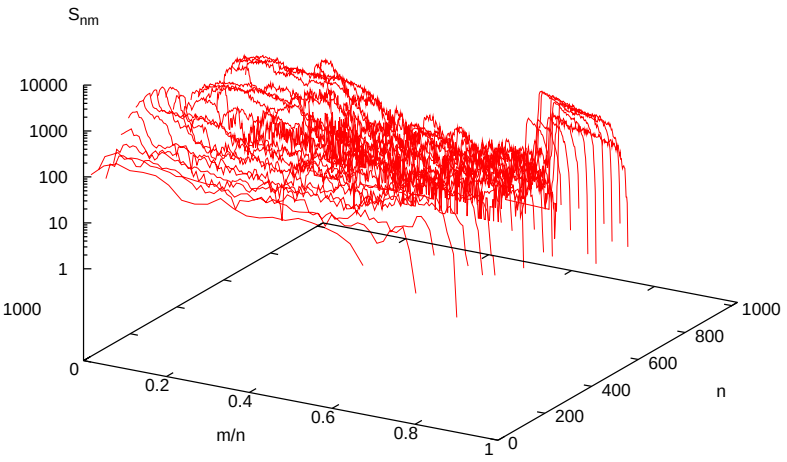


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $60,000,000$

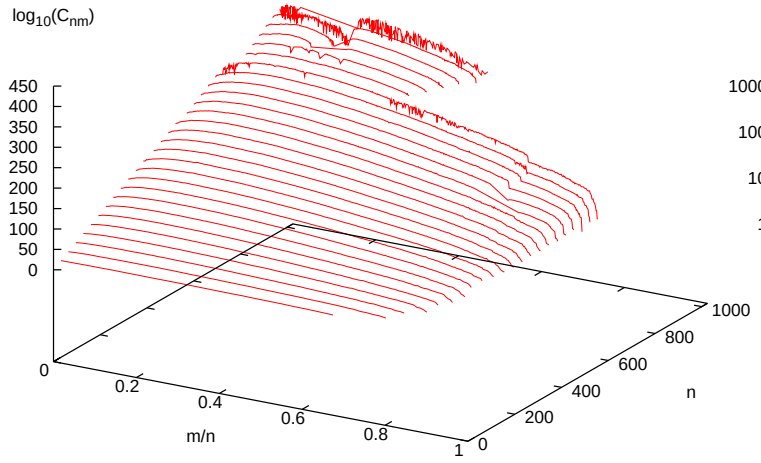


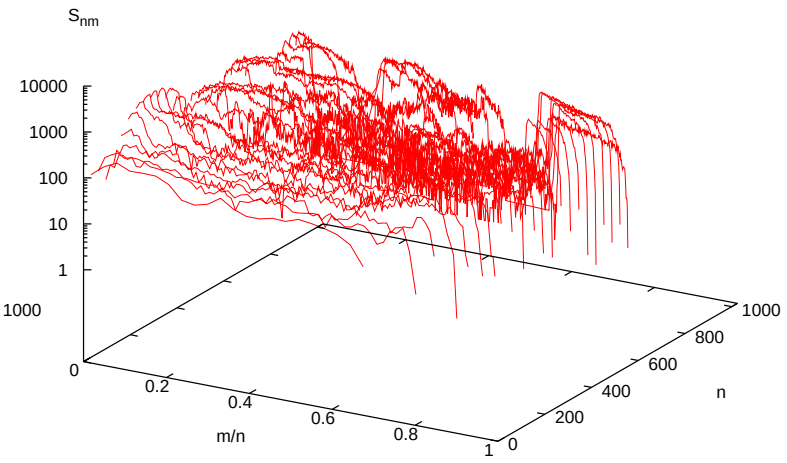


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $70,000,000$

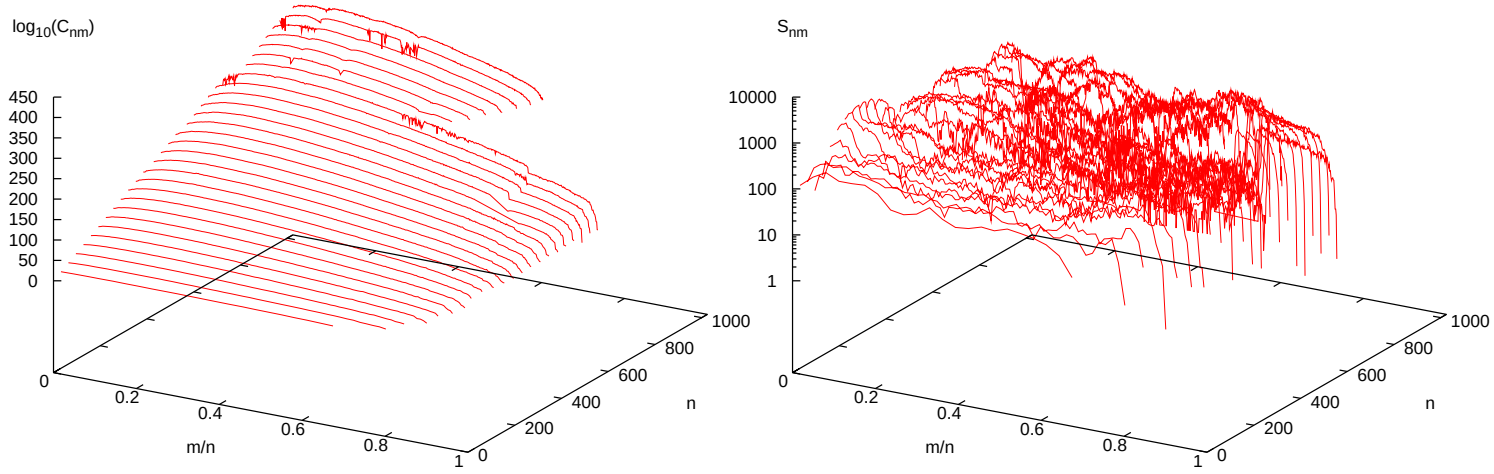




Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

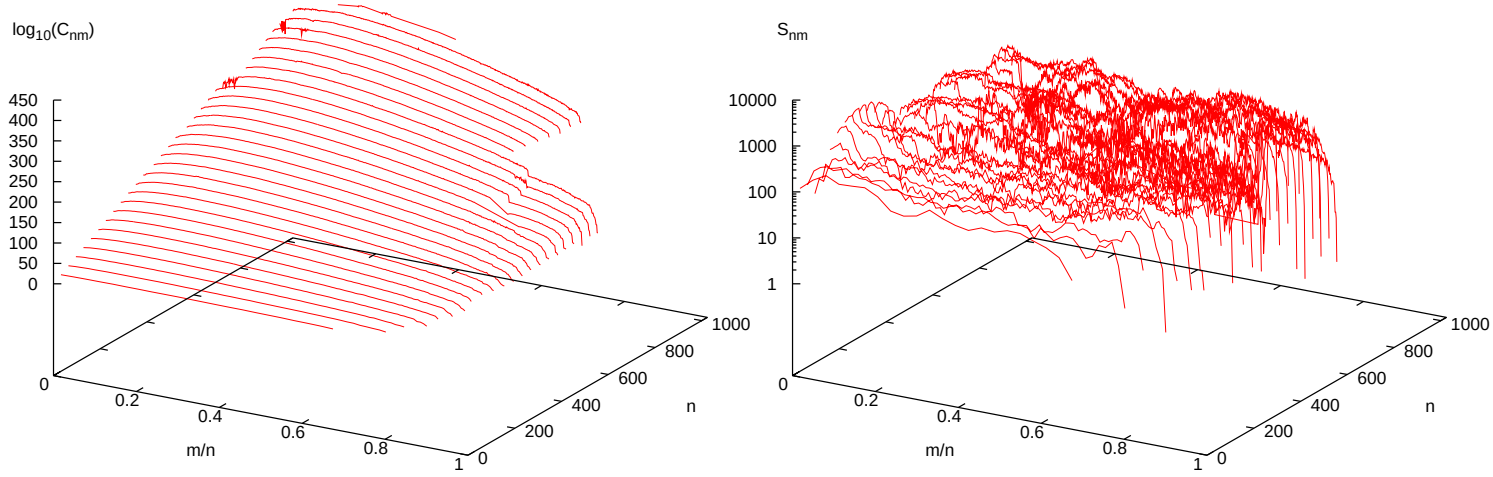
- Total sample size: $80,000,000$



Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

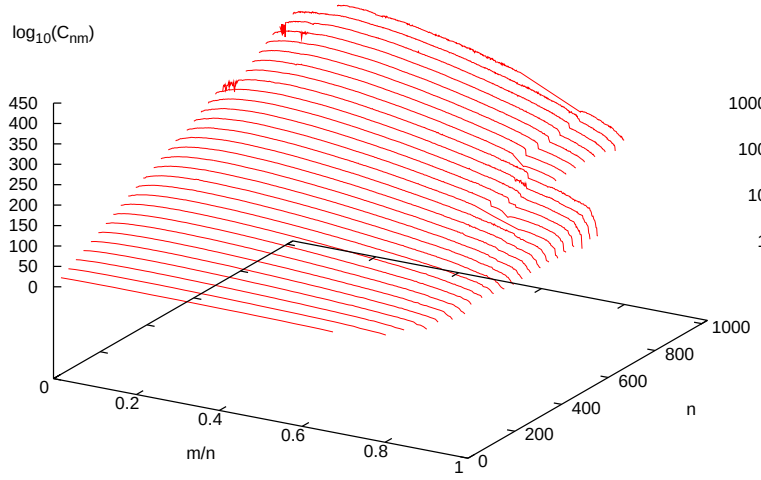
- Total sample size: $90,000,000$

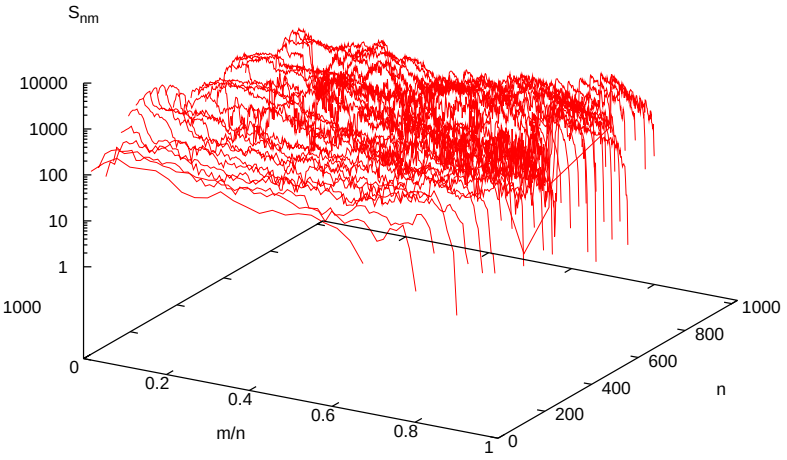


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $100,000,000$

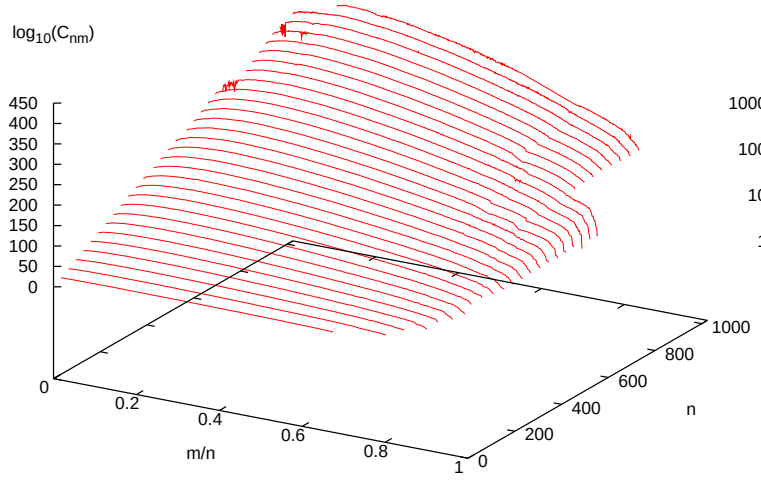


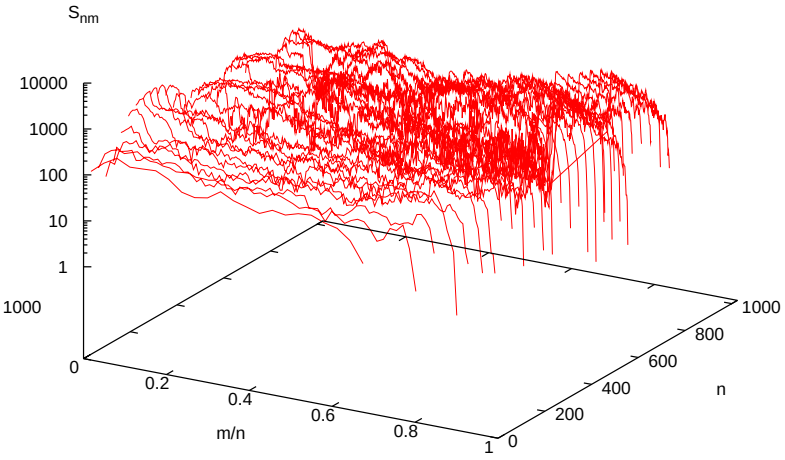


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $110,000,000$

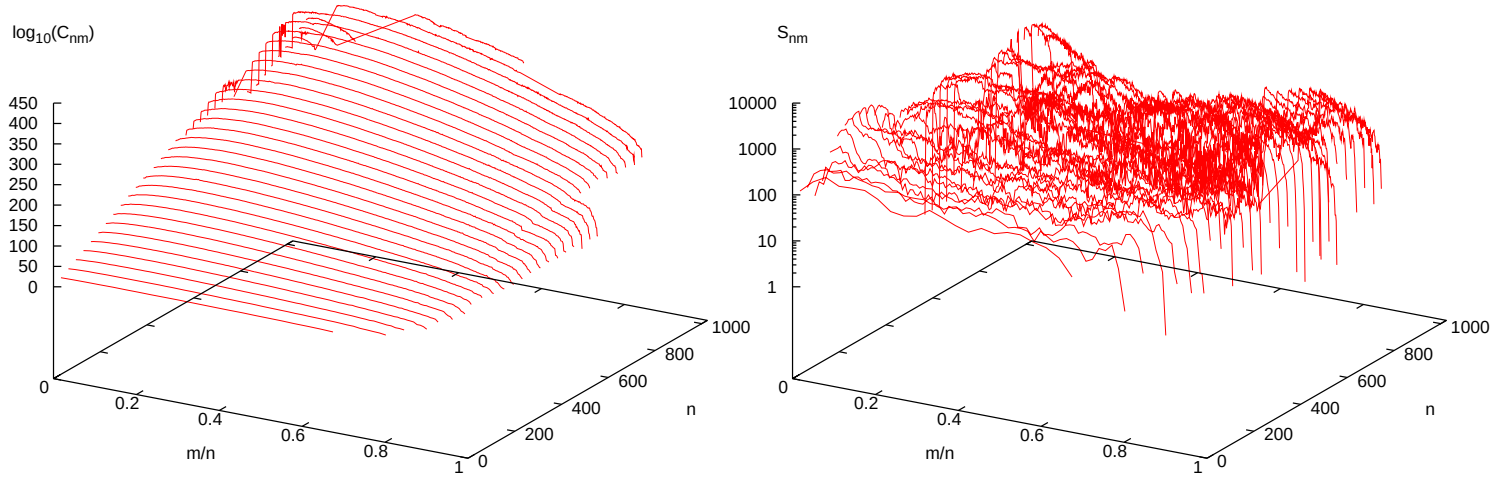




Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

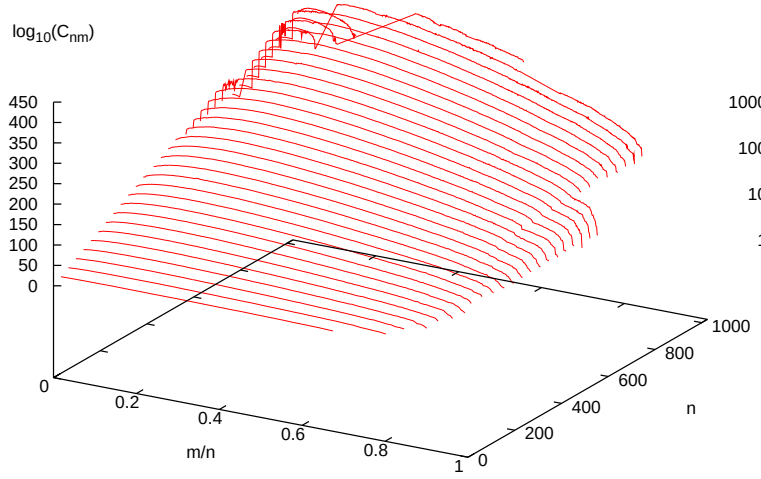
- Total sample size: $120,000,000$

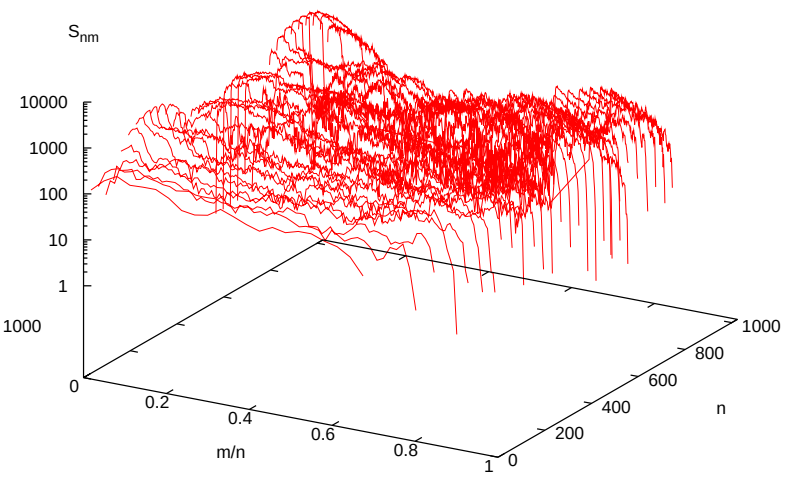


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $130,000,000$

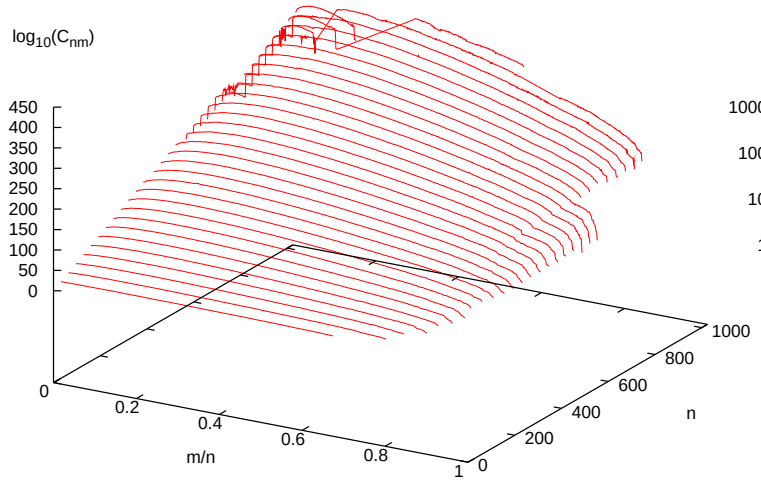


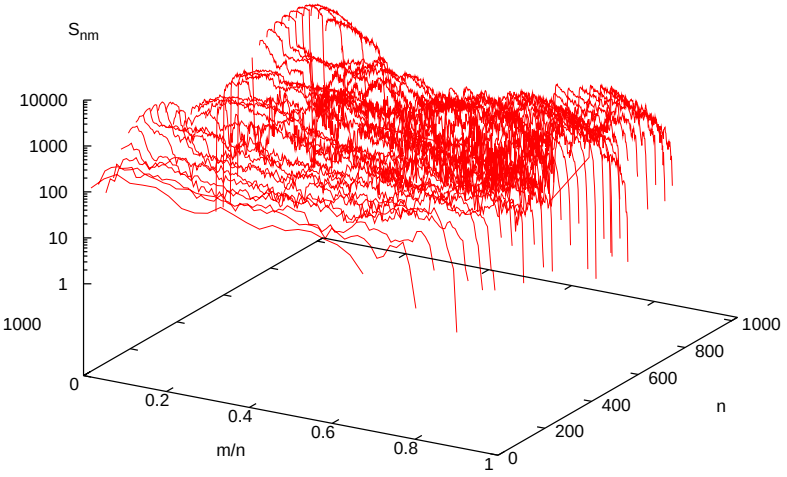


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $140,000,000$

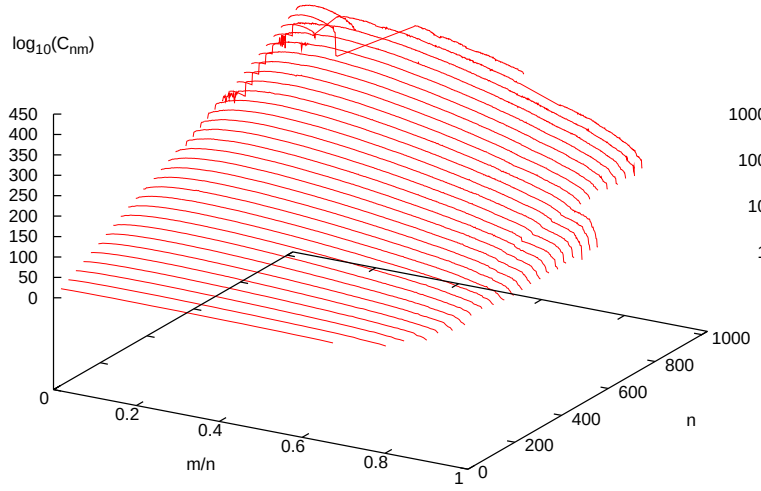


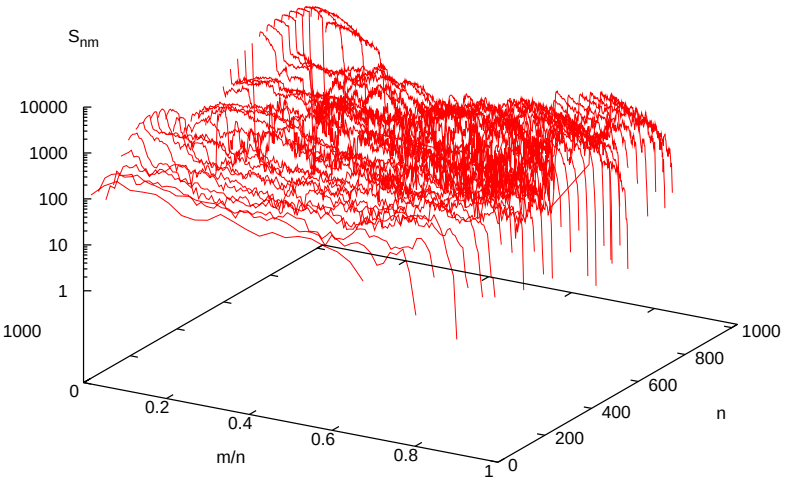


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $150,000,000$

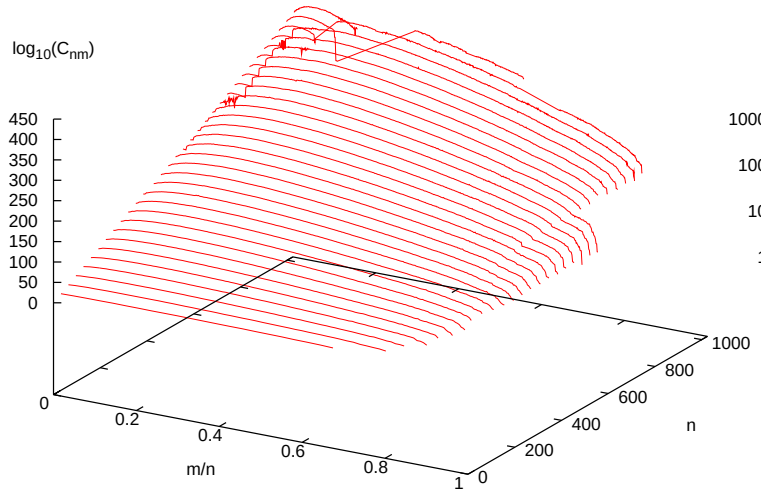


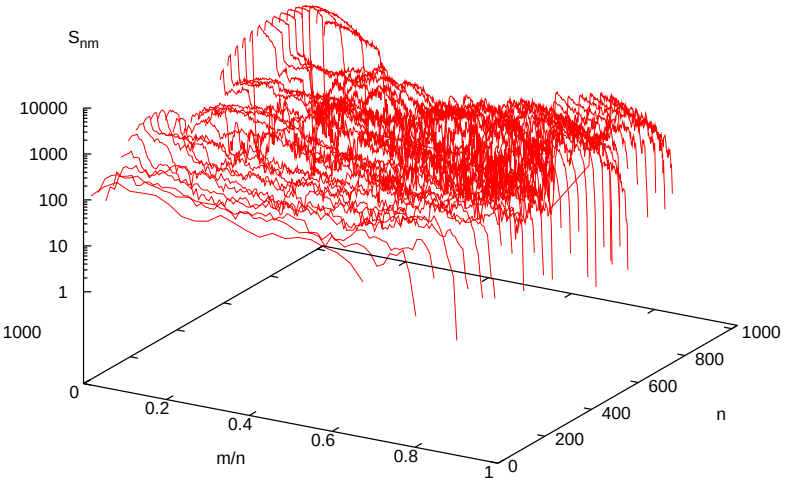


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $160,000,000$

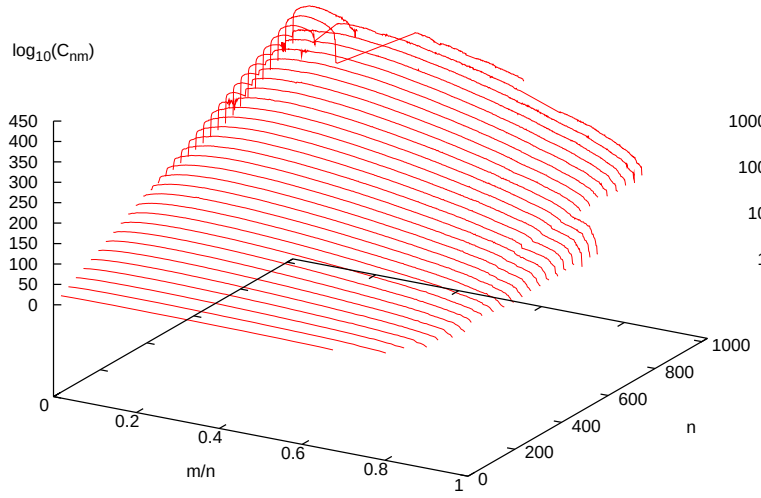


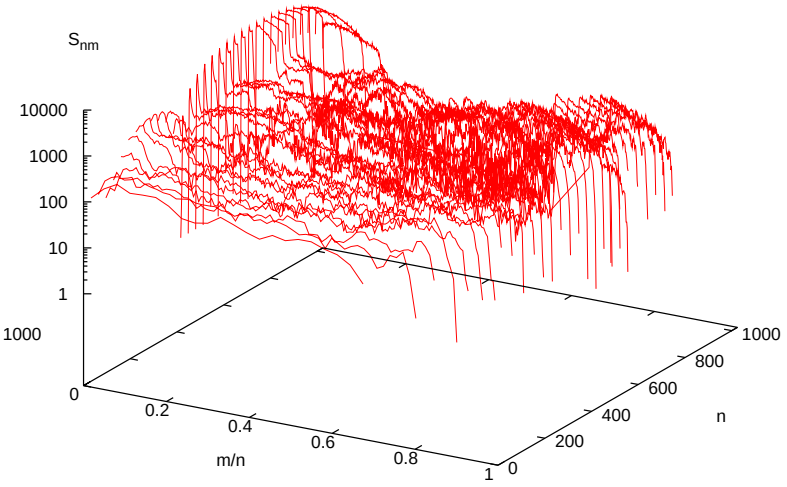


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $170,000,000$

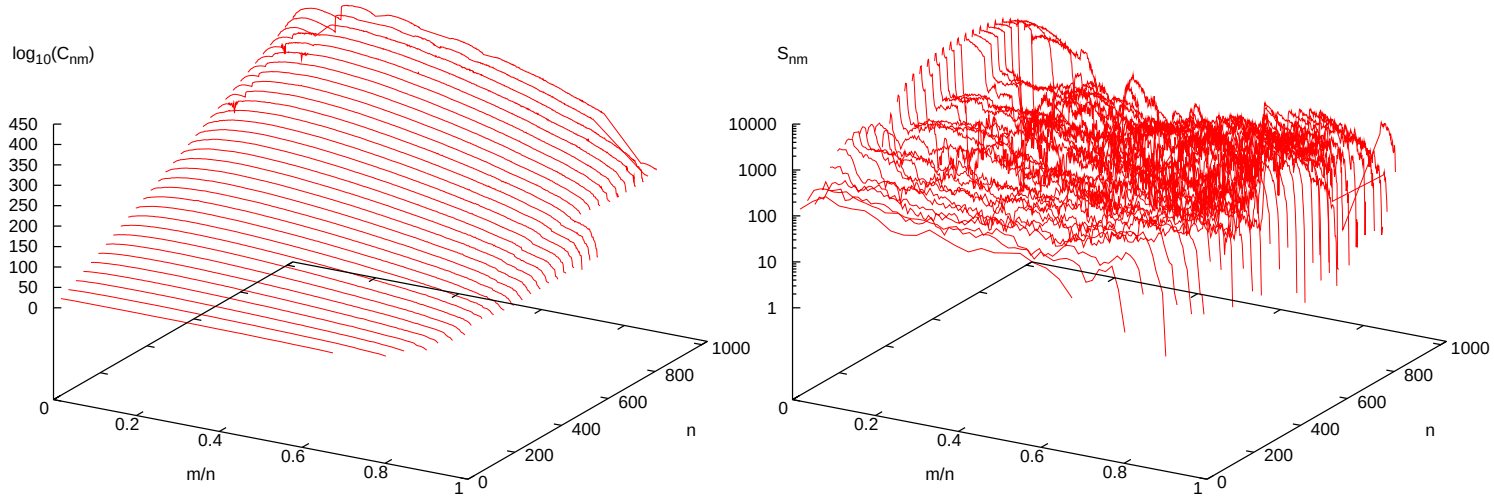




Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

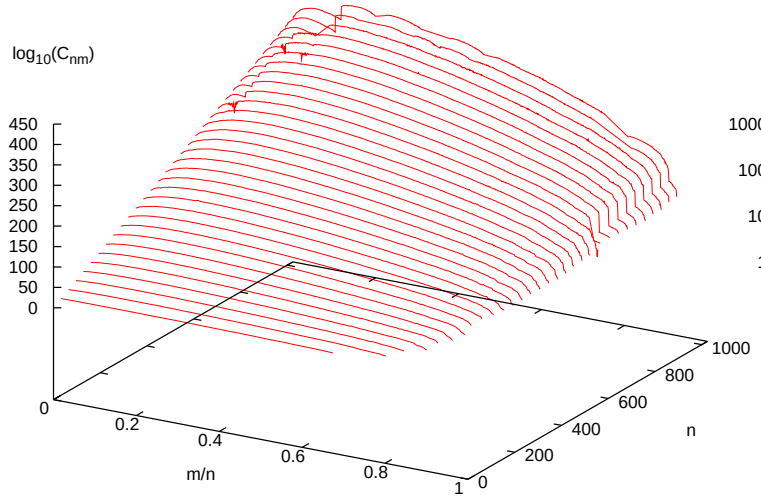
- Total sample size: $180,000,000$

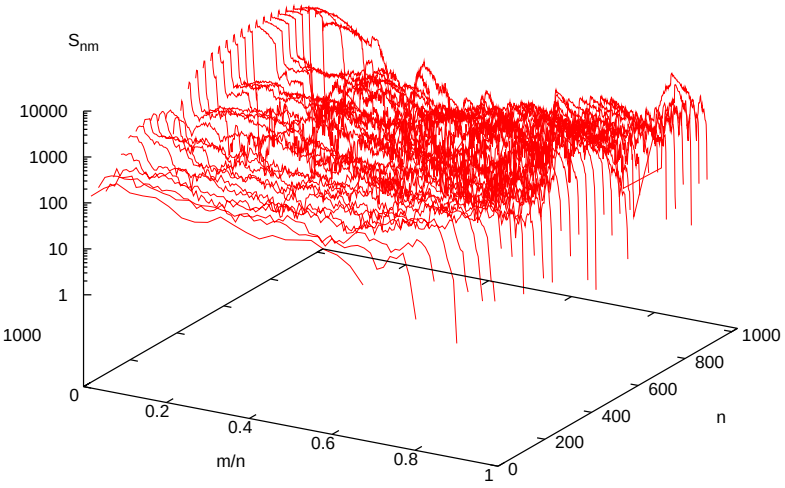


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $190,000,000$

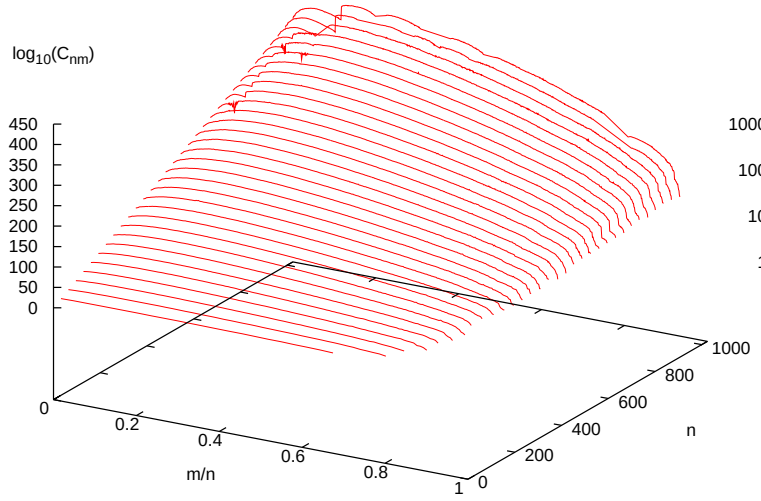


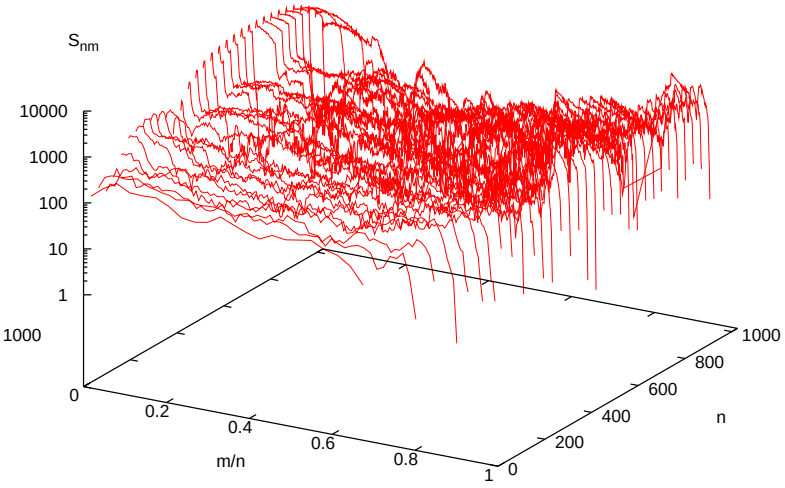


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $200,000,000$

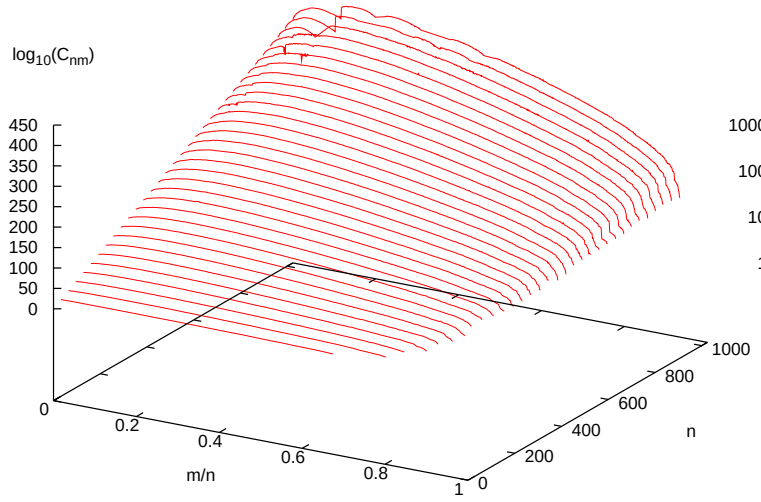


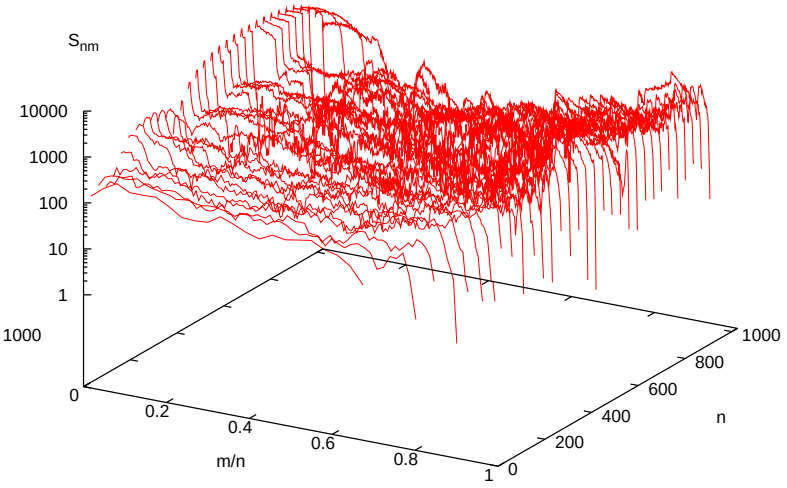


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $210,000,000$

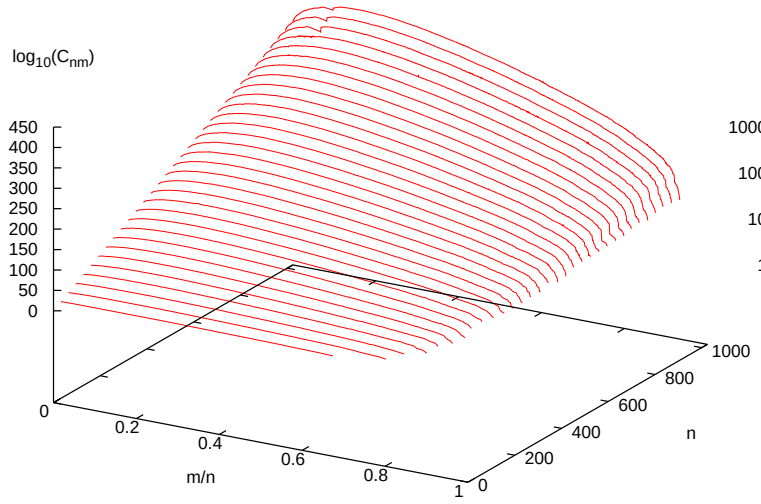


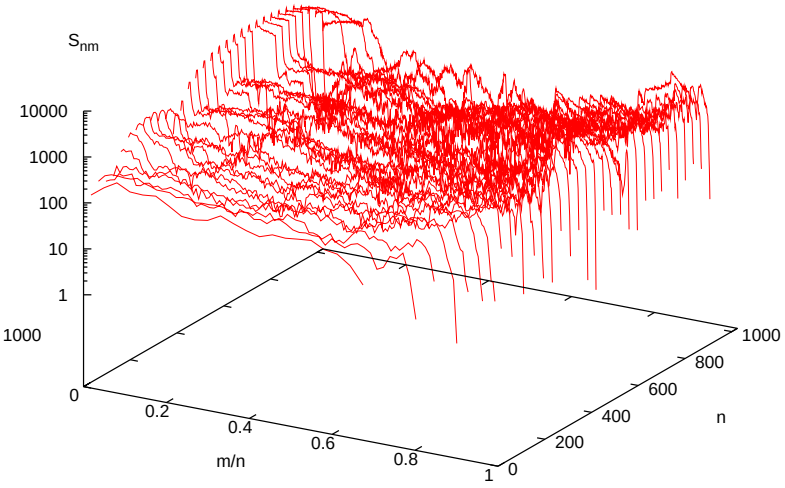


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $220,000,000$

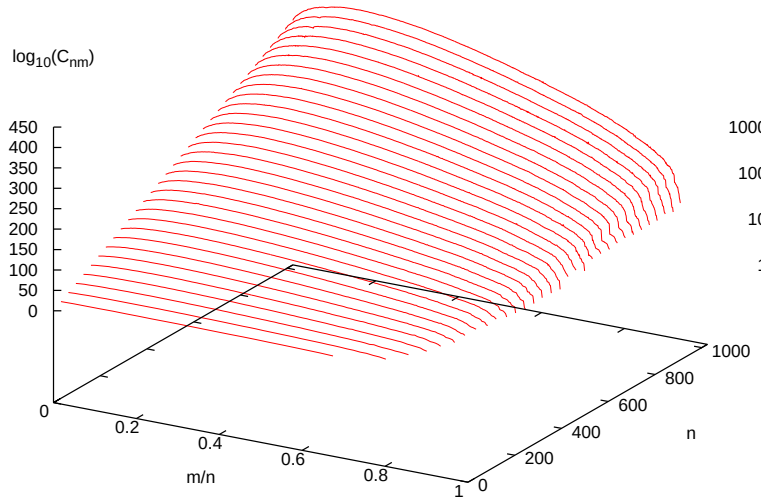


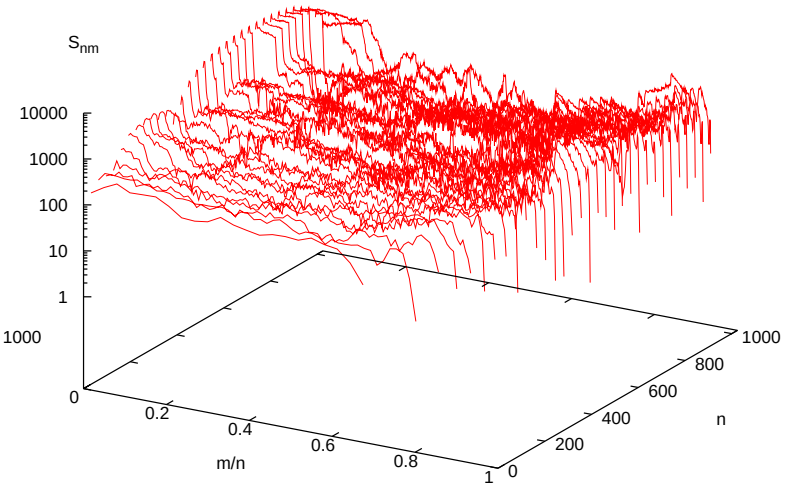


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $230,000,000$

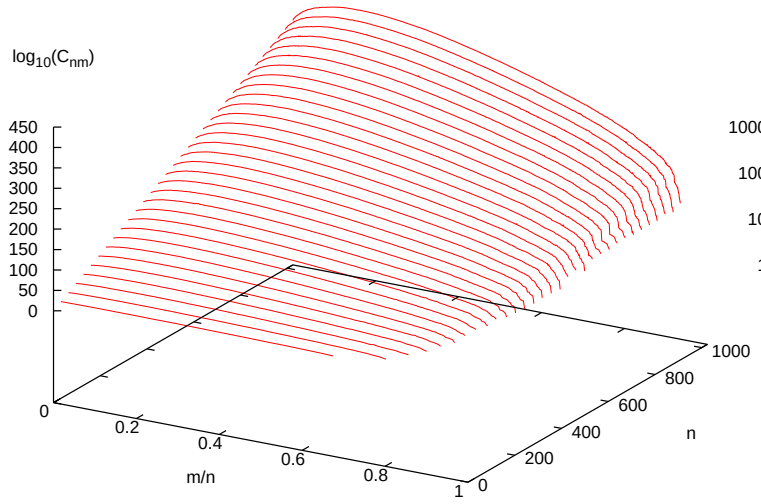


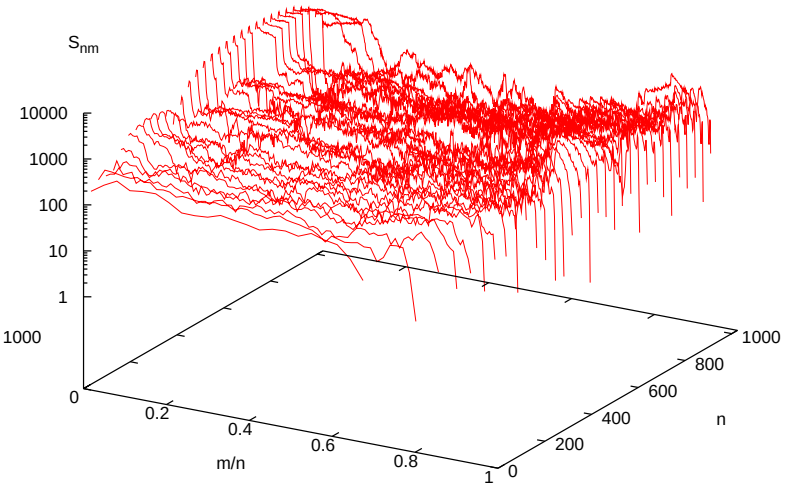


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $240,000,000$

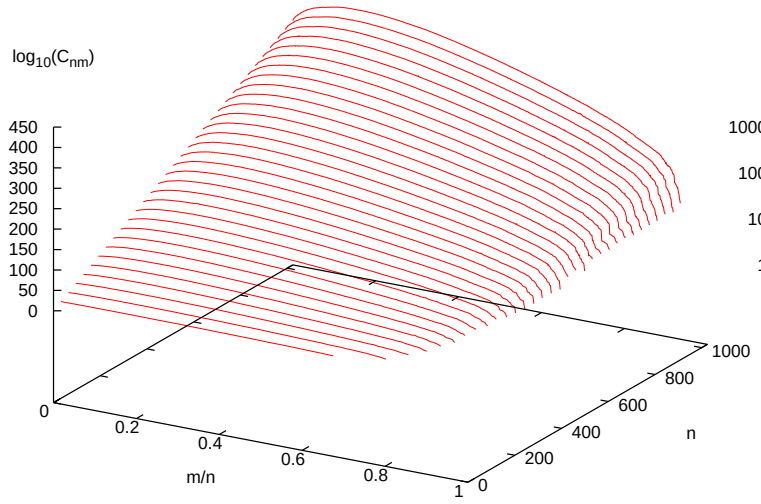


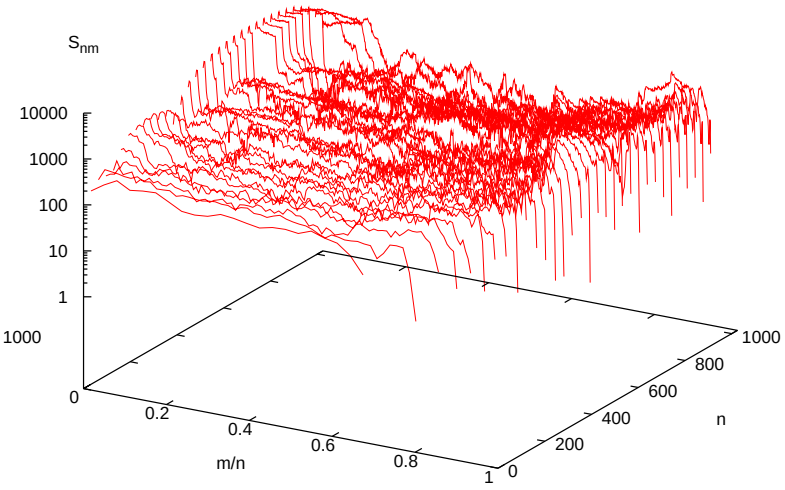


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $250,000,000$

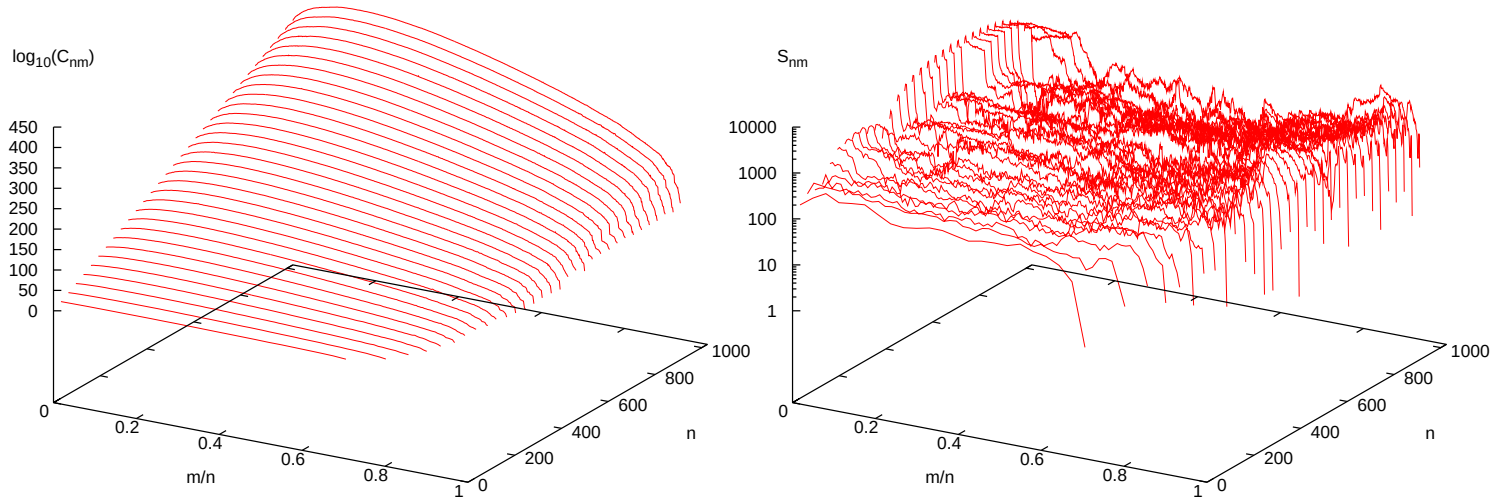




Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

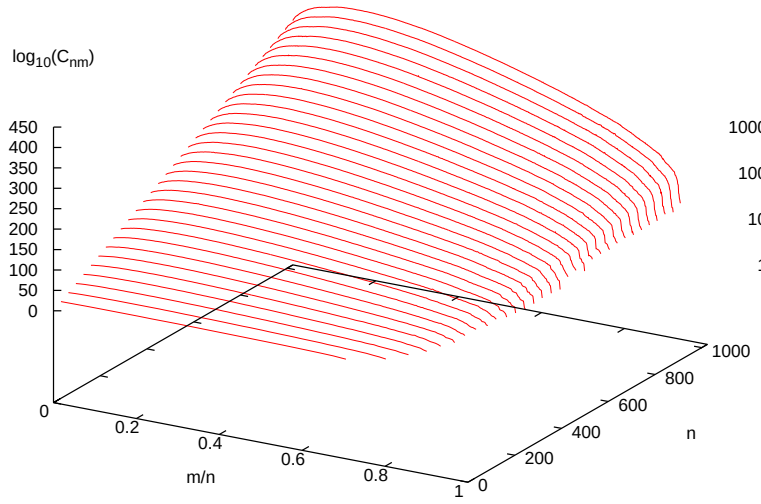
- Total sample size: $260,000,000$

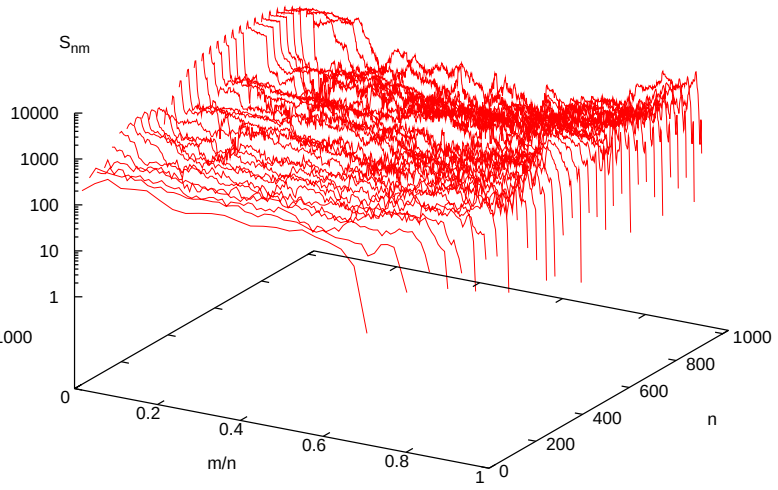


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $270,000,000$

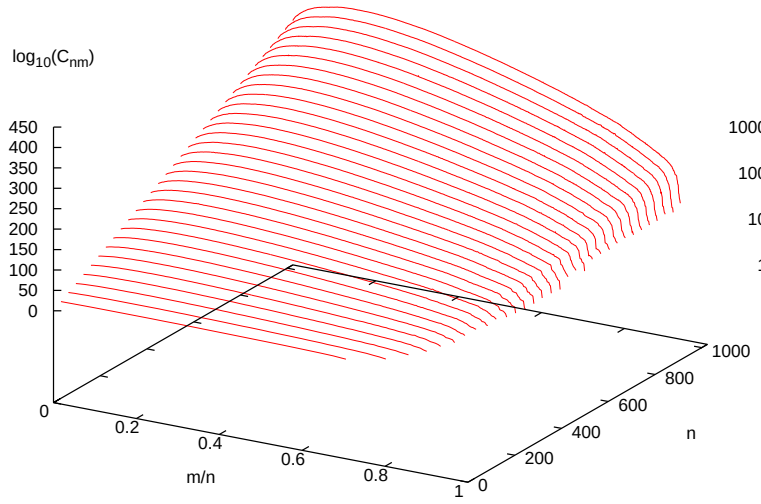


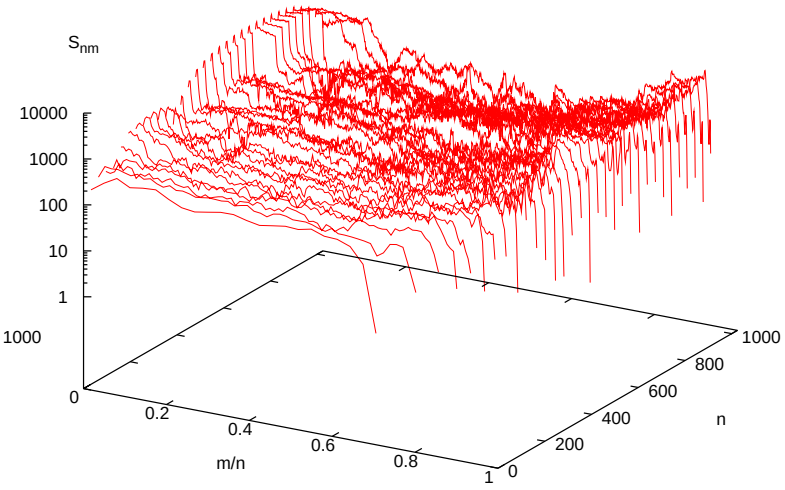


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $280,000,000$

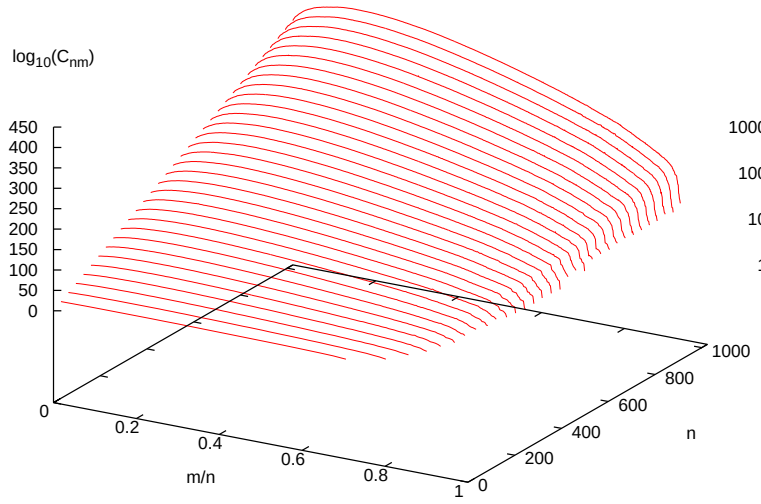


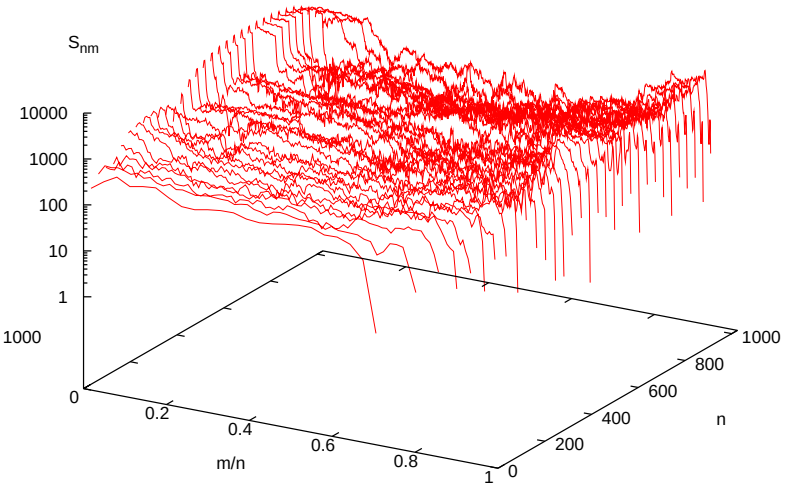


Simulation: dynamics of the algorithm

2d ISAW simulation up to $n = 1024$

- Total sample size: $290,000,000$

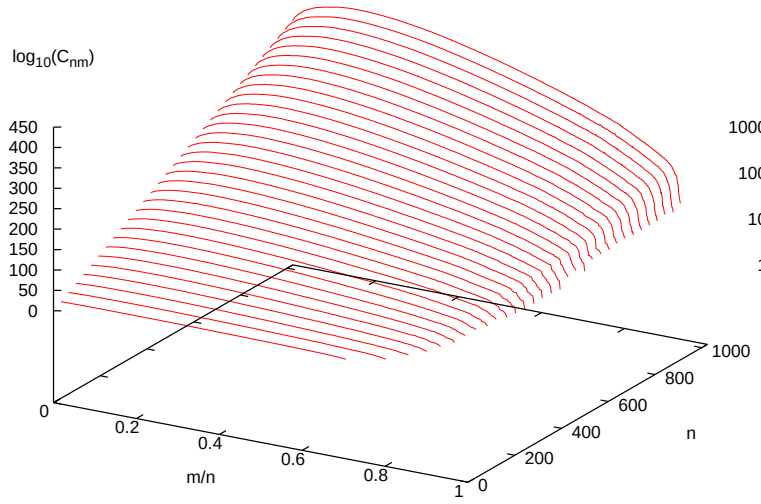


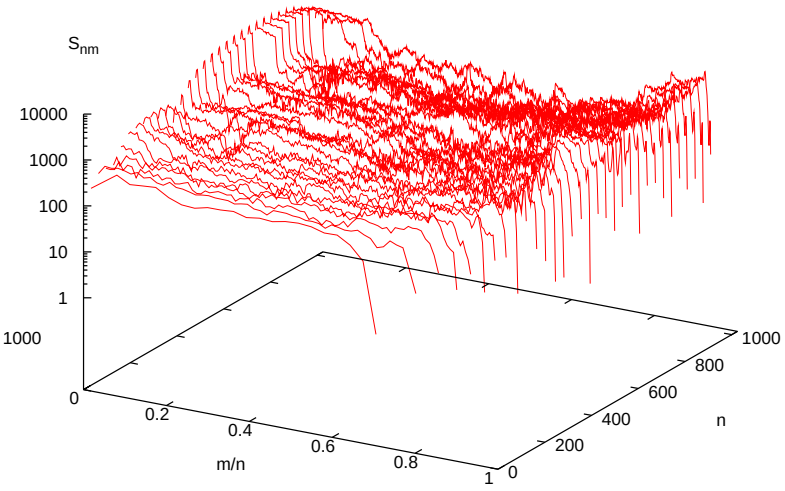


Simulation: dynamics of the algorithm

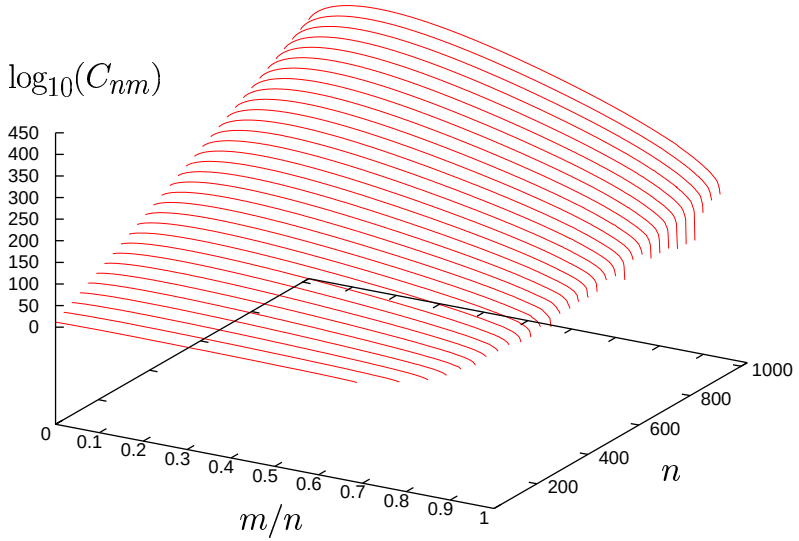
2d ISAW simulation up to $n = 1024$

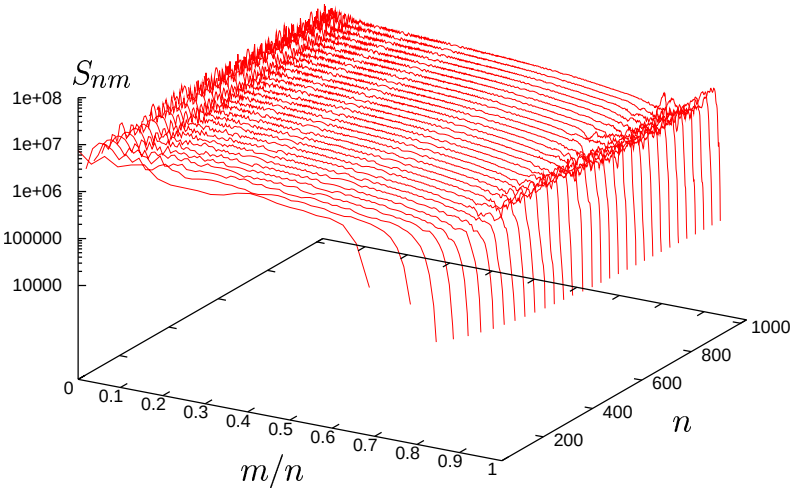
- Total sample size: $300,000,000$



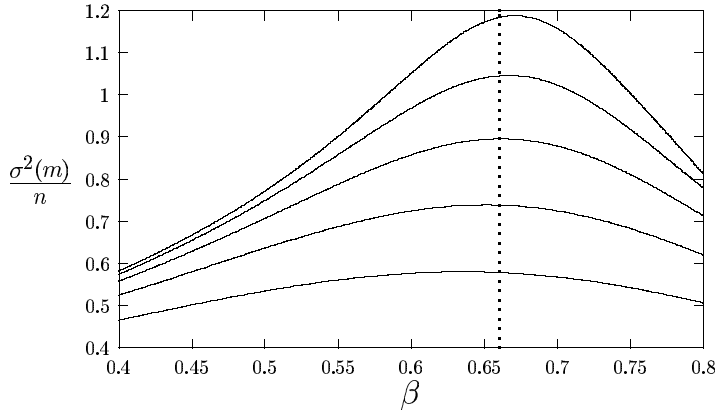


Simulation results: 2d ISAW

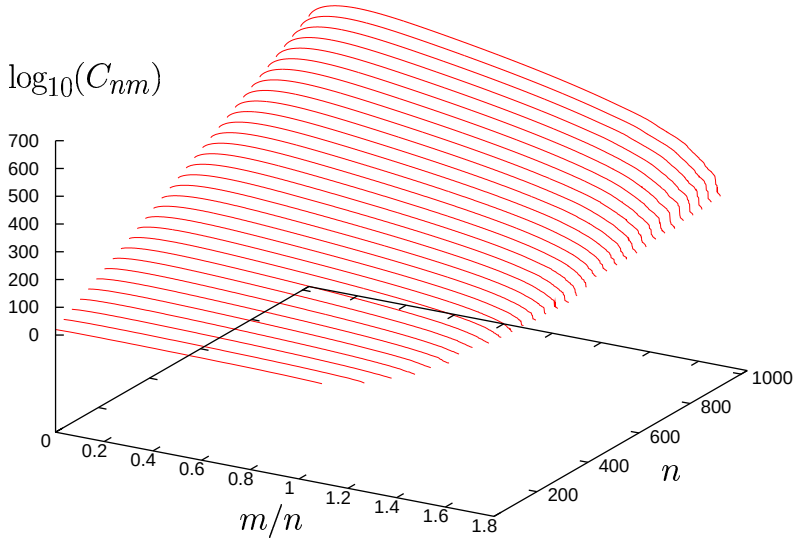


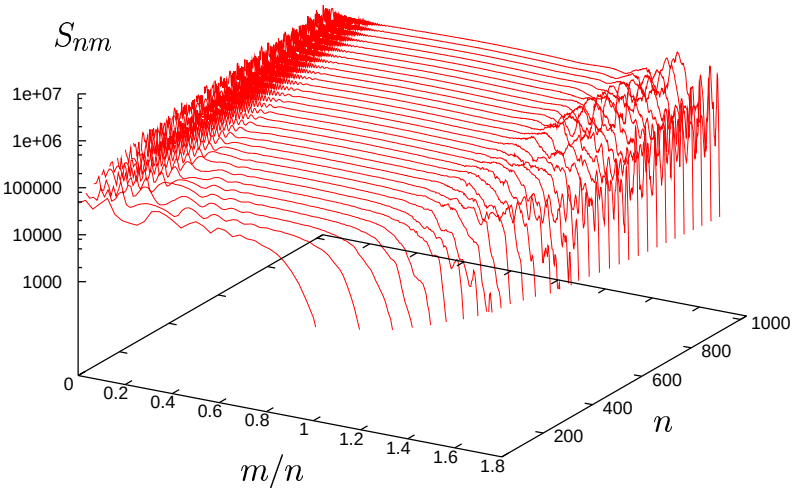


- 2d ISAW up to $n = 1024$
- One simulation suffices
- 300 orders of magnitude

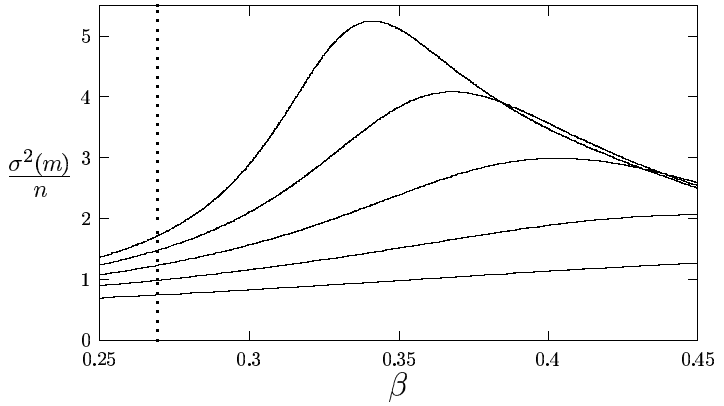


Simulation results: 3d ISAW





- 3d ISAW up to $n = 1024$
- One simulation suffices
- 400 orders of magnitude



Summary

Conclusion: A Promising New Algorithm

- Reviewed Stochastic Growth Algorithms for Polymers
- Presented “flat histogram” version of PERM
  - *One* simulation for complete density of states!
    (the range can also be selectively restricted)
- Applications: SAW in a strip, HP model, ISAW

Conclusion: A Promising New Algorithm

- Reviewed Stochastic Growth Algorithms for Polymers
- Presented “flat histogram” version of PERM
  - *One* simulation for complete density of states!
    (the range can also be selectively restricted)
- Applications: SAW in a strip, HP model, ISAW

Outlook: applications to further models, e.g.

- Square lattice trees / branched polymers: with A. Rechnitzer
  - Connective constant estimate $5.1435(5)$
    (compare to $5.1434(7)$ with Pivot algorithm)
- Two-dimensional density of states with *one* simulation
  - Absorbing collapsing polymers: with A. Owczarek
  - Extended Domb-Joyce model: with J. Krawczyk

The End