How to calculate the connective constant for self-avoiding walks really, really accurately

Nathan Clisby
MASCOS, The University of Melbourne

Annual Statistical Mechanics Meeting
The University of Melbourne
December 1, 2011

Outline

Outline

- Introduction
- Advances in algorithms for SAW
- Finding a suitable observable
- Advances in Monte Carlo “move sets”
- Minimizing statistical error
- Conclusion

Outline

- Introduction
- Advances in algorithms for SAW
- Finding a suitable observable
- Advances in Monte Carlo “move sets”
- Minimizing statistical error
- Conclusion

Outline

- Introduction
- Advances in algorithms for SAW
- Finding a suitable observable
- Advances in Monte Carlo "move sets"
- Minimizing statistical error
- Conclusion

Outline

- Introduction
- Advances in algorithms for SAW
- Finding a suitable observable
- Advances in Monte Carlo “move sets”
- Minimizing statistical error
- Conclusion

Outline

- Introduction
- Advances in algorithms for SAW
- Finding a suitable observable
- Advances in Monte Carlo “move sets”
- Minimizing statistical error
- Conclusion

Self-avoiding walk model

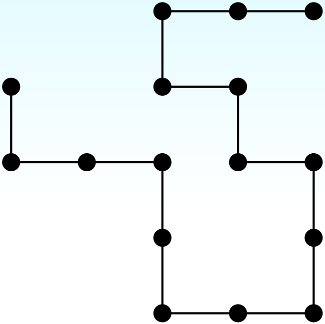
- A walk on a lattice, step to neighbouring site provided it has not already been visited.
- Models polymers in good solvent limit.
- Exactly captures universal properties such as critical exponents.

Self-avoiding walk model

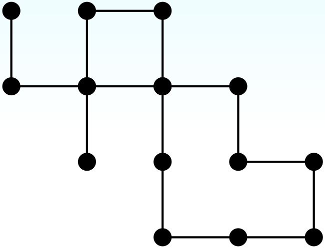
- A walk on a lattice, step to neighbouring site provided it has not already been visited.
- Models polymers in good solvent limit.
- Exactly captures universal properties such as critical exponents.

Self-avoiding walk model

- A walk on a lattice, step to neighbouring site provided it has not already been visited.
- Models polymers in good solvent limit.
- Exactly captures universal properties such as critical exponents.



SAW



Not a SAW

Animation

Critical phenomena

- The number of SAW of length N, c_N, tells us about how many conformations are available to SAW of a particular length:

$$c_N \sim A\ N^{\gamma-1}\mu^N\,[1+\text{corrections}]$$

- γ is a universal exponent.
- μ is the connective constant; lattice dependent.

Critical phenomena

- The number of SAW of length N, c_N, tells us about how many conformations are available to SAW of a particular length:

$$c_N \sim A\ N^{\gamma-1}\mu^N\,[1+\mathrm{corrections}]$$

- γ is a universal exponent.
- μ is the connective constant; lattice dependent.

Critical phenomena

- The number of SAW of length N, c_N, tells us about how many conformations are available to SAW of a particular length:

$$c_N \sim A\ N^{\gamma-1}\mu^N\,[1+\mathrm{corrections}]$$

- γ is a universal exponent.
- μ is the connective constant; lattice dependent.

Pivot algorithm

- Sample from the set of SAWs of a particular length.
- Markov chain:
  - Select a pivot site *uniformly at random*.
  - Randomly choose a lattice symmetry g (rotation or reflection).
  - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.

Pivot algorithm

- Sample from the set of SAWs of a particular length.
- Markov chain:
    - Select a pivot site *uniformly at random*.
    - Randomly choose a lattice symmetry q (rotation or reflection).
    - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
    - If walk is self-avoiding: *accept* the pivot and update the configuration.
    - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.

Pivot algorithm

- Sample from the set of SAWs of a particular length.
- Markov chain:
  - Select a pivot site *uniformly at random*.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.

Pivot algorithm

- Sample from the set of SAWs of a particular length.
- Markov chain:
  - Select a pivot site *uniformly at random*.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.

Pivot algorithm

- Sample from the set of SAWs of a particular length.
- Markov chain:
  - Select a pivot site *uniformly at random*.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.

Pivot algorithm

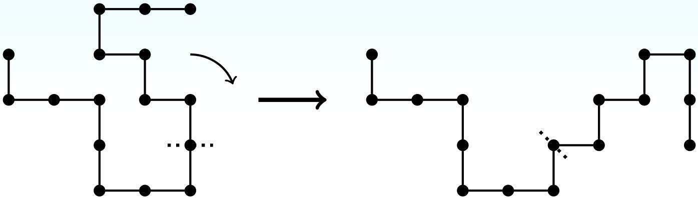
- Sample from the set of SAWs of a particular length.
- Markov chain:
  - Select a pivot site *uniformly at random*.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.

Pivot algorithm

- Sample from the set of SAWs of a particular length.
- Markov chain:
  - Select a pivot site *uniformly at random*.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.

Pivot algorithm

- Sample from the set of SAWs of a particular length.
- Markov chain:
  - Select a pivot site *uniformly at random*.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two sub-walks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- Ergodic, samples SAWs uniformly at random.



Example pivot move

Why is it so effective?

- Pivots are rarely successful, $\Pr = O(N^{-p})$, $p \approx 0.11$ for $\mathbb{Z}^3$.
- Every time a pivot attempt *is* successful there is a large change in global observables.
- Only need $O(1)$ successful pivots before we have an *essentially new* configuration.
- $\Rightarrow \tau_{\text{int}} = O(N^p)$

Why is it so effective?

- Pivots are rarely successful, $\Pr = O(N^{-p})$, $p \approx 0.11$ for $\mathbb{Z}^3$.
- Every time a pivot attempt *is* successful there is a large change in global observables.
- Only need $O(1)$ successful pivots before we have an *essentially new* configuration.
- $\Rightarrow \tau_{\text{int}} = O(N^p)$

Why is it so effective?

- Pivots are rarely successful, $\Pr = O(N^{-p})$, $p \approx 0.11$ for $\mathbb{Z}^3$.
- Every time a pivot attempt *is* successful there is a large change in global observables.
- Only need $O(1)$ successful pivots before we have an *essentially new* configuration.
- $\Rightarrow \tau_{\text{int}} = O(N^p)$

Why is it so effective?

- Pivots are rarely successful, $\Pr = O(N^{-p})$, $p \approx 0.11$ for $\mathbb{Z}^3$.
- Every time a pivot attempt *is* successful there is a large change in global observables.
- Only need $O(1)$ successful pivots before we have an *essentially new* configuration.
- $\Rightarrow \tau_{\mathrm{int}} = O(N^p)$

Looking in Frauenkron et al.[1] on Tuesday, for different reason:
"For SAWs in planar geometry, the fastest known algorithm is the pivot algorithm - at least if one is not interested in mu"

As we'll see, the pivot algorithm does a pretty good job of calculating μ.

[1] Helge Frauenkron, Maria Serena Causo, and Peter Grassberger. "Two-dimensional self-avoiding walks on a cylinder". In: *Phys. Rev. E* 59 (1 1999), R16–R19.

Looking in Frauenkron et al.[1] on Tuesday, for different reason:
"For SAWs in planar geometry, the fastest known algorithm is the pivot algorithm - at least if one is not interested in mu"

As we'll see, the pivot algorithm does a pretty good job of calculating μ.

[1] Helge Frauenkron, Maria Serena Causo, and Peter Grassberger. "Two-dimensional self-avoiding walks on a cylinder". In: *Phys. Rev. E* 59 (1 1999), R16–R19.

- Represent SAW as a binary tree.
- Enables global moves like pivots to be performed in CPU time $T(N) = O(\log N)$.
- Define: $\widetilde{\tau}_{\mathrm{int}} = \tau_{\mathrm{int}} T(N)$, integrated autocorrelation time in CPU units.
- When estimating mean value of an observable, run for CPU time t: Error $\propto \sqrt{\widetilde{\tau}_{\mathrm{int}}/t}$.
- SAW-tree: $\widetilde{\tau}_{\mathrm{int}} = O(N^p \log N)$ for pivot algorithm (c.f. $O(N)^2$).
- Dramatic improvement for large N.
- Thus far $\nu = 0.587597(7)$[3] and $\gamma = 1.156957(9)$ (in preparation).

---

[2] Neal Madras and Alan D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[3] N. Clisby. "Accurate Estimate of the Critical Exponent ν for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm". In: *Phys. Rev. Lett.* 104 (2010), p. 055702.

- Represent SAW as a binary tree.
- Enables global moves like pivots to be performed in CPU time $T(N) = O(\log N)$.
- Define: $\widetilde{\tau}_{\mathrm{int}} = \tau_{\mathrm{int}} T(N)$, integrated autocorrelation time in CPU units.
- When estimating mean value of an observable, run for CPU time t: Error $\propto \sqrt{\widetilde{\tau}_{\mathrm{int}}/t}$.
- SAW-tree: $\widetilde{\tau}_{\mathrm{int}} = O(N^p \log N)$ for pivot algorithm (c.f. $O(N)$[2]).
- Dramatic improvement for large N.
- Thus far $\nu = 0.587597(7)$[3] and $\gamma = 1.156957(9)$ (in preparation).

---

[2] Neal Madras and Alan D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[3] N. Clisby. "Accurate Estimate of the Critical Exponent ν for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm". In: *Phys. Rev. Lett.* 104 (2010), p. 055702.

- Represent SAW as a binary tree.
- Enables global moves like pivots to be performed in CPU time $T(N) = O(\log N)$.
- Define: $\widetilde{\tau}_{\rm int} = \tau_{\rm int} T(N)$, integrated autocorrelation time in CPU units.
- When estimating mean value of an observable, run for CPU time t: Error $\propto \sqrt{\widetilde{\tau}_{\rm int}/t}$.
- SAW-tree: $\widetilde{\tau}_{\rm int} = O(N^p \log N)$ for pivot algorithm (c.f. $O(N)$[2]).
- Dramatic improvement for large N.
- Thus far $\nu = 0.587597(7)$[3] and $\gamma = 1.156957(9)$ (in preparation).

[2] Neal Madras and Alan D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[3] N. Clisby. "Accurate Estimate of the Critical Exponent ν for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm". In: *Phys. Rev. Lett.* 104 (2010), p. 055702.

- Represent SAW as a binary tree.
- Enables global moves like pivots to be performed in CPU time $T(N) = O(\log N)$.
- Define: $\widetilde{\tau}_{\mathrm{int}} = \tau_{\mathrm{int}} T(N)$, integrated autocorrelation time in CPU units.
- When estimating mean value of an observable, run for CPU time t: Error $\propto \sqrt{\widetilde{\tau}_{\mathrm{int}}/t}$.
- SAW-tree: $\widetilde{\tau}_{\mathrm{int}} = O(N^p \log N)$ for pivot algorithm (c.f. $O(N)$[2]).
- Dramatic improvement for large N.
- Thus far $\nu = 0.587597(7)$[3] and $\gamma = 1.156957(9)$ (in preparation).

[2] Neal Madras and Alan D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[3] N. Clisby. "Accurate Estimate of the Critical Exponent ν for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm". In: *Phys. Rev. Lett.* 104 (2010), p. 055702.

- Represent SAW as a binary tree.
- Enables global moves like pivots to be performed in CPU time $T(N) = O(\log N)$.
- Define: $\widetilde{\tau}_{\rm int} = \tau_{\rm int} T(N)$, integrated autocorrelation time in CPU units.
- When estimating mean value of an observable, run for CPU time t: Error $\propto \sqrt{\widetilde{\tau}_{\rm int}/t}$.
- SAW-tree: $\widetilde{\tau}_{\rm int} = O(N^p \log N)$ for pivot algorithm (c.f. $O(N)^2$).
- Dramatic improvement for large N.
- Thus far $\nu = 0.587597(7)$[3] and $\gamma = 1.156957(9)$ (in preparation).

[2]Neal Madras and Alan D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[3]N. Clisby. "Accurate Estimate of the Critical Exponent ν for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm". In: *Phys. Rev. Lett.* 104 (2010), p. 055702.

- Represent SAW as a binary tree.
- Enables global moves like pivots to be performed in CPU time $T(N) = O(\log N)$.
- Define: $\widetilde{\tau}_{\rm int} = \tau_{\rm int}T(N)$, integrated autocorrelation time in CPU units.
- When estimating mean value of an observable, run for CPU time t: Error $\propto \sqrt{\widetilde{\tau}_{\rm int}/t}$.
- SAW-tree: $\widetilde{\tau}_{\rm int} = O(N^p \log N)$ for pivot algorithm (c.f. $O(N)^2$).
- Dramatic improvement for large N.
- Thus far $\nu = 0.587597(7)$[3] and $\gamma = 1.156957(9)$ (in preparation).

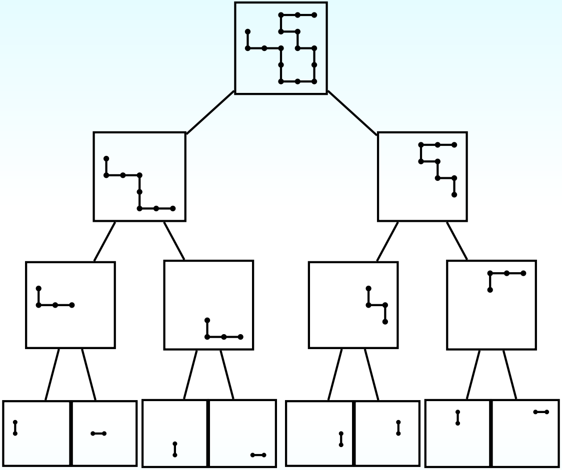
[2] Neal Madras and Alan D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[3] N. Clisby. "Accurate Estimate of the Critical Exponent ν for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm". In: *Phys. Rev. Lett.* 104 (2010), p. 055702.

- Represent SAW as a binary tree.
- Enables global moves like pivots to be performed in CPU time $T(N) = O(\log N)$.
- Define: $\widetilde{\tau}_{\mathrm{int}} = \tau_{\mathrm{int}} T(N)$, integrated autocorrelation time in CPU units.
- When estimating mean value of an observable, run for CPU time t: Error $\propto \sqrt{\widetilde{\tau}_{\mathrm{int}}/t}$.
- SAW-tree: $\widetilde{\tau}_{\mathrm{int}} = O(N^p \log N)$ for pivot algorithm (c.f. $O(N)^2$).
- Dramatic improvement for large N.
- Thus far $\nu = 0.587597(7)$[3] and $\gamma = 1.156957(9)$ (in preparation).

[2] Neal Madras and Alan D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[3] N. Clisby. "Accurate Estimate of the Critical Exponent ν for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm". In: *Phys. Rev. Lett.* 104 (2010), p. 055702.



SAW-tree representation of a walk.

How to calculate μ?

- Effectively need to count SAW to determine μ.
- Would like to apply pivot algorithm in canonical ensemble.
- Approach: measure probability that object from larger set is a SAW, $|S| = P(x \in S | x \in T)|T|$, with $|T|$ known.
- Obvious choice: concatenating pairs of SAWs.
- S_n set of walks of length n.
- $|S_{m+n}| = P(\omega_1 \circ \omega_2 \in S_{m+n} | (\omega_1, \omega_2) \in S_m \times S_n)|S_m||S_n|$
- Indicator function for successful concatenation is our observable, and

$$B(\omega_1, \omega_2) = \begin{cases} 0 & \text{if } \omega_1 \circ \omega_2 \text{ not self-avoiding} \\ 1 & \text{if } \omega_1 \circ \omega_2 \text{ self-avoiding} \end{cases}$$

How to calculate μ?

- Effectively need to count SAW to determine μ.
- Would like to apply pivot algorithm in canonical ensemble.
- Approach: measure probability that object from larger set is a SAW, $|S| = P(x \in S | x \in T)|T|$, with $|T|$ known.
- Obvious choice: concatenating pairs of SAWs.
- S_n set of walks of length n.
- $|S_{m+n}| = P(\omega_1 \circ \omega_2 \in S_{m+n} | (\omega_1, \omega_2) \in S_m \times S_n)|S_m||S_n|$
- Indicator function for successful concatenation is our observable, and

$$B(\omega_1, \omega_2) = \begin{cases} 0 & \text{if } \omega_1 \circ \omega_2 \text{ not self-avoiding} \\ 1 & \text{if } \omega_1 \circ \omega_2 \text{ self-avoiding} \end{cases}$$

How to calculate μ?

- Effectively need to count SAW to determine μ.
- Would like to apply pivot algorithm in canonical ensemble.
- Approach: measure probability that object from larger set is a SAW, $|S| = P(x \in S | x \in T)|T|$, with $|T|$ known.
- Obvious choice: concatenating pairs of SAWs.
- S_n set of walks of length n.
- $|S_{m+n}| = P(\omega_1 \circ \omega_2 \in S_{m+n} | (\omega_1, \omega_2) \in S_m \times S_n)|S_m||S_n|$
- Indicator function for successful concatenation is our observable, and

$$B(\omega_1, \omega_2) = \begin{cases} 0 & \text{if } \omega_1 \circ \omega_2 \text{ not self-avoiding} \\ 1 & \text{if } \omega_1 \circ \omega_2 \text{ self-avoiding} \end{cases}$$

How to calculate μ?

- Effectively need to count SAW to determine μ.
- Would like to apply pivot algorithm in canonical ensemble.
- Approach: measure probability that object from larger set is a SAW, $|S| = P(x \in S | x \in T)|T|$, with $|T|$ known.
- Obvious choice: concatenating pairs of SAWs.
- S_n set of walks of length n.
- $|S_{m+n}| = P(\omega_1 \circ \omega_2 \in S_{m+n} | (\omega_1, \omega_2) \in S_m \times S_n)|S_m||S_n|$
- Indicator function for successful concatenation is our observable, and

$$B(\omega_1, \omega_2) = \begin{cases} 0 & \text{if } \omega_1 \circ \omega_2 \text{ not self-avoiding} \\ 1 & \text{if } \omega_1 \circ \omega_2 \text{ self-avoiding} \end{cases}$$

How to calculate μ?

- Effectively need to count SAW to determine μ.
- Would like to apply pivot algorithm in canonical ensemble.
- Approach: measure probability that object from larger set is a SAW, $|S| = P(x \in S | x \in T)|T|$, with $|T|$ known.
- Obvious choice: concatenating pairs of SAWs.
- S_n set of walks of length n.
- $|S_{m+n}| = P(\omega_1 \circ \omega_2 \in S_{m+n} | (\omega_1, \omega_2) \in S_m \times S_n)|S_m||S_n|$
- Indicator function for successful concatenation is our observable, and

$$B(\omega_1, \omega_2) = \begin{cases} 0 & \text{if } \omega_1 \circ \omega_2 \text{ not self-avoiding} \\ 1 & \text{if } \omega_1 \circ \omega_2 \text{ self-avoiding} \end{cases}$$

How to calculate μ?

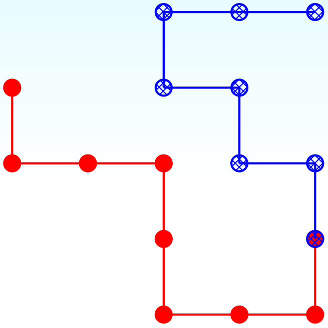
- Effectively need to count SAW to determine μ.
- Would like to apply pivot algorithm in canonical ensemble.
- Approach: measure probability that object from larger set is a SAW, $|S| = P(x \in S | x \in T)|T|$, with $|T|$ known.
- Obvious choice: concatenating pairs of SAWs.
- S_n set of walks of length n.
- $|S_{m+n}| = P(\omega_1 \circ \omega_2 \in S_{m+n} | (\omega_1, \omega_2) \in S_m \times S_n)|S_m||S_n|$
- Indicator function for successful concatenation is our observable, and

$$B(\omega_1, \omega_2) = \begin{cases} 0 & \text{if } \omega_1 \circ \omega_2 \text{ not self-avoiding} \\ 1 & \text{if } \omega_1 \circ \omega_2 \text{ self-avoiding} \end{cases}$$

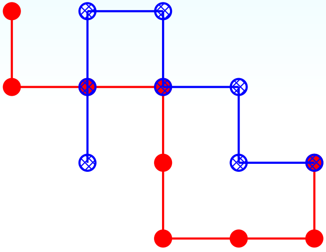
How to calculate μ?

- Effectively need to count SAW to determine μ.
- Would like to apply pivot algorithm in canonical ensemble.
- Approach: measure probability that object from larger set is a SAW, $|S| = P(x \in S | x \in T)|T|$, with $|T|$ known.
- Obvious choice: concatenating pairs of SAWs.
- S_n set of walks of length n.
- $|S_{m+n}| = P(\omega_1 \circ \omega_2 \in S_{m+n} | (\omega_1, \omega_2) \in S_m \times S_n)|S_m||S_n|$
- Indicator function for successful concatenation is our observable, and

$$B(\omega_1, \omega_2) = \begin{cases} 0 & \text{if } \omega_1 \circ \omega_2 \text{ not self-avoiding} \\ 1 & \text{if } \omega_1 \circ \omega_2 \text{ self-avoiding} \end{cases}$$



$B(\omega_1, \omega_2) = 1$



$B(\omega_1, \omega_2) = 0$

- Could choose $m = 36$ (longest known for $\mathbb{Z}^3$)

$$\langle B_{N,36} \rangle = \frac{c_{N+36}}{c_N c_{36}} = \frac{1}{c_{36}} \mu^{36} \left(1 + \frac{36(\gamma - 1)}{N} + \cdots \right)$$

- Direct calculation of μ, idea closely related to atmospheres[4].
- Effective, but can do better. Performance penalty due to large N, $\widetilde{\tau}_{\mathrm{int}} = O(N^p \log N)$.

[4] A. Rechnitzer and E. J. Janse van Rensburg. "Canonical Monte Carlo determination of the connective constant of self-avoiding walks". In: *J. Phys. A: Math. Gen.* 35 (2002), pp. L605–L612.

- Could choose $m = 36$ (longest known for $\mathbb{Z}^3$)

$$\begin{aligned} \langle B_{N,36} \rangle &= \frac{c_{N+36}}{c_N c_{36}} \\ &= \frac{1}{c_{36}} \mu^{36} \left(1 + \frac{36(\gamma - 1)}{N} + \cdots \right) \end{aligned}$$

- Direct calculation of μ, idea closely related to atmospheres[4].
- Effective, but can do better. Performance penalty due to large N, $\widetilde{\tau}_{\text{int}} = O(N^p \log N)$.

[4] A. Rechnitzer and E. J. Janse van Rensburg. "Canonical Monte Carlo determination of the connective constant of self-avoiding walks". In: *J. Phys. A: Math. Gen.* 35 (2002), pp. L605–L612.

- Could choose $m = 36$ (longest known for $\mathbb{Z}^3$)

$$\begin{aligned}\langle B_{N,36} \rangle &= \frac{c_{N+36}}{c_N c_{36}} \\ &= \frac{1}{c_{36}} \mu^{36} \left(1 + \frac{36(\gamma - 1)}{N} + \cdots \right)\end{aligned}$$

- Direct calculation of μ, idea closely related to atmospheres[4].
- Effective, but can do better. Performance penalty due to large N, $\widetilde{\tau}_{\text{int}} = O(N^p \log N)$.

[4] A. Rechnitzer and E. J. Janse van Rensburg. "Canonical Monte Carlo determination of the connective constant of self-avoiding walks". In: *J. Phys. A: Math. Gen.* 35 (2002), pp. L605–L612.

- Could choose $m, n = 36$:

$$\langle B_{36,36} \rangle = \frac{c_{72}}{c_{36} c_{36}} = \frac{1}{c_{36}^2} \mu^{72} \left(1 + \frac{\gamma - 1}{72} + \cdots \right)$$

- Need a way to improve these estimates and approach the asymptotic regime.
- Use:

$$c_N = \frac{c_N}{c_{N/2}^2} \cdot \frac{c_{N/2}^2}{c_{N/4}^4} \cdots \frac{\cdots}{c_k^{N/k}}$$

where c_k is known.

- Could choose $m, n = 36$:

$$\begin{aligned}\langle B_{36,36}\rangle &= \frac{c_{72}}{c_{36}c_{36}} \\ &= \frac{1}{c_{36}^2}\mu^{72}\left(1 + \frac{\gamma - 1}{72} + \cdots\right)\end{aligned}$$

- Need a way to improve these estimates and approach the asymptotic regime.
- Use:

$$c_N = \frac{c_N}{c_{N/2}^2} \cdot \frac{c_{N/2}^2}{c_{N/4}^4} \cdots \frac{\cdots}{c_k^{N/k}}$$

where c_k is known.

- Could choose $m, n = 36$:

$$\langle B_{36,36} \rangle = \frac{c_{72}}{c_{36} c_{36}} = \frac{1}{c_{36}^2} \mu^{72} \left(1 + \frac{\gamma - 1}{72} + \cdots \right)$$

- Need a way to improve these estimates and approach the asymptotic regime.
- Use:

$$c_N = \frac{c_N}{c_{N/2}^2} \cdot \frac{c_{N/2}^2}{c_{N/4}^4} \cdots \frac{\cdots}{c_k^{N/k}}$$

where c_k is known.

- In terms of B:

$$\begin{aligned}
\log \mu_N &= \frac{1}{N} \log c_N \\
&= \frac{1}{k} \log c_k + \frac{1}{2k} \log \frac{c_{2k}}{c_k^2} + \frac{1}{4k} \log \frac{c_{4k}}{c_{2k}^2} + \cdots \\
&\qquad \cdots + \frac{1}{N} \log \frac{c_N}{c_{N/2}^2} \\
&= \frac{1}{k} \log c_k + \frac{1}{2k} \log \langle B_{k,k} \rangle + \frac{1}{4k} \log \langle B_{2k,2k} \rangle + \cdots \\
&\qquad \cdots + \frac{1}{N} \log \langle B_{N/2,N/2} \rangle \\
&= \log \mu + \frac{(\gamma - 1) \log N}{N} + \frac{\log A}{N} + \text{corrections}
\end{aligned}$$

Scale free moves

- Need to calculate $\langle B_{k,k} \rangle$, $\langle B_{2k,2k} \rangle$, $\cdots$
- Use pivot algorithm / SAW-tree.
- How many pivots must be completed before two walks are essentially new configurations with respect to observable B?
- Shape of walks close to the joint clearly important.
- Uniform pivot sites: $\widetilde{\tau}_{\mathrm{int}} = \Omega(N)$.
- Choose distance from joint uniformly from all distance scales, i.e. $u = \log(\text{distance})$ chosen uniformly at random.
- Now: $\widetilde{\tau}_{\mathrm{int}} = N^p \log^2 N$.

Scale free moves

- Need to calculate $\langle B_{k,k} \rangle$, $\langle B_{2k,2k} \rangle$, $\cdots$
- Use pivot algorithm / SAW-tree.
- How many pivots must be completed before two walks are essentially new configurations with respect to observable B?
- Shape of walks close to the joint clearly important.
- Uniform pivot sites: $\widetilde{\tau}_{\mathrm{int}} = \Omega(N)$.
- Choose distance from joint uniformly from all distance scales, i.e. $u = \log(\text{distance})$ chosen uniformly at random.
- Now: $\widetilde{\tau}_{\mathrm{int}} = N^p \log^2 N$.

Scale free moves

- Need to calculate $\langle B_{k,k}\rangle$, $\langle B_{2k,2k}\rangle$, $\cdots$
- Use pivot algorithm / SAW-tree.
- How many pivots must be completed before two walks are essentially new configurations with respect to observable B?
- Shape of walks close to the joint clearly important.
- Uniform pivot sites: $\widetilde{\tau}_{\text{int}} = \Omega(N)$.
- Choose distance from joint uniformly from all distance scales, i.e. $u = \log(\text{distance})$ chosen uniformly at random.
- Now: $\widetilde{\tau}_{\text{int}} = N^p \log^2 N$.

Scale free moves

- Need to calculate $\langle B_{k,k} \rangle$, $\langle B_{2k,2k} \rangle$, $\cdots$
- Use pivot algorithm / SAW-tree.
- How many pivots must be completed before two walks are essentially new configurations with respect to observable B?
- Shape of walks close to the joint clearly important.
- Uniform pivot sites: $\widetilde{\tau}_{\mathrm{int}} = \Omega(N)$.
- Choose distance from joint uniformly from all distance scales, i.e. $u = \log(\text{distance})$ chosen uniformly at random.
- Now: $\widetilde{\tau}_{\mathrm{int}} = N^p \log^2 N$.

Scale free moves

- Need to calculate $\langle B_{k,k} \rangle$, $\langle B_{2k,2k} \rangle$, $\cdots$
- Use pivot algorithm / SAW-tree.
- How many pivots must be completed before two walks are essentially new configurations with respect to observable B?
- Shape of walks close to the joint clearly important.
- Uniform pivot sites: $\widetilde{\tau}_{\mathrm{int}} = \Omega(N)$.
- Choose distance from joint uniformly from all distance scales, i.e. $u = \log(\text{distance})$ chosen uniformly at random.
- Now: $\widetilde{\tau}_{\mathrm{int}} = N^p \log^2 N$.

Scale free moves

- Need to calculate $\langle B_{k,k} \rangle$, $\langle B_{2k,2k} \rangle$, $\cdots$
- Use pivot algorithm / SAW-tree.
- How many pivots must be completed before two walks are essentially new configurations with respect to observable B?
- Shape of walks close to the joint clearly important.
- Uniform pivot sites: $\widetilde{\tau}_{\mathrm{int}} = \Omega(N)$.
- Choose distance from joint uniformly from all distance scales, i.e. $u = \log(\text{distance})$ chosen uniformly at random.
- Now: $\widetilde{\tau}_{\mathrm{int}} = N^p \log^2 N$.

Scale free moves

- Need to calculate $\langle B_{k,k} \rangle$, $\langle B_{2k,2k} \rangle$, $\cdots$
- Use pivot algorithm / SAW-tree.
- How many pivots must be completed before two walks are essentially new configurations with respect to observable B?
- Shape of walks close to the joint clearly important.
- Uniform pivot sites: $\widetilde{\tau}_{\mathrm{int}} = \Omega(N)$.
- Choose distance from joint uniformly from all distance scales, i.e. $u = \log(\text{distance})$ chosen uniformly at random.
- Now: $\widetilde{\tau}_{\mathrm{int}} = N^p \log^2 N$.

Error estimate

- Expected error, for same CPU time, diminishes as a power law for higher order terms in the sum!

$$\log \mu_N = \frac{1}{k} \log c_k + \frac{1}{2k} \log\langle B_{k,k}\rangle + \cdots + \frac{1}{N} \log\langle B_{N/2,N/2}\rangle$$

- Choose N high enough so that corrections to scaling are completely negligible.
- Partition CPU time amongst different terms to minimize overall statistical error (short test run).

$$\sigma^2 = \sum \frac{a_i^2}{t_i} \qquad \text{Total time } t$$

$$\Rightarrow t_i = \frac{a_i}{\sum a_i} t, \qquad \sigma = \frac{\sum a_i}{\sqrt{t}}$$

- Can accurately predict error on estimate for μ prior to start of computer experiment.

Error estimate

- Expected error, for same CPU time, diminishes as a power law for higher order terms in the sum!

$$\log \mu_N = \frac{1}{k}\log c_k + \frac{1}{2k}\log\langle B_{k,k}\rangle + \cdots + \frac{1}{N}\log\langle B_{N/2,N/2}\rangle$$

- Choose N high enough so that corrections to scaling are completely negligible.
- Partition CPU time amongst different terms to minimize overall statistical error (short test run).

$$\sigma^2 = \sum \frac{a_i^2}{t_i} \qquad \text{Total time } t$$

$$\Rightarrow t_i = \frac{a_i}{\sum a_i} t, \qquad \sigma = \frac{\sum a_i}{\sqrt{t}}$$

- Can accurately predict error on estimate for μ prior to start of computer experiment.

Error estimate

- Expected error, for same CPU time, diminishes as a power law for higher order terms in the sum!

$$\log \mu_N = \frac{1}{k} \log c_k + \frac{1}{2k} \log\langle B_{k,k}\rangle + \cdots + \frac{1}{N} \log\langle B_{N/2,N/2}\rangle$$

- Choose N high enough so that corrections to scaling are completely negligible.
- Partition CPU time amongst different terms to minimize overall statistical error (short test run).

$$\sigma^2 = \sum \frac{a_i^2}{t_i} \qquad \text{Total time } t$$

$$\Rightarrow t_i = \frac{a_i}{\sum a_i} t, \qquad \sigma = \frac{\sum a_i}{\sqrt{t}}$$

- Can accurately predict error on estimate for μ prior to start of computer experiment.

Error estimate

- Expected error, for same CPU time, diminishes as a power law for higher order terms in the sum!

$$\log \mu_N = \frac{1}{k}\log c_k + \frac{1}{2k}\log\langle B_{k,k}\rangle + \cdots + \frac{1}{N}\log\langle B_{N/2,N/2}\rangle$$

- Choose N high enough so that corrections to scaling are completely negligible.
- Partition CPU time amongst different terms to minimize overall statistical error (short test run).

$$\sigma^2 = \sum \frac{a_i^2}{t_i} \qquad \text{Total time } t$$

$$\Rightarrow t_i = \frac{a_i}{\sum a_i} t, \qquad \sigma = \frac{\sum a_i}{\sqrt{t}}$$

- Can accurately predict error on estimate for μ prior to start of computer experiment.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.

Conclusion

- For $\mathbb{Z}^3$ we have:
- PERM: $\mu = 4.684038(6)$ (Hsu and Grassberger, "Polymers confined between two parallel plane walls")
- Series: $\mu = 4.68404(1)$ (Clisby, Liang, and Slade, "Self-avoiding walk enumeration via the lace expansion")
- Series: $\mu = 4.684040(5)$ (Schram, Barkema, and Bisseling, "Exact enumeration of self-avoiding walks")
- Pivot: $\mu = 4.6840xxxx(4)$, 100 times more accurate than previous best.
- Power law improvement in error from each of: SAW-tree, move set, and choice of observable.
- $\sigma = 4 \times 10^8$, 35000 CPU hours.
- Computer experiment to start soon.