A Quantum Cavity Method

and some applications to Monte-Carlo simulations

Francesco Zamponi
LPT, Ecole Normale Superieure

Collaborators:
F.Krzakala, ESPCI
A. Rosso, Orsay
G. Semerjian, ENS Paris

Phys. Rev. B 78, 134428 (2008)

See also: (Cavity)
C. Laumann, A. Scardicchio, S.L. Sondhi
Phys.Rev.B 78, 134424 (2008)
S. Knysh, V.N. Smelyanskiy
arXiv:0803.0149

See also (Monte-Carlo):
Beard-Wiese 96,
Prokof'ev et al. 98,
Rieger-Kawashima 99

Generalize the cavity method to quantum systems

Why?

- A consistent mean field theory for finite-connectivity quantum models:
  - distance between variables (correlation length)
  - fluctuations of the local environment (disorder)
  - localization phenomena (e.g. Anderson localization)
- Exact solution of quantum models on random graphs
  - phase diagram of random K-sat, q-col, ...
- Studies of quantum annealing or quantum information
- Monte-Carlo methods for disordered systems

Quantum Spins Model in Transverse Field

Hilbert space: $(|+\rangle, |-\rangle)^{\otimes N}$

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Quantum Spins Model in Transverse Field

Hilbert space: $(|+\rangle, |-\rangle)^{\otimes N}$ $\quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$$\mathcal{H} = E(\{\sigma^z\}) - \Gamma \sum_i \sigma_i^x$$

Classical part
Original interaction

Transverse field
New quantum interaction

Partition Function: $\quad Z = \mathrm{Tr}\; e^{-\beta\mathcal{H}}$

Quantum Spins Model in Transverse Field

Hilbert space: $(|+\rangle, |-\rangle)^{\otimes N}$ $\quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$$\mathcal{H} = E(\{\sigma^z\}) - \Gamma \sum_i \sigma_i^x$$

Classical part
Original interaction

Transverse field
New quantum interaction

Partition Function: $Z = \mathrm{Tr}\ e^{-\beta \mathcal{H}}$

Example: The Ising Ferromagnet

$$E = -J \sum_{<i,j>} S_i S_j \quad \Rightarrow \quad \mathcal{H} = -J \sum_{<i,j>} \sigma_i^z \sigma_j^z - \Gamma \sum_i \sigma_i^x$$

Two technically related questions:

1) How to simulate such models using the Heat Bath Monte Carlo Simulation ?

2) How to apply the Bethe-Peierls (Cavity/Message Passing/TAP....) approach to such models ?

Overview

- Heat bath for classical and quantum spins
- Cavity Method for classical and quantum spins
- Concusions and perspectives

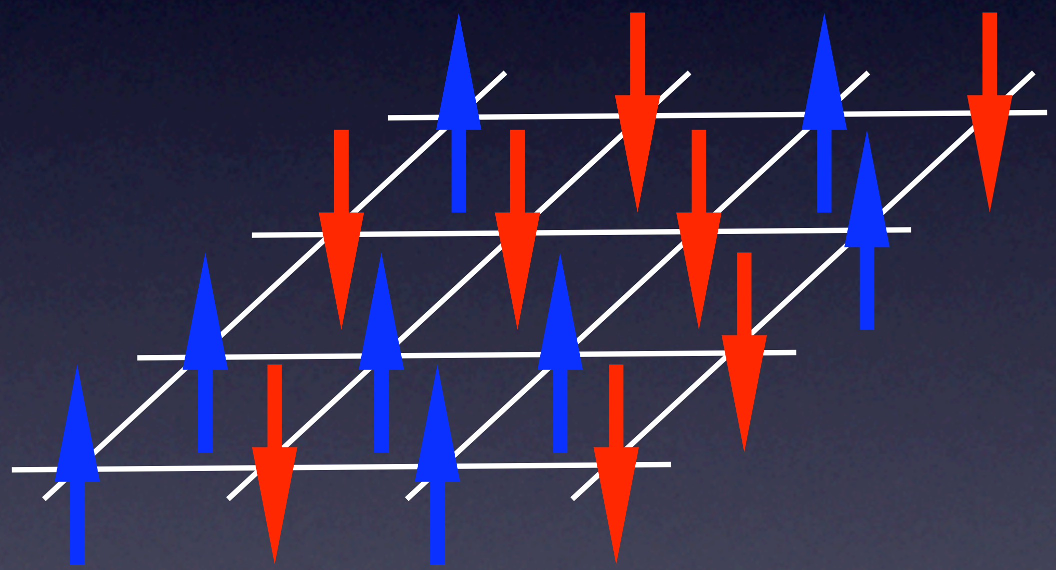
Overview

- Heat bath for classical and quantum spins
- Cavity Method for classical and quantum spins
- Concusions and perspectives

How to perform classical Monte-Carlo Simulations ?

The Heat-Bath Monte-Carlo algorithm

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j$$

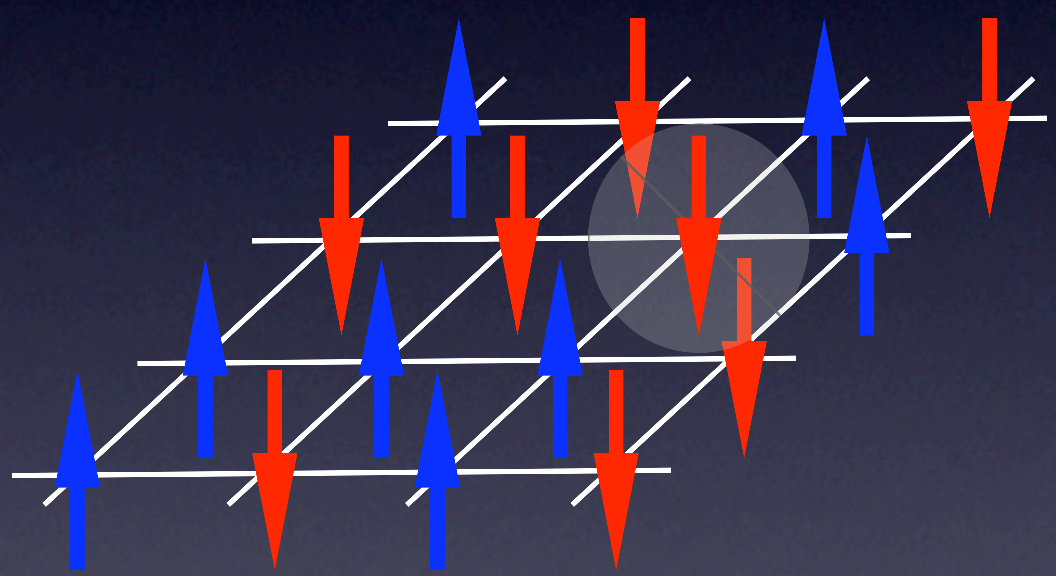


How to perform classical Monte-Carlo Simulations ?

The Heat-Bath Monte-Carlo algorithm

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j$$

1) Choose a spin at random



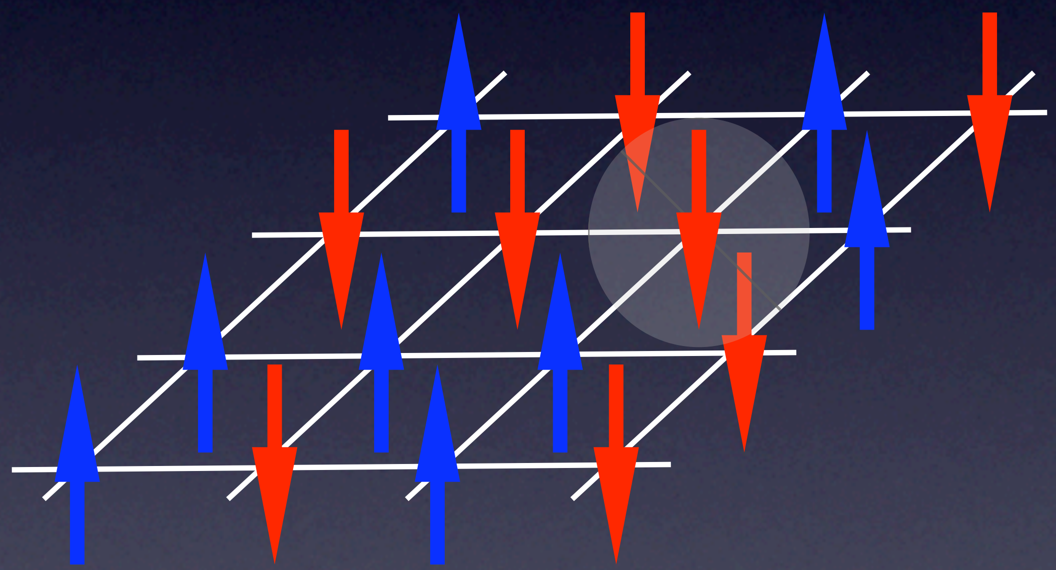
How to perform classical Monte-Carlo Simulations ?

The Heat-Bath Monte-Carlo algorithm

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j$$

1) Choose a spin at random

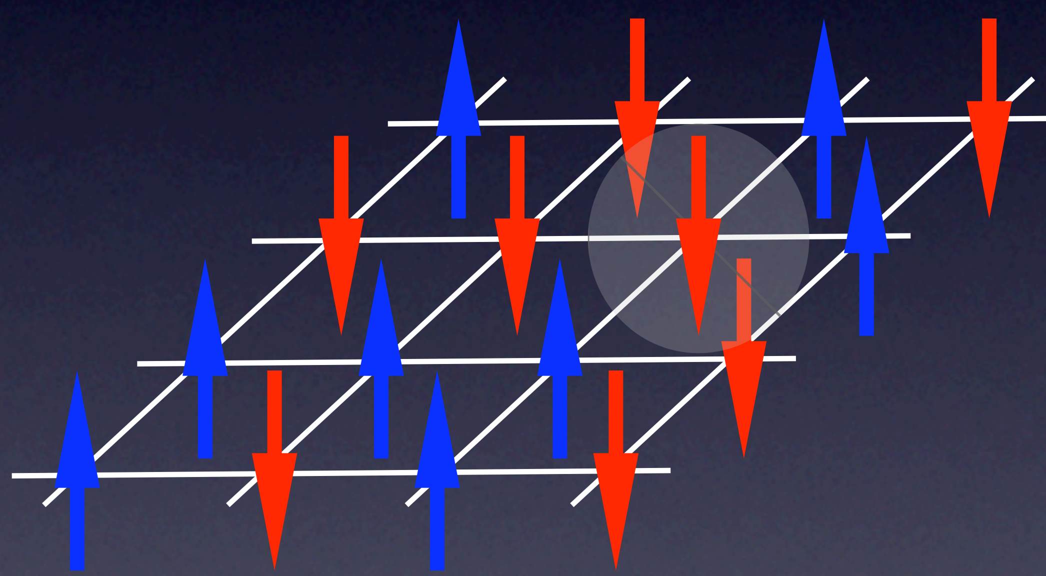
2) Compute its "local field"



How to perform classical Monte-Carlo Simulations ?

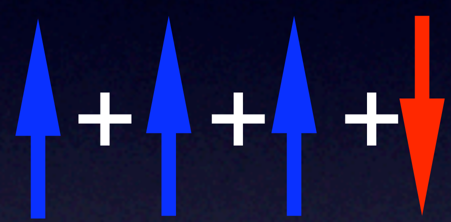
The Heat-Bath Monte-Carlo algorithm

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j$$



1) Choose a spin at random

2) Compute its "local field"



3) Choose the new value of the spin with Boltzman probability

$$p_{up} = \frac{e^{2\beta}}{Z} \qquad p_{down} = \frac{e^{-2\beta}}{Z}$$

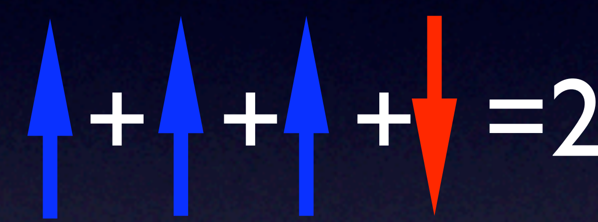
How to perform classical Monte-Carlo Simulations ?

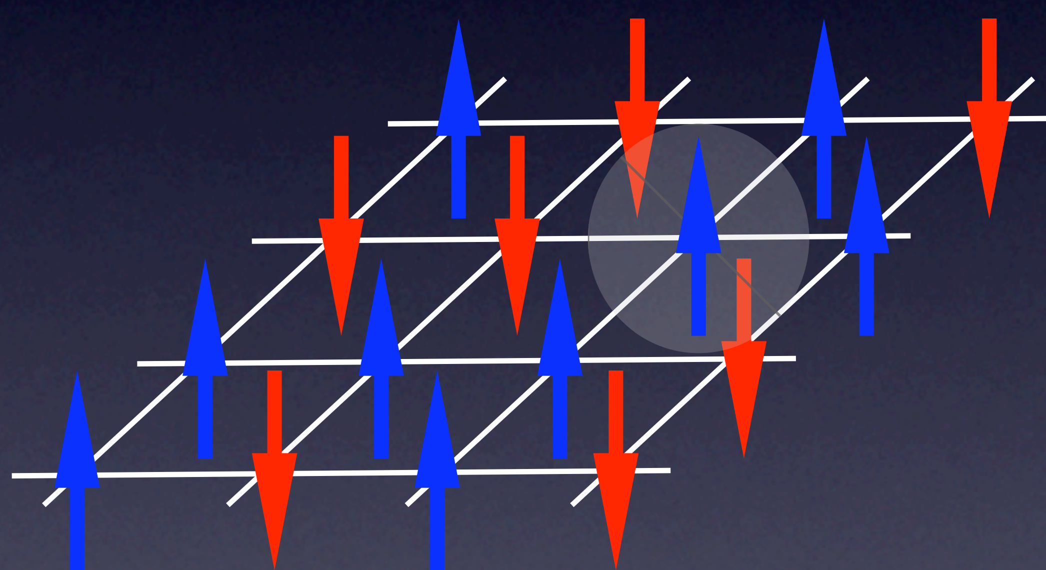
The Heat-Bath Monte-Carlo algorithm

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j$$

1) Choose a spin at random

2) Compute its "local field"





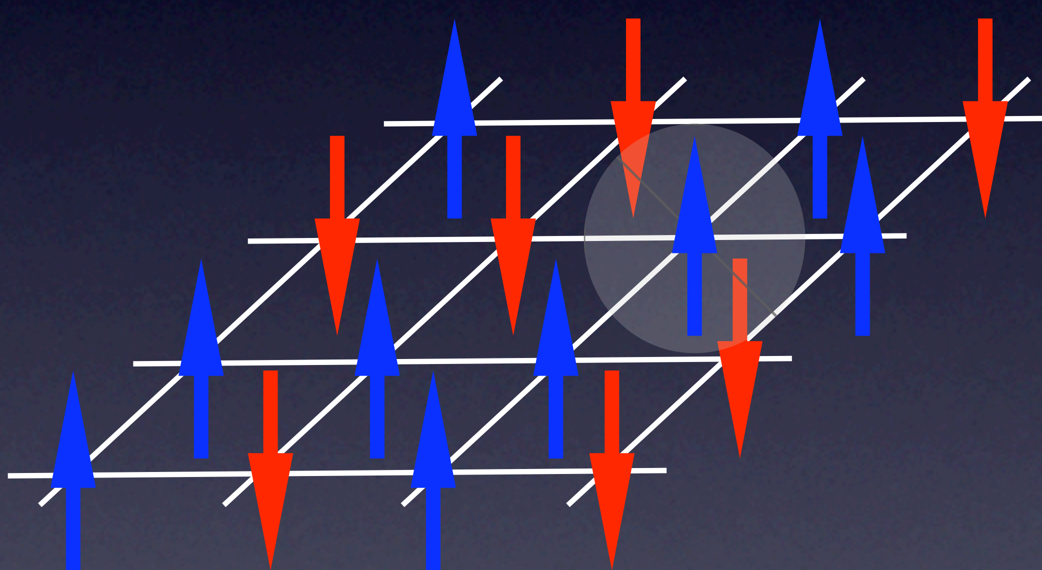
3) Choose the new value of the spin with Boltzman probability

$$p_{up} = \frac{e^{2\beta}}{Z} \qquad p_{down} = \frac{e^{-2\beta}}{Z}$$

How to perform classical Monte-Carlo Simulations ?

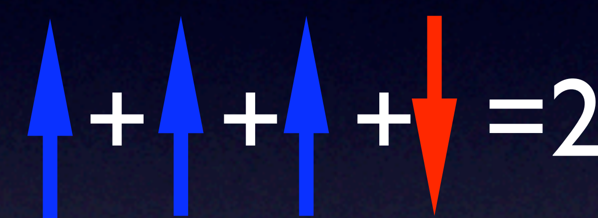
The Heat-Bath Monte-Carlo algorithm

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j$$



1) Choose a spin at random

2) Compute its "local field"



3) Choose the new value of the spin with Boltzman probability

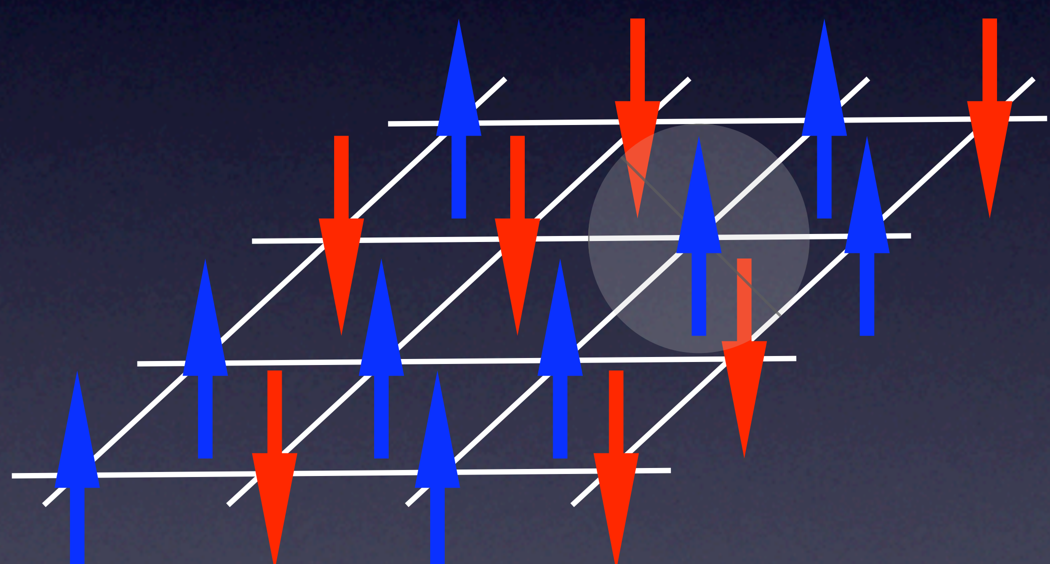
$$p_{up} = \frac{e^{2\beta}}{Z} \qquad p_{down} = \frac{e^{-2\beta}}{Z}$$

4) ...and repeat....

How to perform classical Monte-Carlo Simulations ?

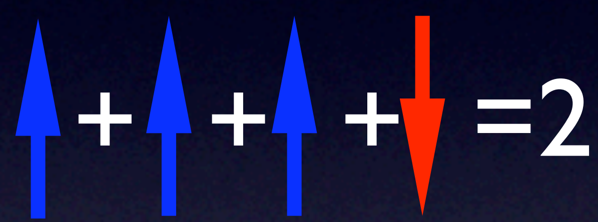
The Heat-Bath Monte-Carlo algorithm

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j$$



1) Choose a spin at random

2) Compute its "local field"



3) Choose the new value of the spin with Boltzman probability

$$p_{up} = \frac{e^{2\beta}}{Z} \qquad p_{down} = \frac{e^{-2\beta}}{Z}$$

4) ...and repeat....

How to generalize this procedure to the quantum case?

Suzuki-Trotter

$$Z = \mathrm{Tr}\left(e^{-\beta\widehat{E}+\beta\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)$$

Suzuki-Trotter

$$Z = \mathrm{Tr}\left(\left(e^{-\frac{\beta}{N_s}\widehat{E}+\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)^{N_s}\right)$$

Suzuki-Trotter

$$Z = \mathrm{Tr}\left(\left(e^{-\frac{\beta}{N_s}\widehat{E}+\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)^{N_s}\right)$$

In the large Ns limit

$$= \lim_{N_s\to\infty}\mathrm{Tr}\left(\left(e^{-\frac{\beta}{N_s}\widehat{E}}e^{\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)^{N_s}\right)$$

Suzuki-Trotter

$$Z = \mathrm{Tr}\left(\left(e^{-\frac{\beta}{N_s}\widehat{E}+\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)^{N_s}\right)$$

In the large Ns limit

$$= \lim_{N_s\to\infty}\mathrm{Tr}\left(\prod_{\alpha=1}^{N_s} e^{-\frac{\beta}{N_s}\widehat{E}} e^{\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)$$

Suzuki-Trotter

$$Z = \mathrm{Tr}\left(\left(e^{-\frac{\beta}{N_s}\widehat{E}+\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)^{N_s}\right)$$

In the large Ns limit

$$= \lim_{N_s\to\infty}\mathrm{Tr}\left(\prod_{\alpha=1}^{N_s} e^{-\frac{\beta}{N_s}\widehat{E}} e^{\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)$$

Use Ns relation (in the "z-base")

$$\sum_{\underline{\sigma}^\alpha} |\underline{\sigma}^\alpha\rangle\langle\underline{\sigma}^\alpha| = 1$$

where the vector are the set of 2^N "classical" configurations in the z-direction

Suzuki-Trotter

$$Z = \mathrm{Tr}\left(\left(e^{-\frac{\beta}{N_s}\widehat{E}+\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)^{N_s}\right)$$

In the large Ns limit

$$= \lim_{N_s\to\infty} \mathrm{Tr}\left(\prod_{\alpha=1}^{N_s} e^{-\frac{\beta}{N_s}\widehat{E}} e^{\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)$$

$$Z = \lim_{N_s\to\infty} \sum_{\underline{\sigma}^1,\ldots,\underline{\sigma}^{N_s}} \prod_{\alpha=1}^{N_s} \langle\underline{\sigma}^\alpha| e^{-\frac{\beta}{N_s}\widehat{E}}\; e^{\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x} |\underline{\sigma}^{\alpha+1}\rangle$$

Use Ns relation (in the "z-base")

$$\sum_{\underline{\sigma}^\alpha} |\underline{\sigma}^\alpha\rangle\langle\underline{\sigma}^\alpha| = 1$$

where the vector are the set of 2^N "classical" configurations in the z-direction

Suzuki-Trotter

$$Z = \mathrm{Tr}\left(\left(e^{-\frac{\beta}{N_s}\widehat{E}+\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)^{N_s}\right)$$

In the large Ns limit

$$= \lim_{N_s\to\infty} \mathrm{Tr}\left(\prod_{\alpha=1}^{N_s} e^{-\frac{\beta}{N_s}\widehat{E}} e^{\frac{\beta}{N_s}\Gamma\sum_{i=1}^{N}\sigma_i^x}\right)$$

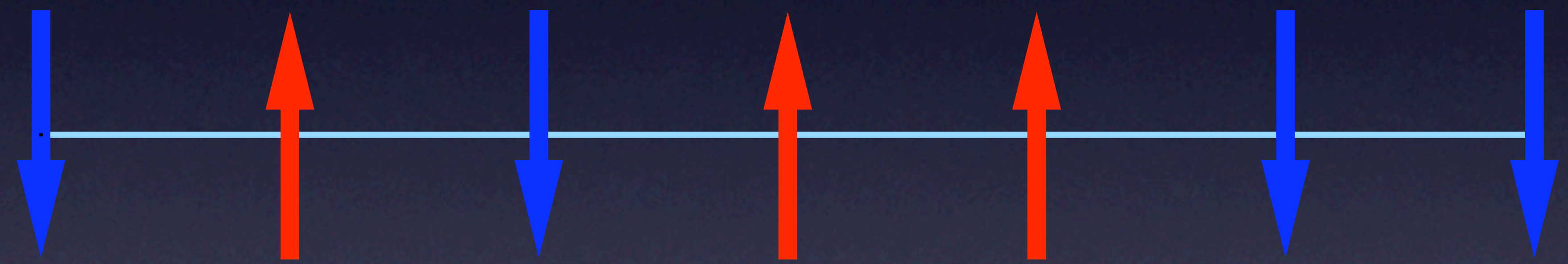
$$Z = \lim_{N_s\to\infty} \sum_{\underline{\sigma}^1,\ldots,\underline{\sigma}^{N_s}} \prod_{\alpha=1}^{N_s} e^{-\frac{\beta}{N_s}E(\underline{\sigma}^\alpha)} \prod_{i=1}^{N}\prod_{\alpha=1}^{N_s} \langle\sigma_i^\alpha| e^{\frac{\beta}{N_s}\Gamma\sigma^x} |\sigma_i^{\alpha+1}\rangle$$

Use Ns relation (in the "z-base")

$$\sum_{\underline{\sigma}^\alpha} |\underline{\sigma}^\alpha\rangle\langle\underline{\sigma}^\alpha| = 1$$

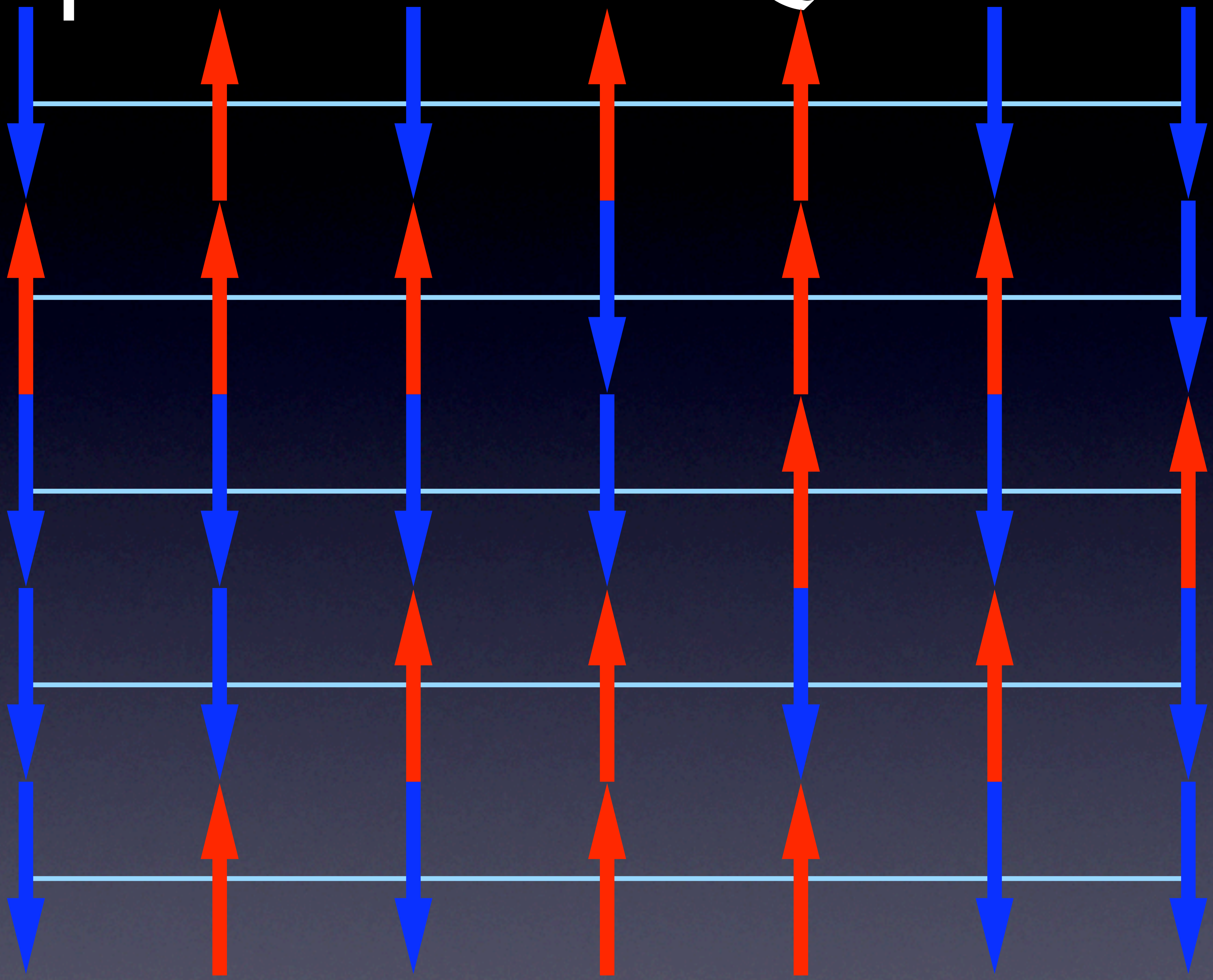
where the vector are the set of 2^N "classical" configurations in the z-direction

Example for the 1d Quantum Chain



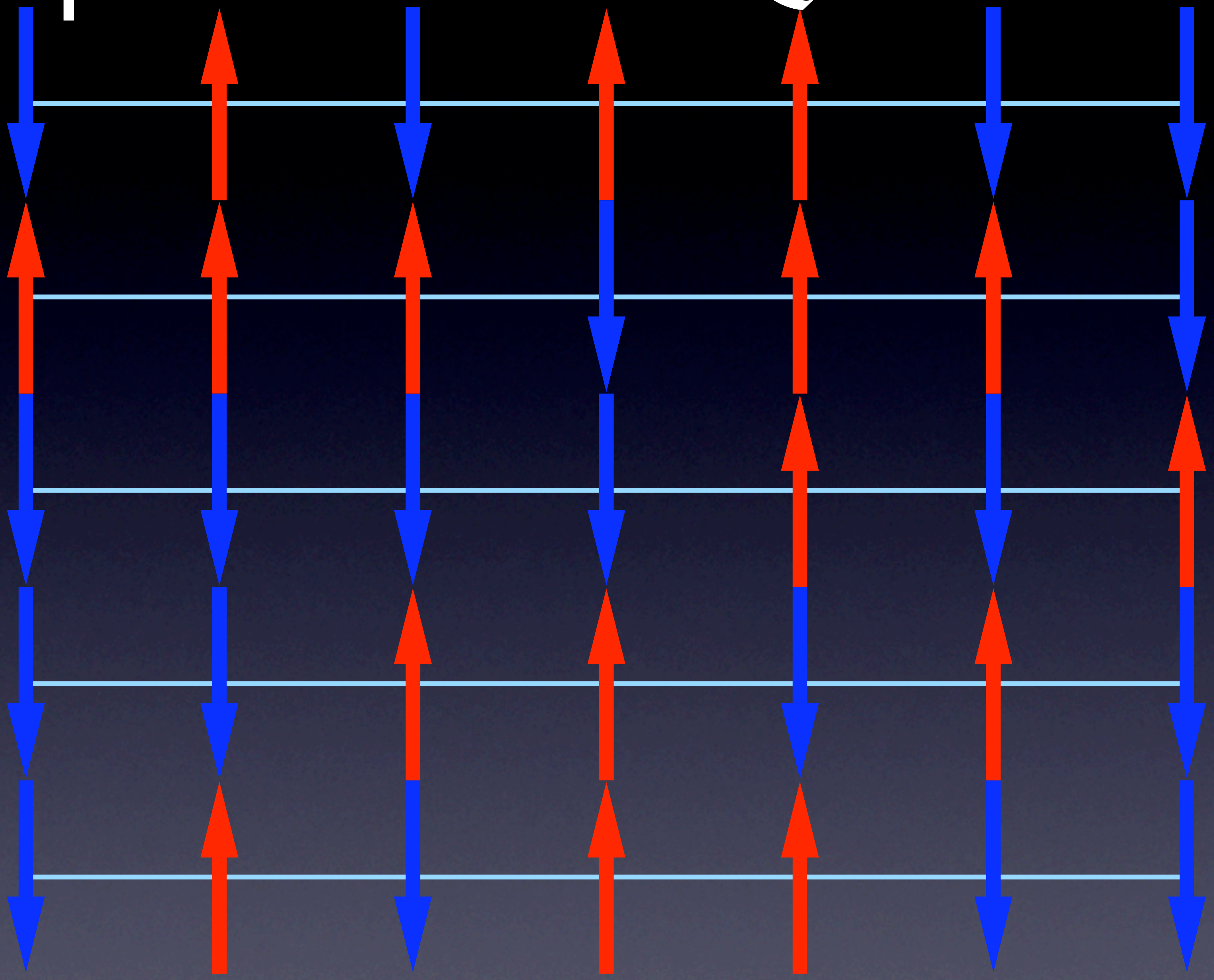
Consider the Original “classical” system

Example for the 1d Quantum Chain



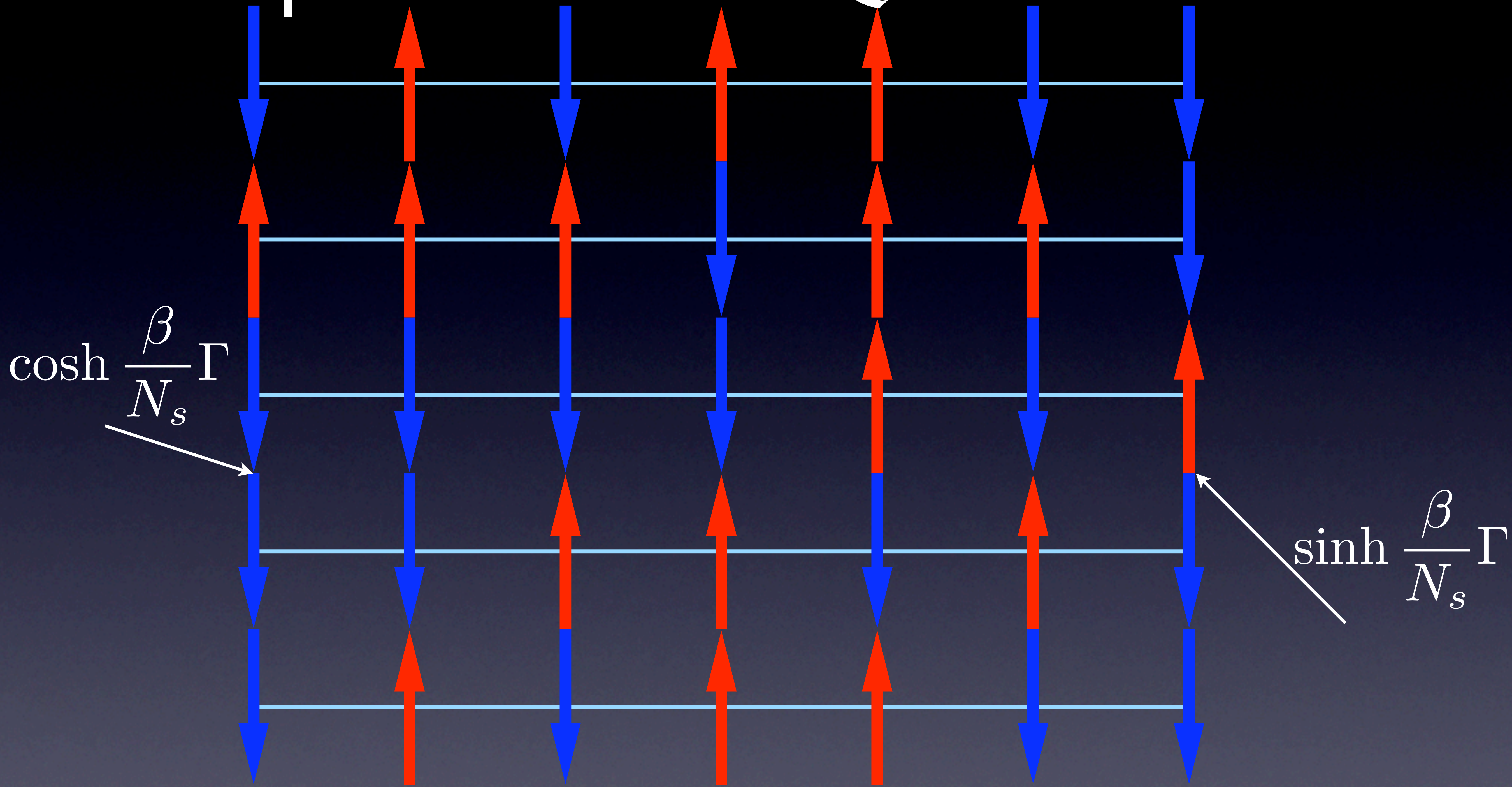
Duplicate the system Ns times

Example for the 1d Quantum Chain



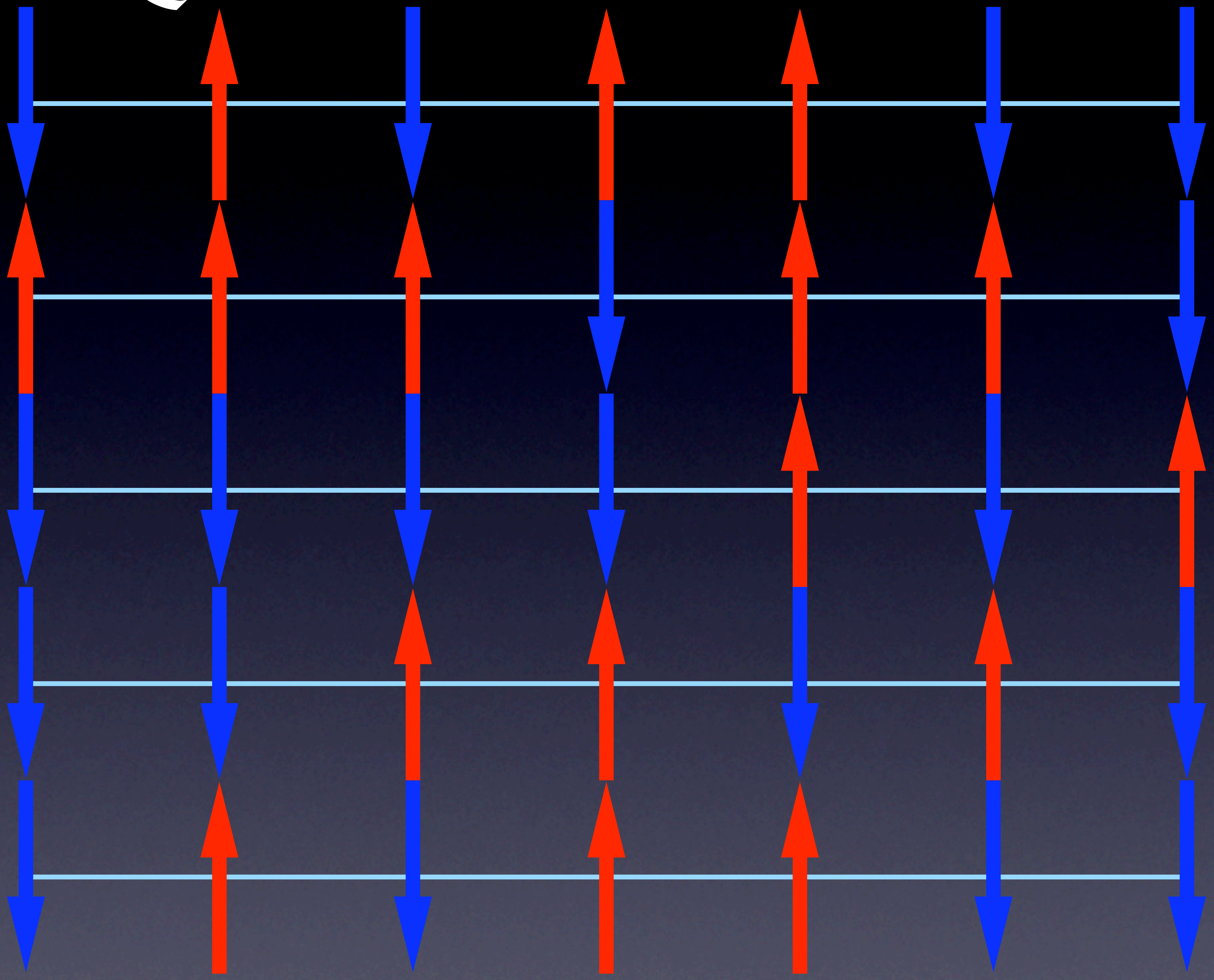
And obtain a system with d+1 dimension

Example for the 1d Quantum Chain



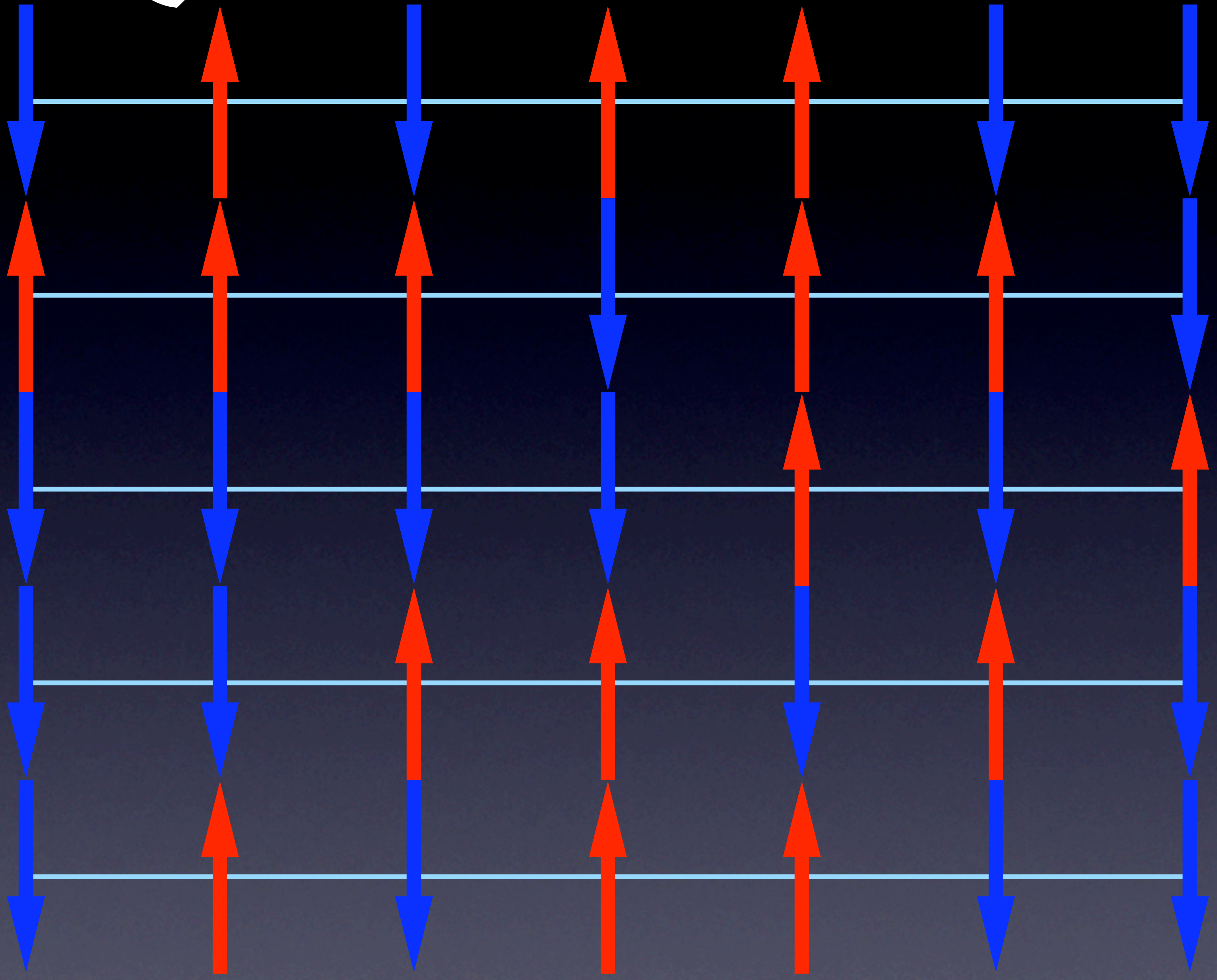
And obtain a system with d+1 dimension
With additional couplings

Quantum Monte Carlo

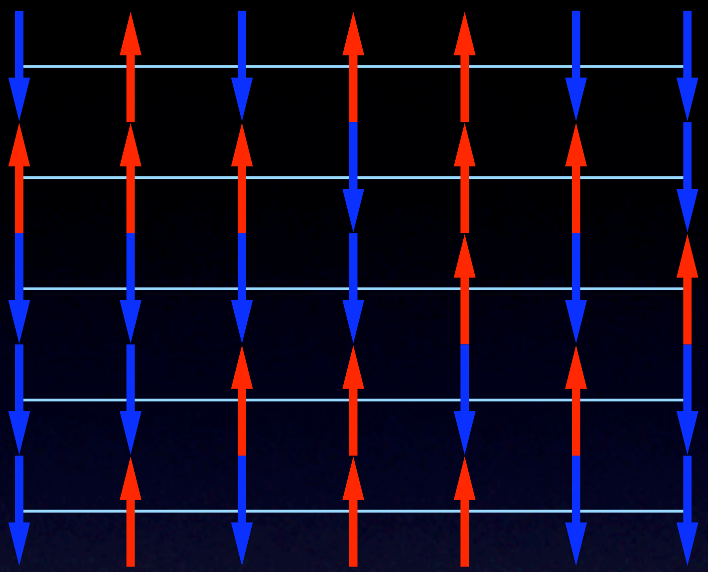


Perform a Classical Monte Carlo on the d+1 Lattice

Quantum Monte Carlo



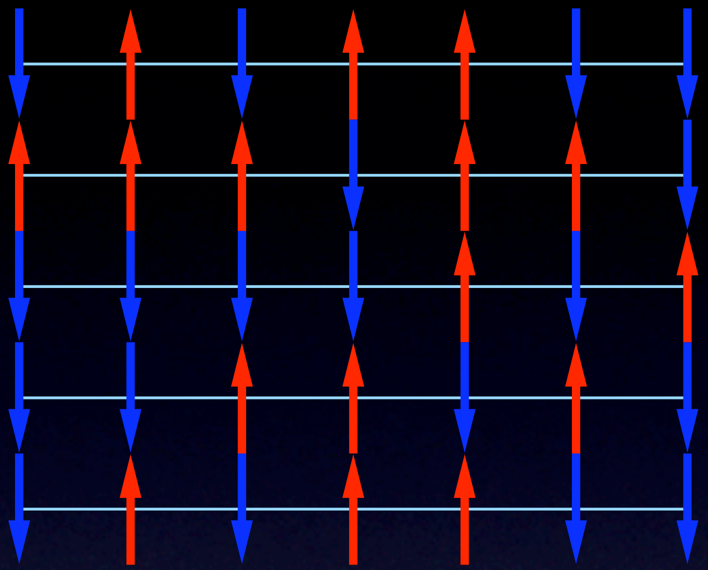
Quantum Monte Carlo



GOOD NEWS:
Very easy implementation
Just add one dimension and use your usual code

BAD NEWS:
New source of finite size effects
(finite-size in the "Trotter" Dimension)
Slow evolution, metastable states

Quantum Monte Carlo

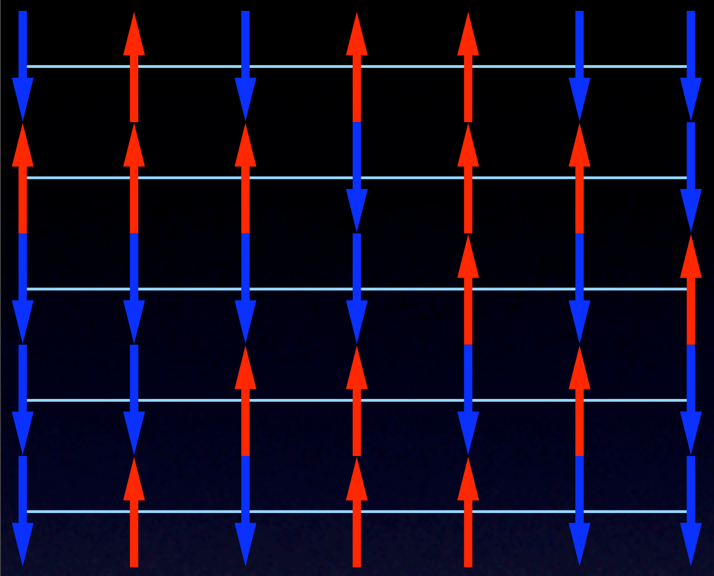


GOOD NEWS:
Very easy implementation
Just add one dimension and use your usual code

BAD NEWS:
New source of finite size effects
(finite-size in the "Trotter" Dimension)
Slow evolution, metastable states

Can we work directly in the infinite N_s limit?

Quantum Monte Carlo



GOOD NEWS:
Very easy implementation
Just add one dimension and use your usual code

BAD NEWS:
New source of finite size effects
(finite-size in the "Trotter" Dimension)
Slow evolution, metastable states

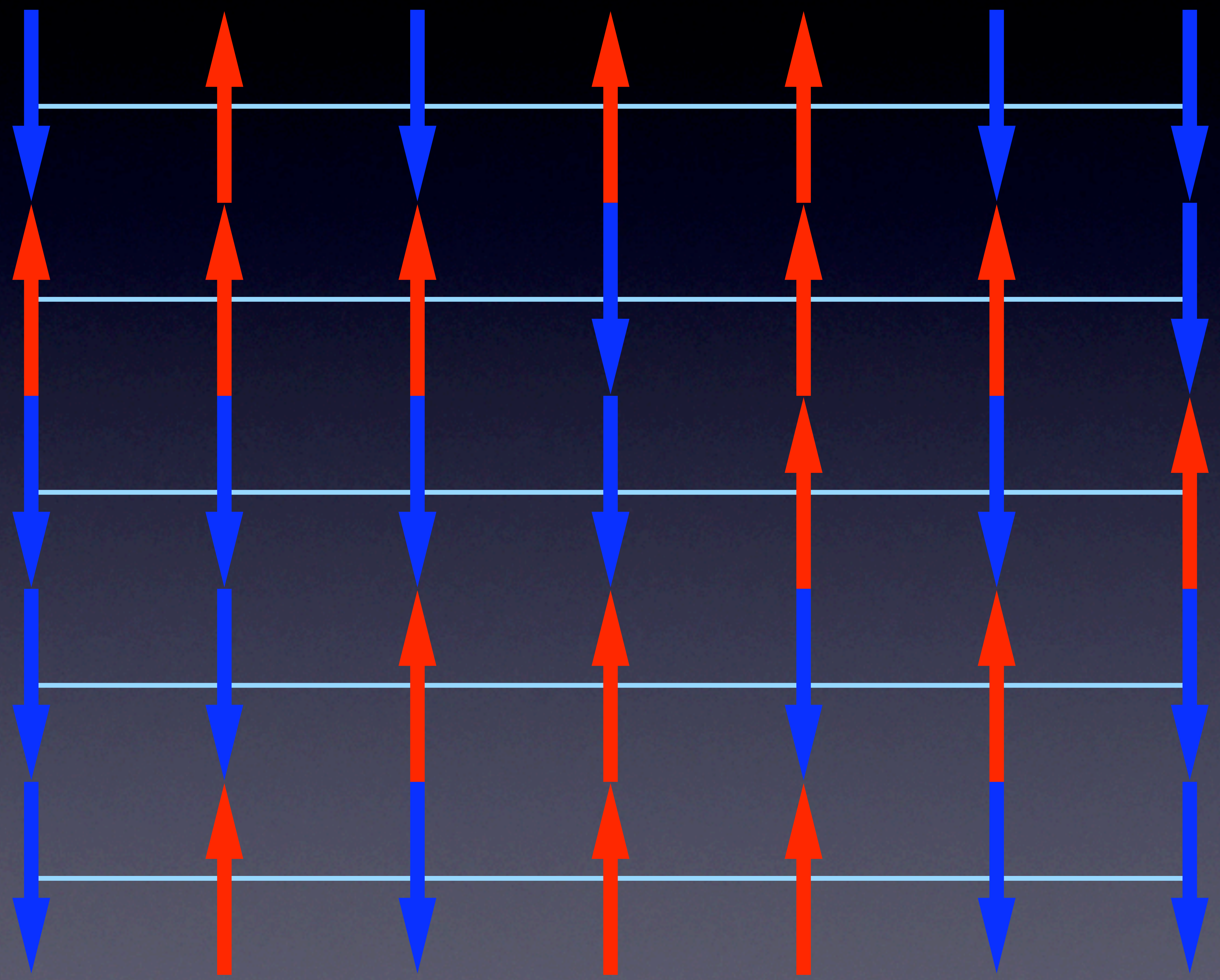
Can we work directly in the infinite N_s limit?

Work directly in the continuous limit :

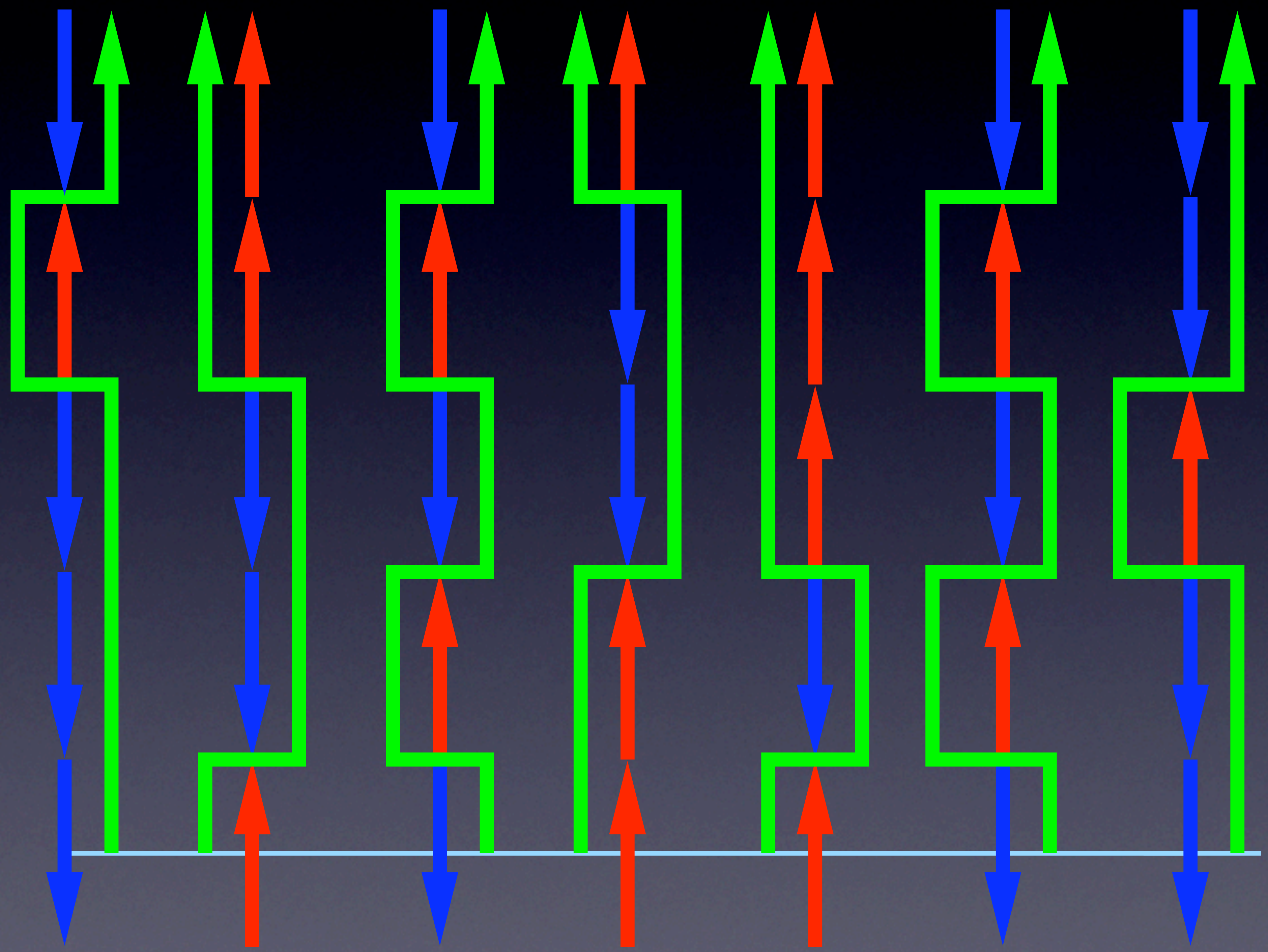
(Loop algorithm: Beard-Wiese 96, Prokof'ev et al. 98, Rieger-Kawashima 1999)

Up to now limited to non-disordered systems

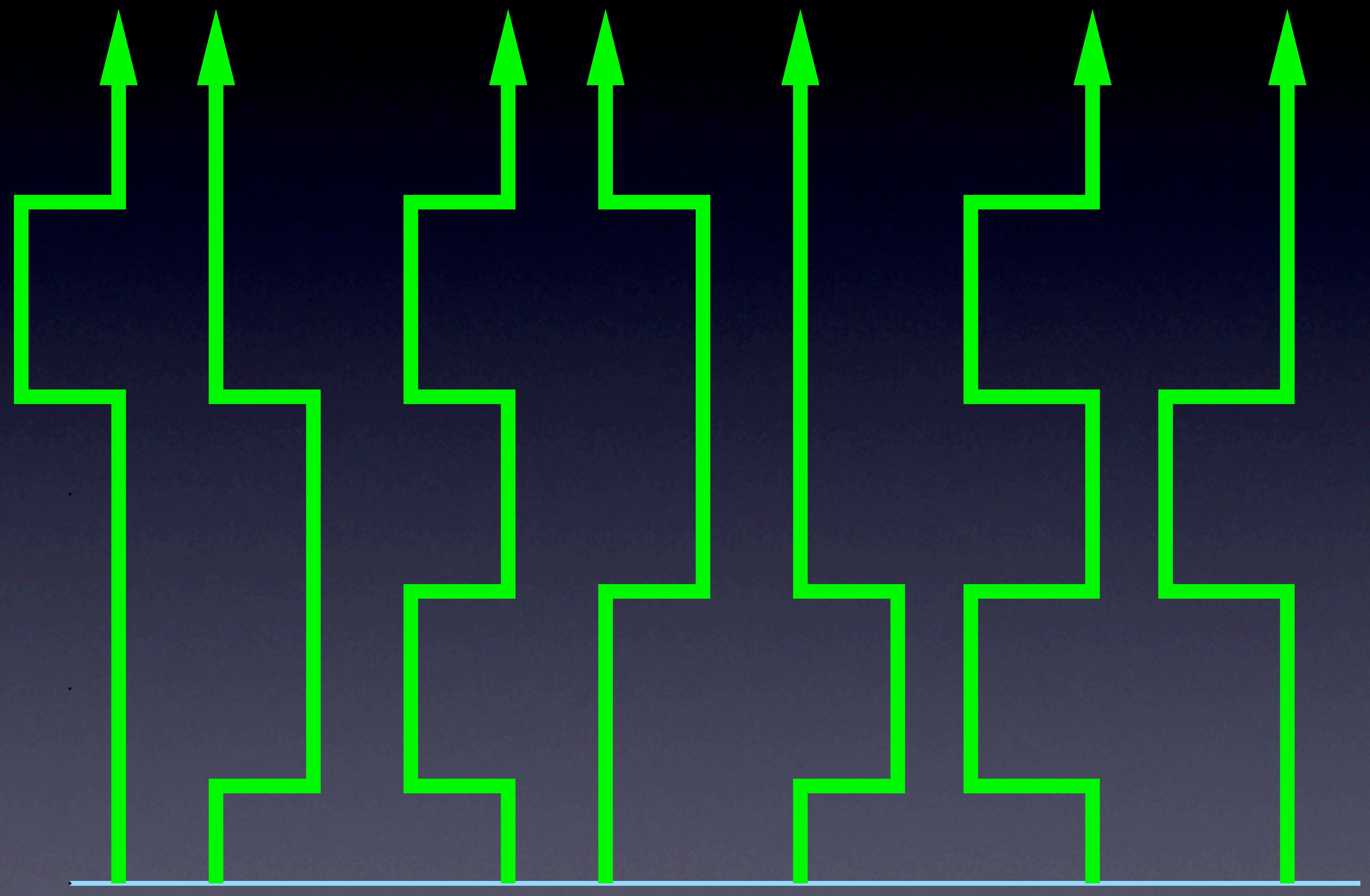
The Continuous limit



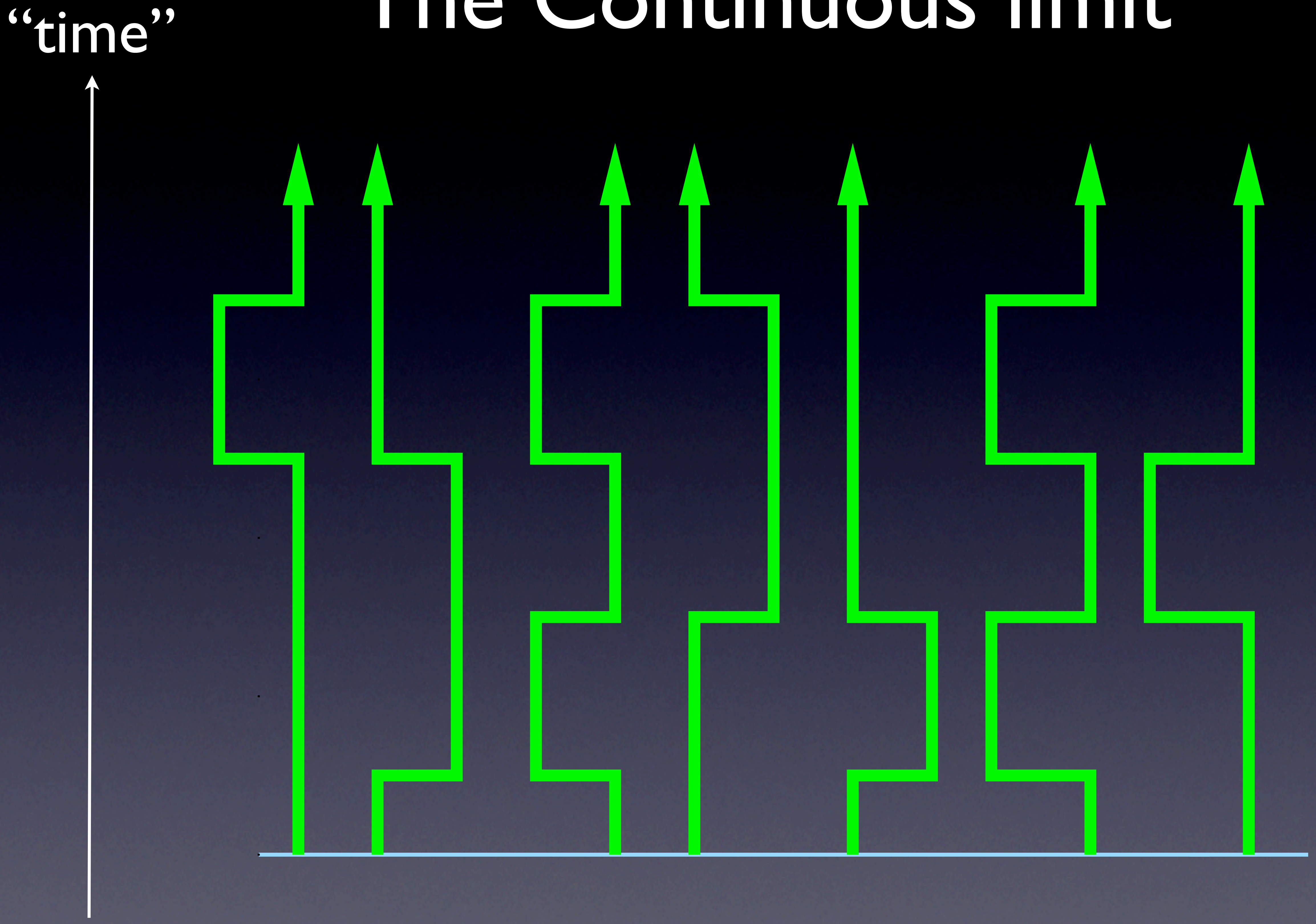
The Continuous limit



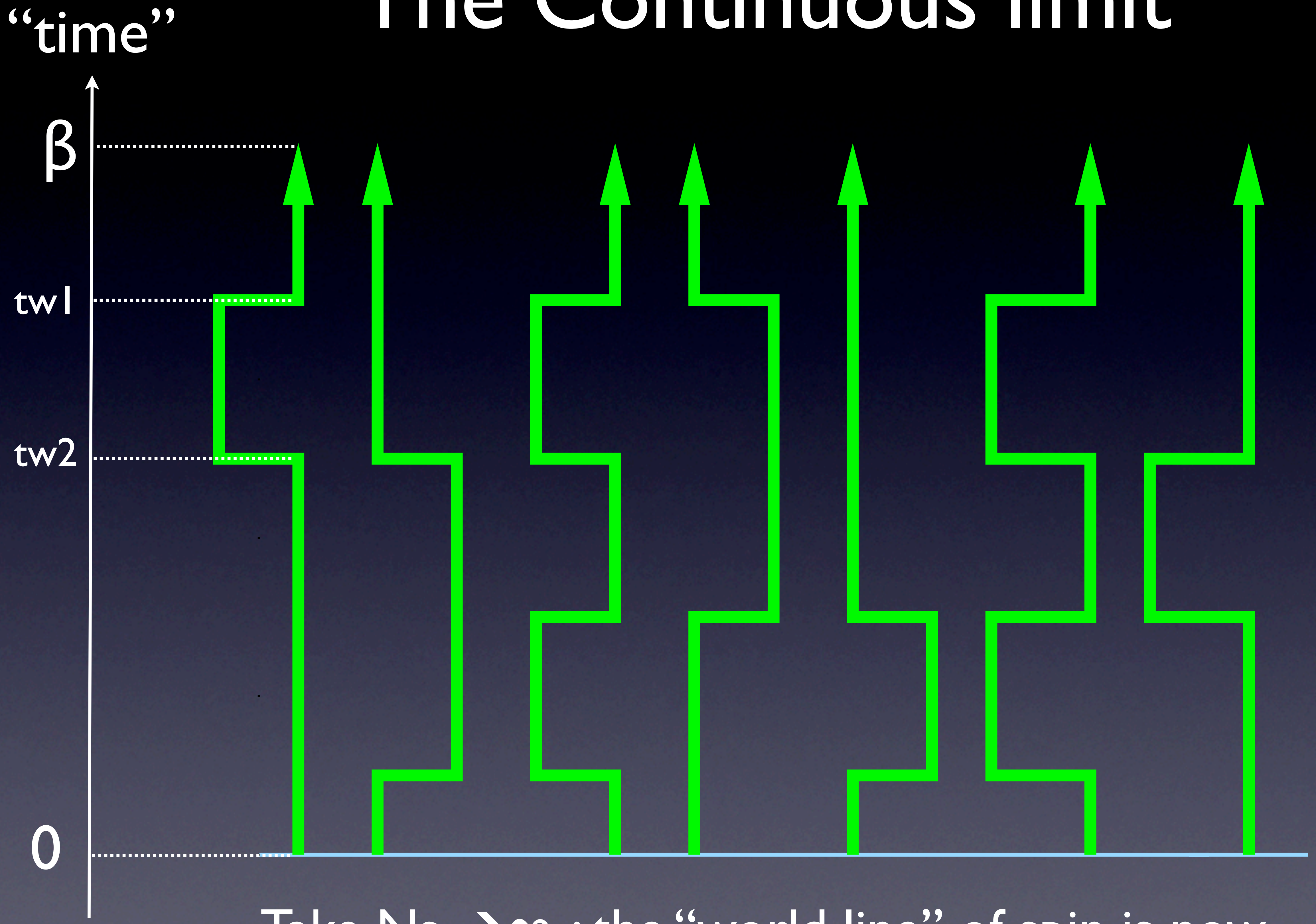
The Continuous limit



The Continuous limit

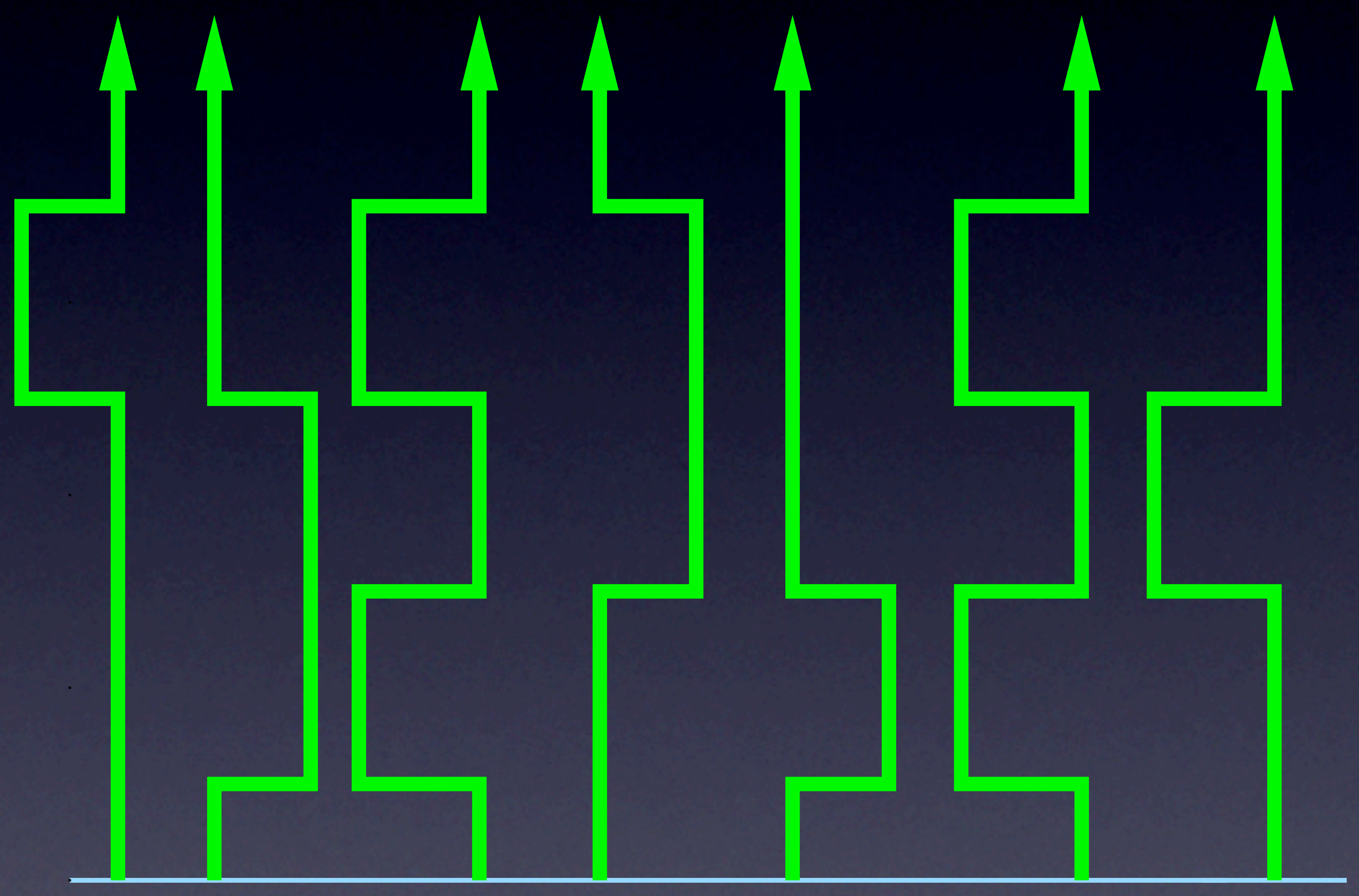


The Continuous limit



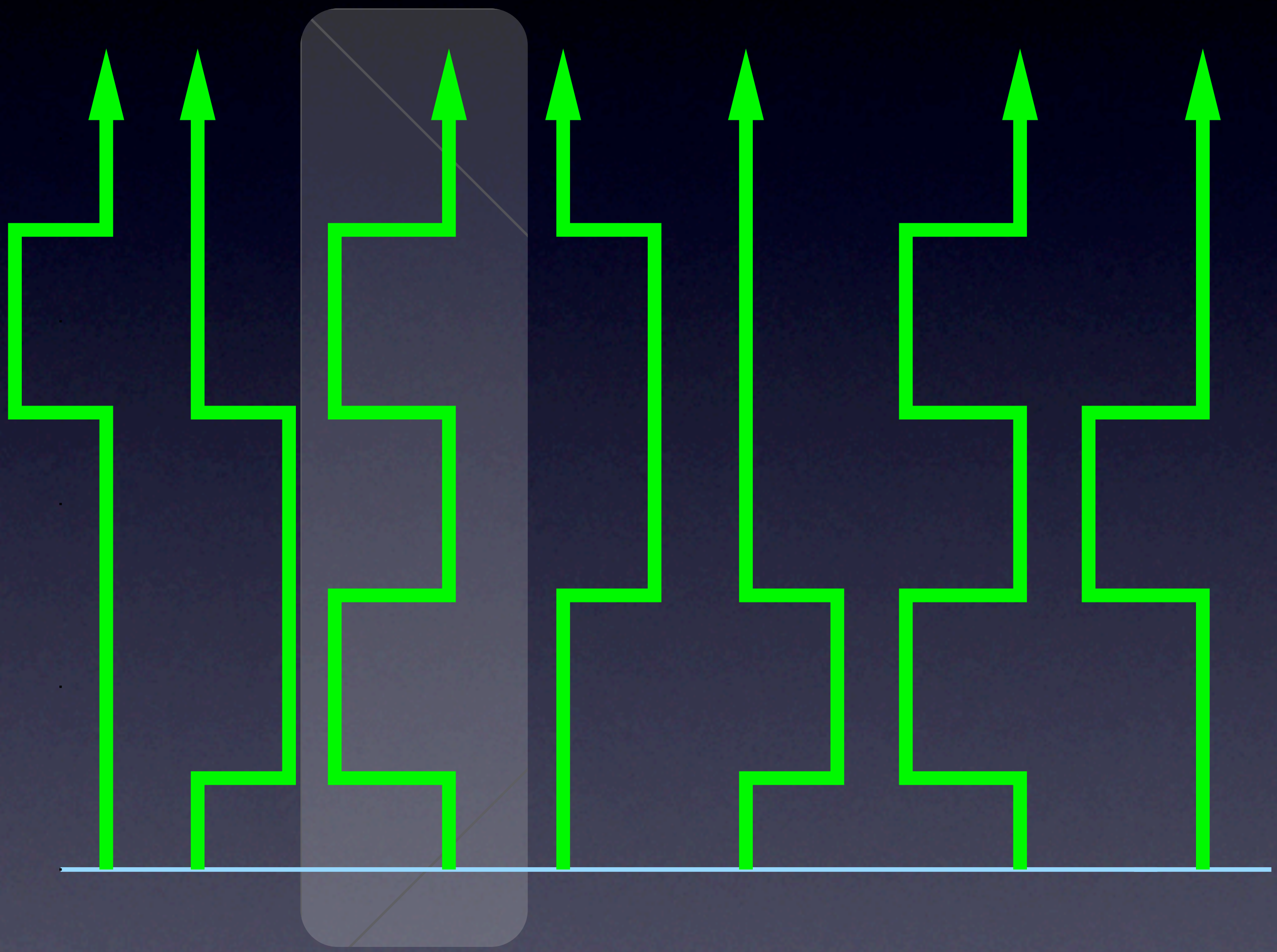
Take Ns → ∞ : the "world line" of spin is now entirely characterized by the set of flipping times

The “Continuous” time Heat Bath

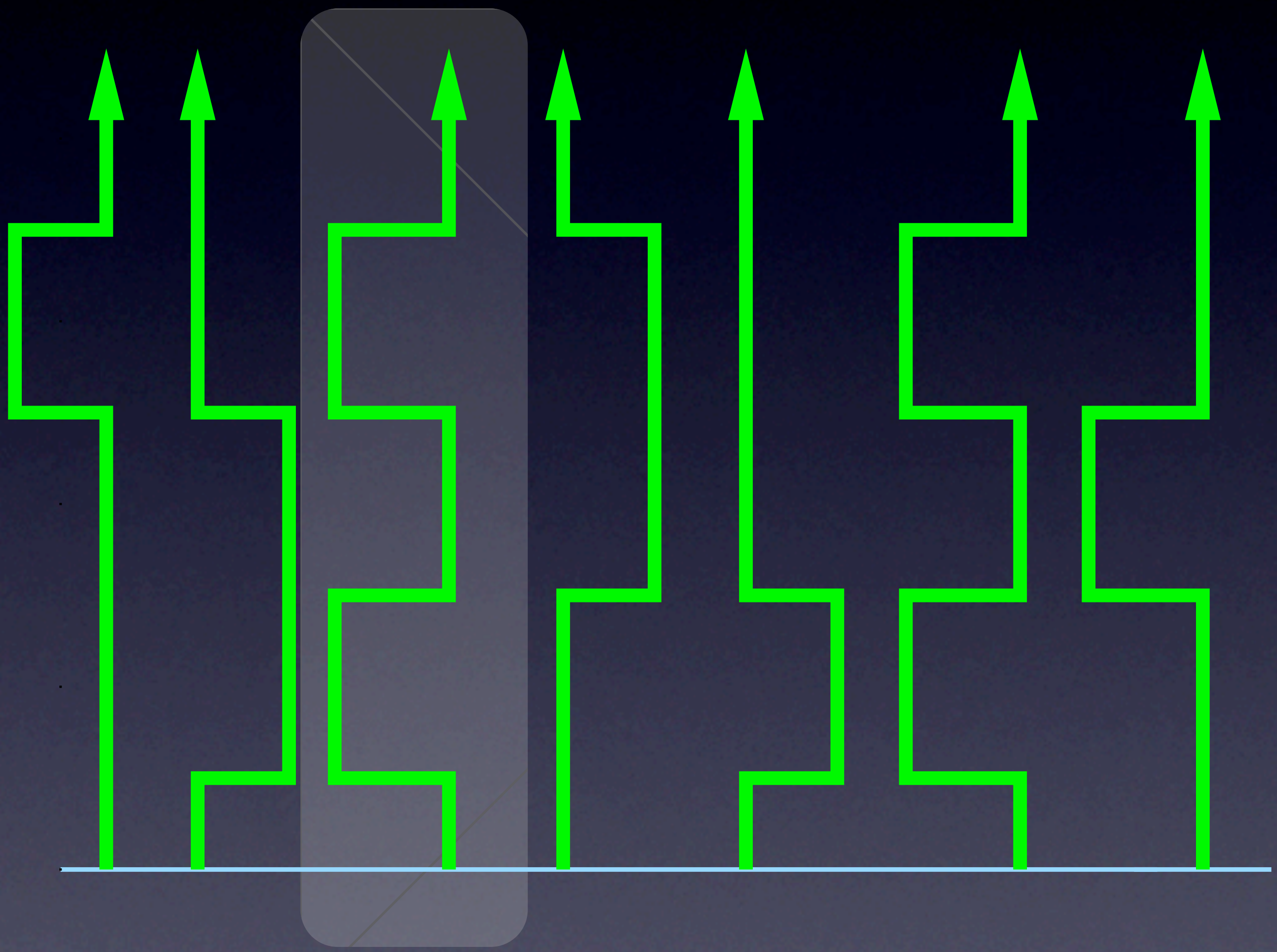


The "Continuous" time Heat Bath

1) Choose a site at random

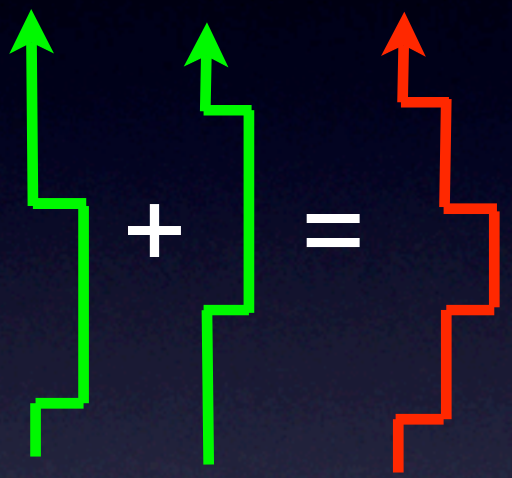


The "Continuous" time Heat Bath

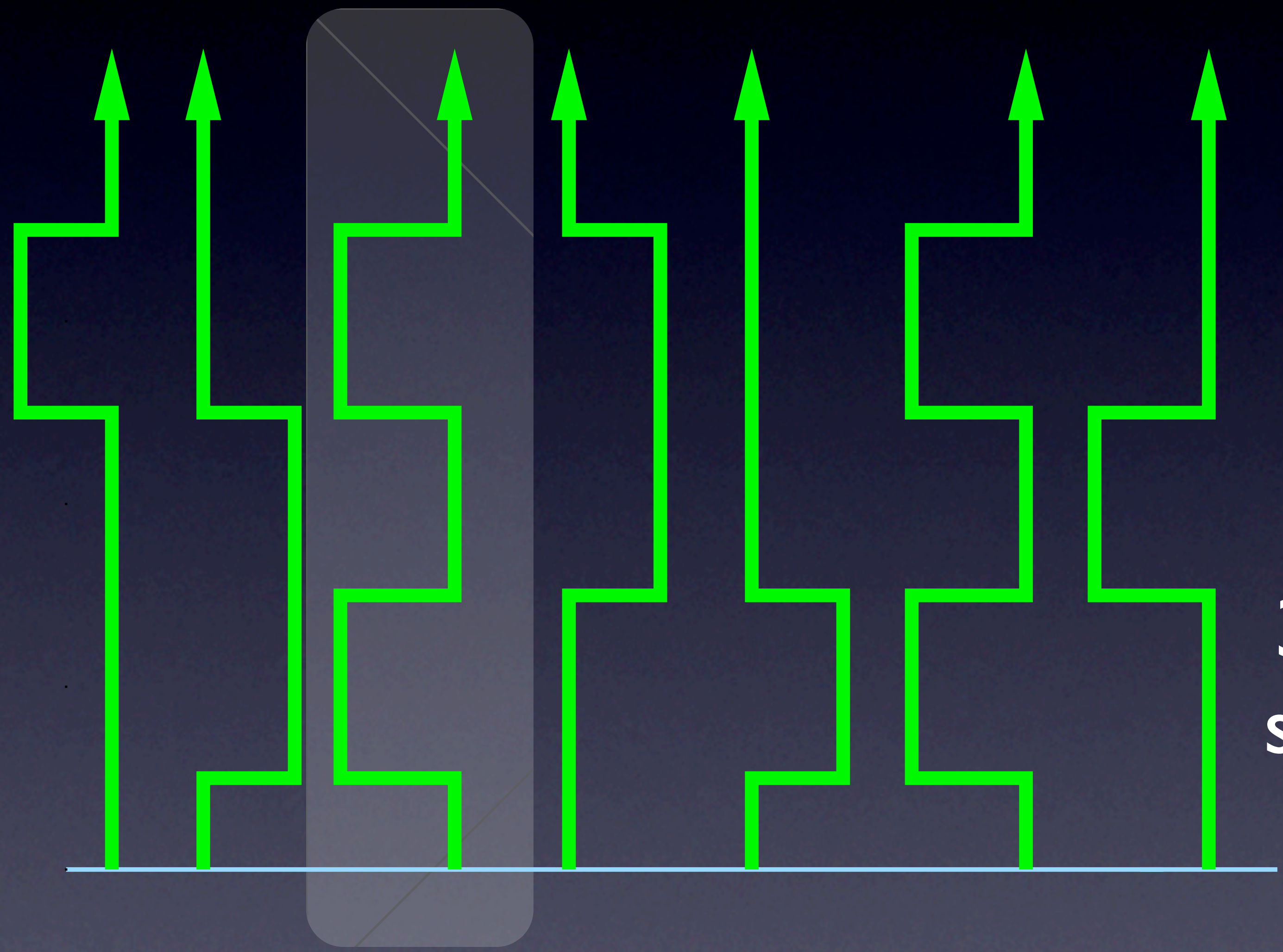


1) Choose a site at random

2) Compute its "local field"

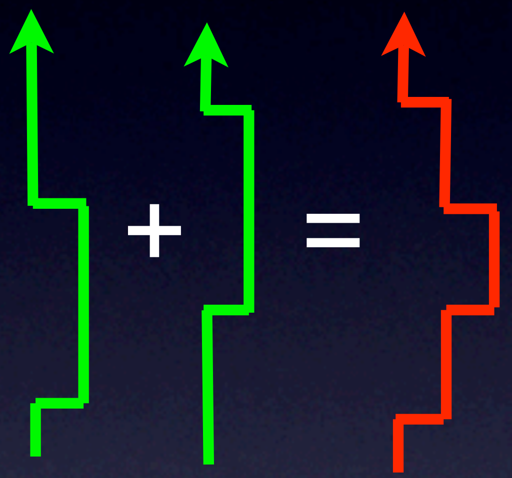


The "Continuous" time Heat Bath



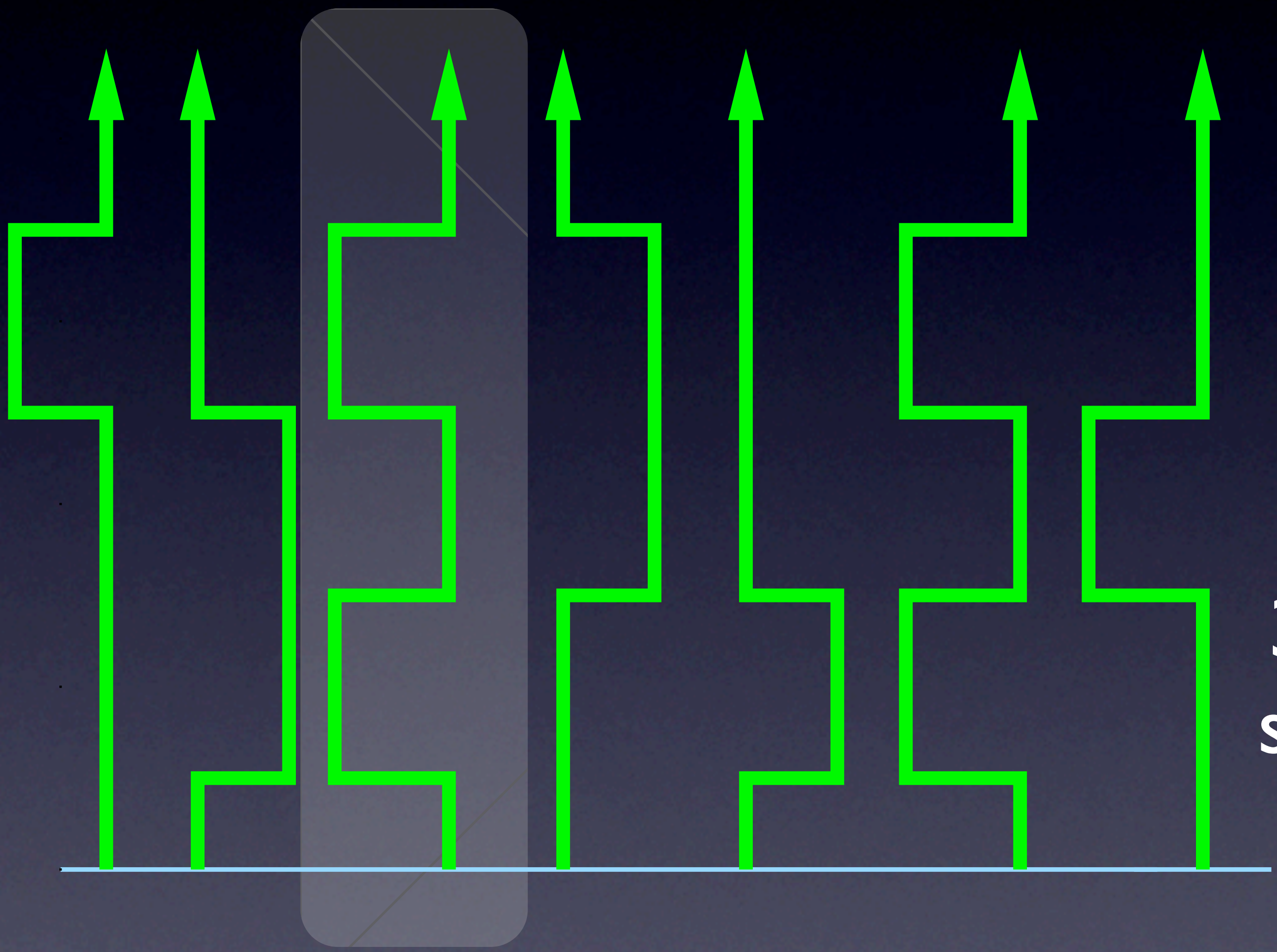
1) Choose a site at random

2) Compute its "local field"



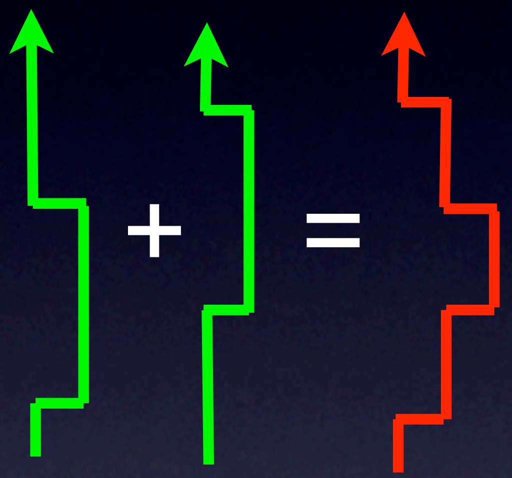
3) Choose the new path of the spin with Boltzmann probability

The "Continuous" time Heat Bath



1) Choose a site at random

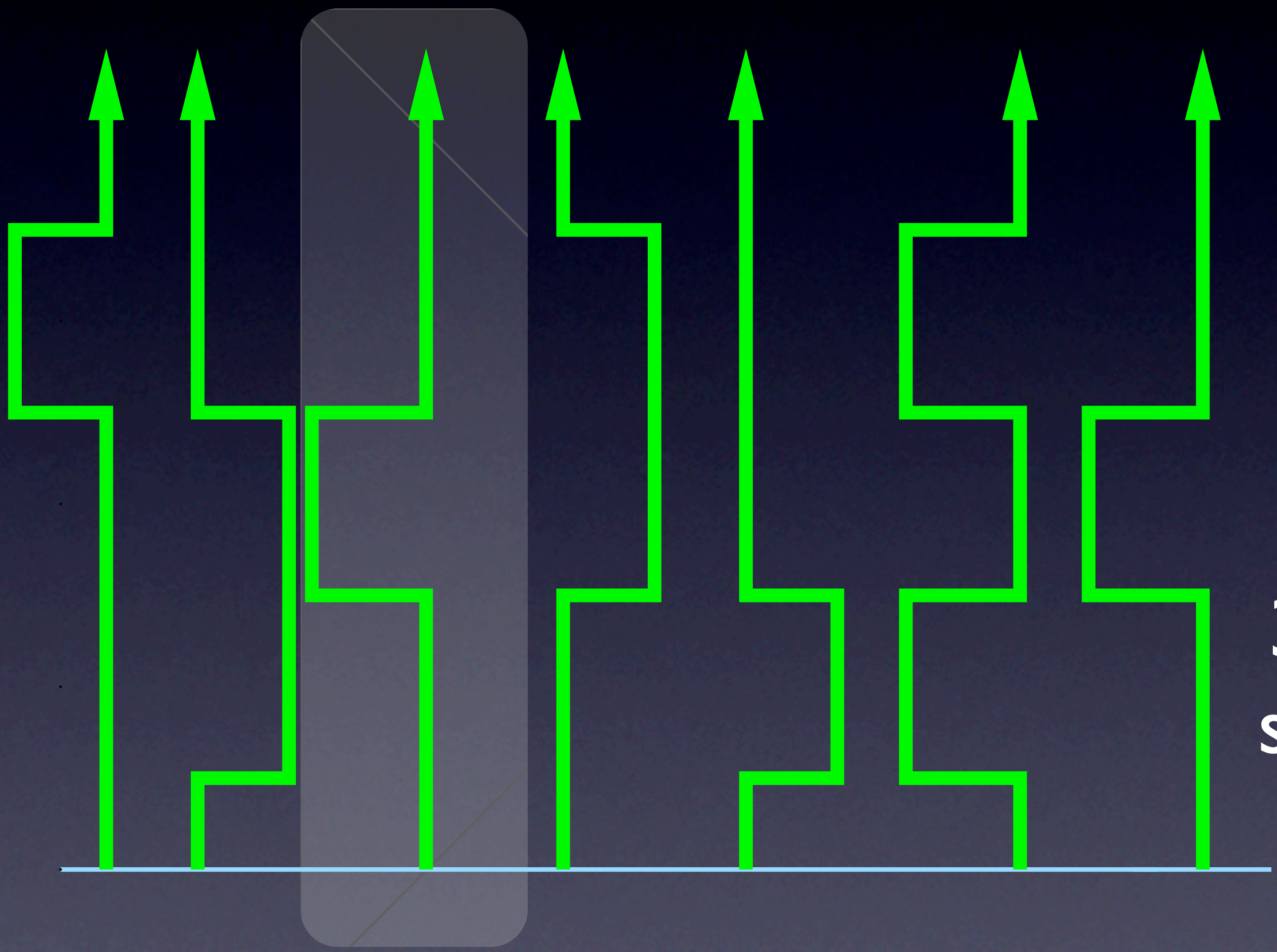
2) Compute its "local field"



3) Choose the new path of the spin with Boltzmann probability

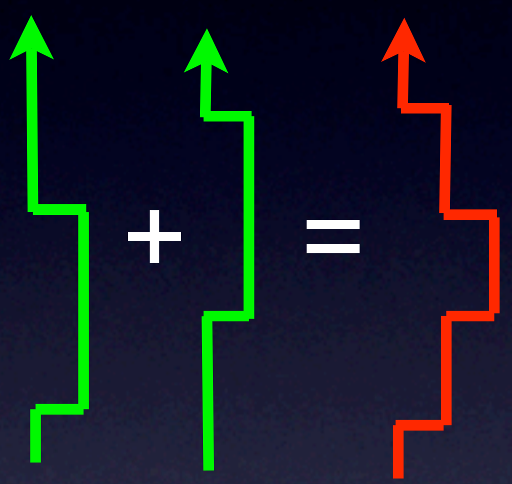
The difficulty is to generate a "world line of spin" given a "world line field"

The "Continuous" time Heat Bath



1) Choose a site at random

2) Compute its "local field"



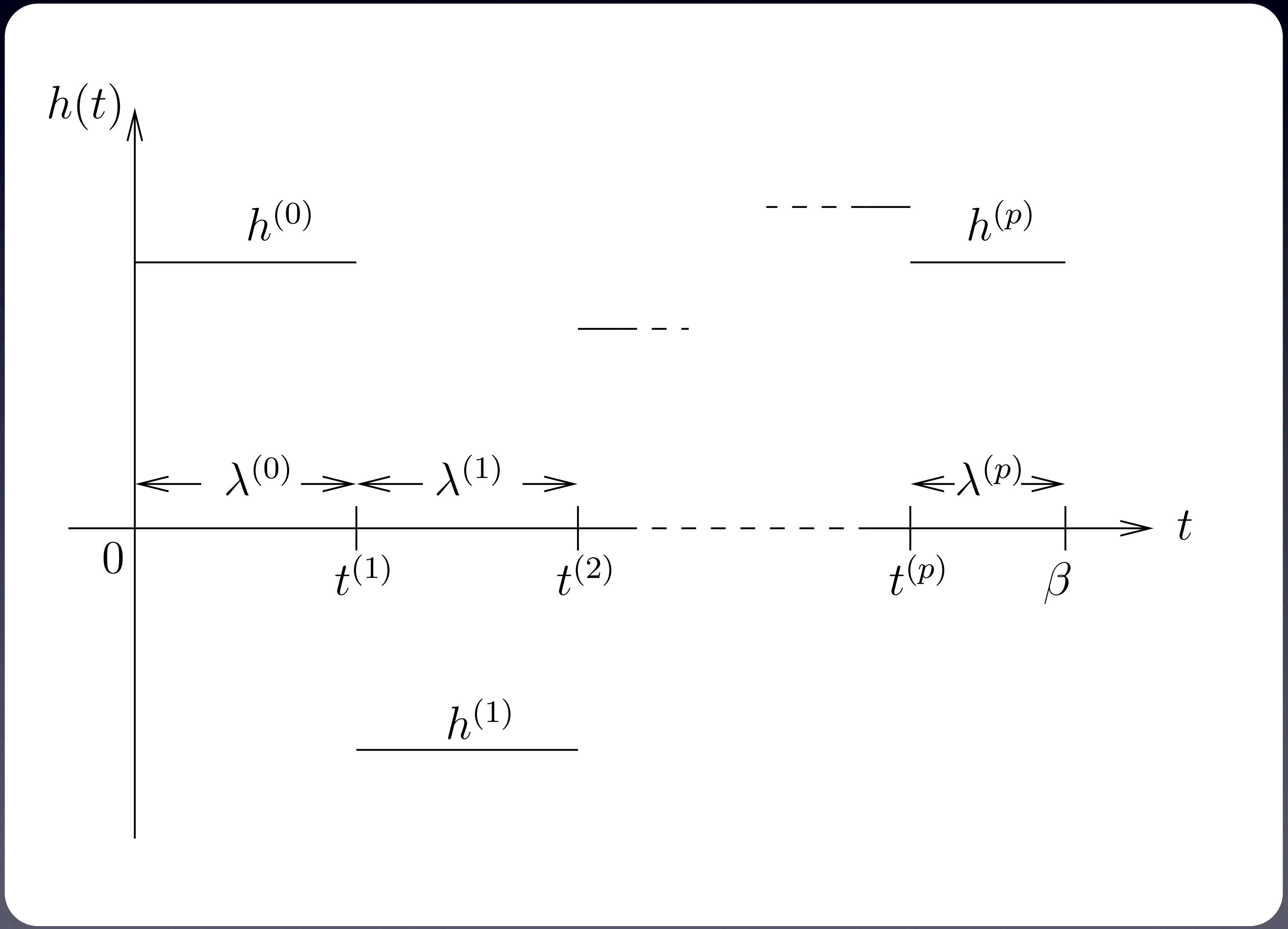
3) Choose the new path of the spin with Boltzmann probability

The difficulty is to generate a "world line of spin" given a "world line field"

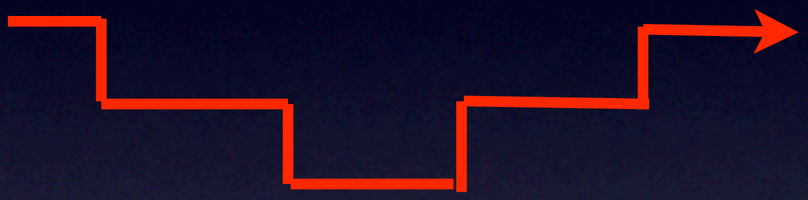
Generating a new spin path in a heat bath way

How to generate the path according to its weight ?

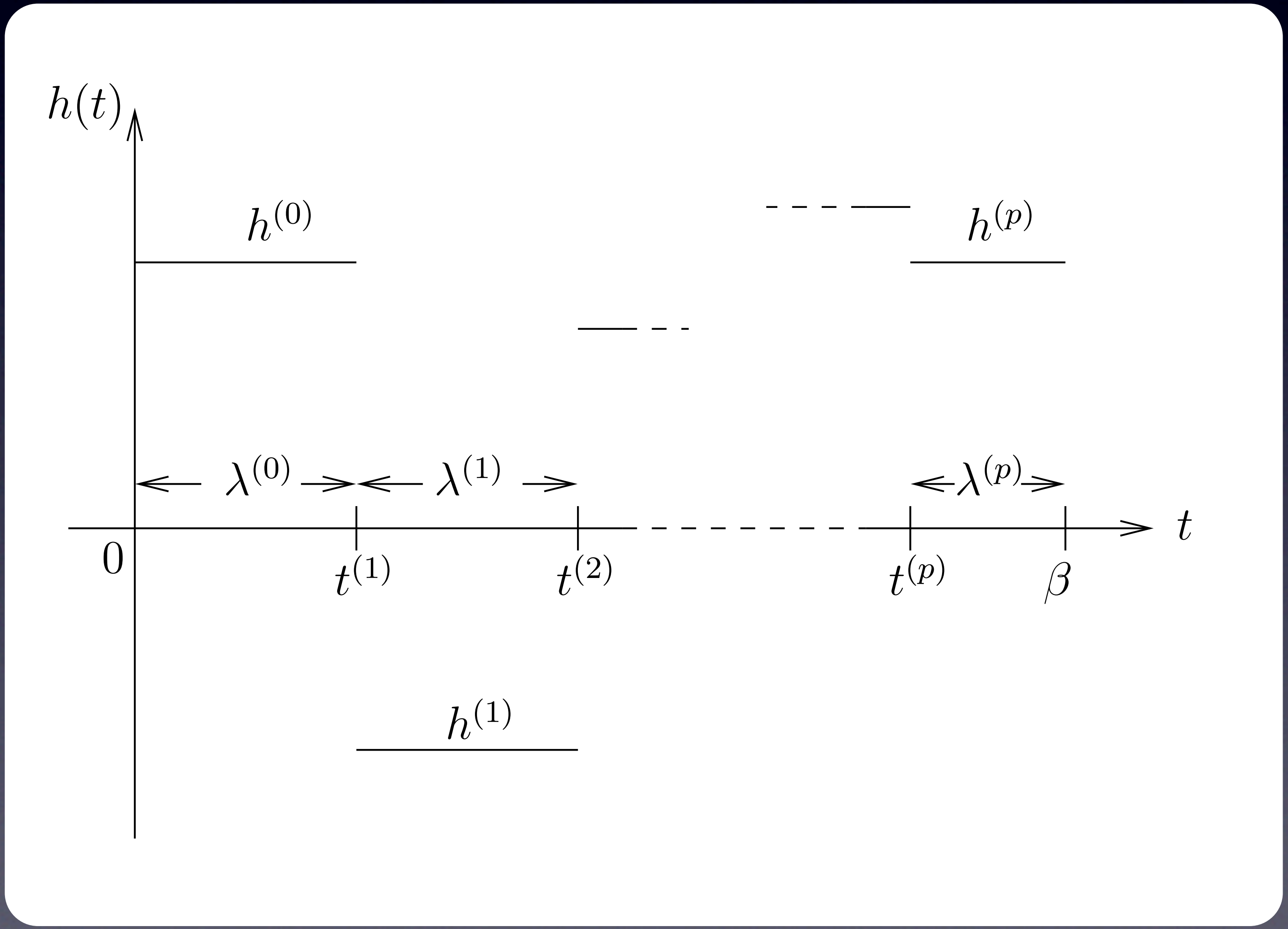
Generating a new spin path in a heat bath way



How to generate the path according to its weight ?



Generating a new spin path in a heat bath way



How to generate the path according to its weight ?

1) How to generate a path in a constant field ?

2) How to generate the path in a piecewise constant field?

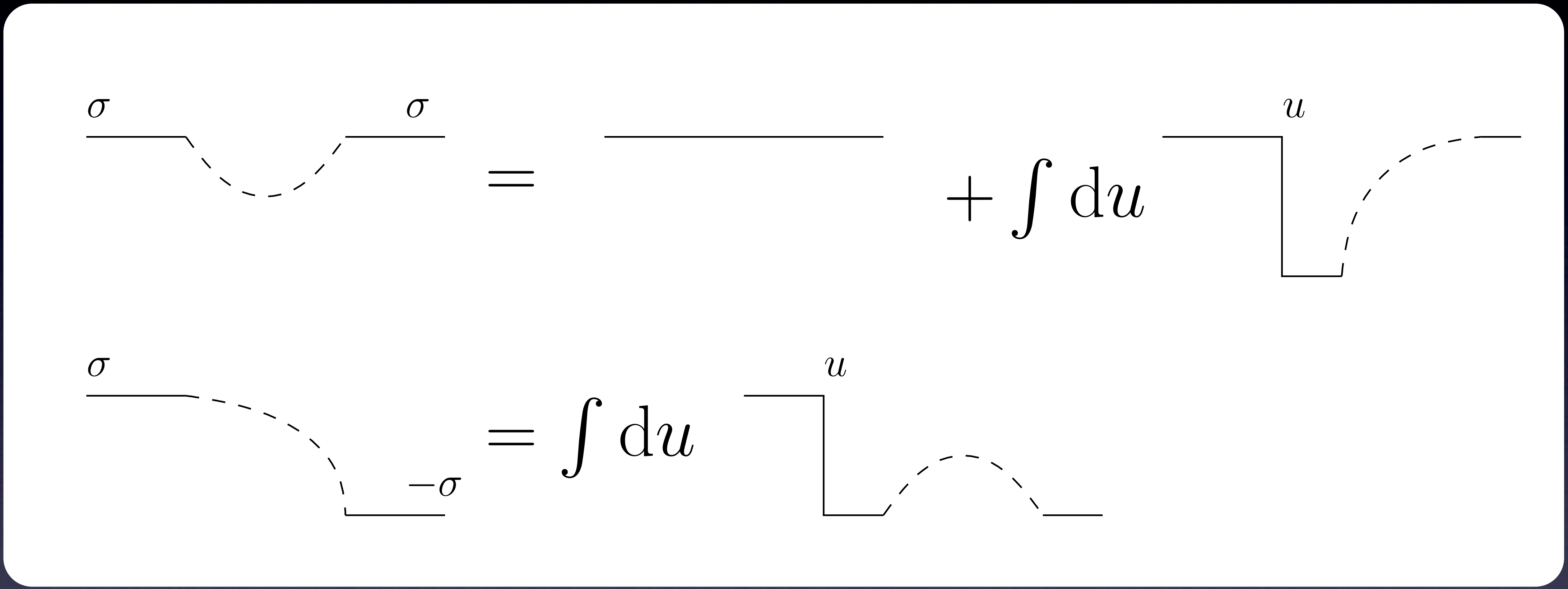
Generating a path in a constant field

Define (and compute) the propagators in constant field h for a time λ:

$$e^{\lambda(h\sigma_z + \lambda\Gamma\sigma_x)} = \begin{pmatrix} W_{u,u} & W_{u,d} \\ W_{d,u} & W_{d,d} \end{pmatrix}$$

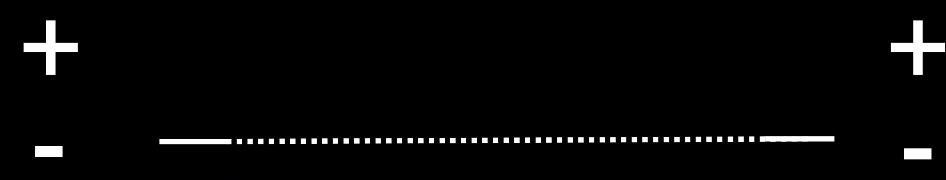
$$W(s \to s', h, \lambda) = \begin{cases} \cosh(\lambda\sqrt{\Gamma^2 + h^2}) + s\frac{h}{\sqrt{\Gamma^2+h^2}}\sinh(\lambda\sqrt{\Gamma^2+h^2}) & \text{if } s = s' \\ \frac{\Gamma}{\sqrt{\Gamma^2+h^2}}\sinh(\lambda\sqrt{\Gamma^2+h^2}) & \text{if } s = -s' \end{cases}$$

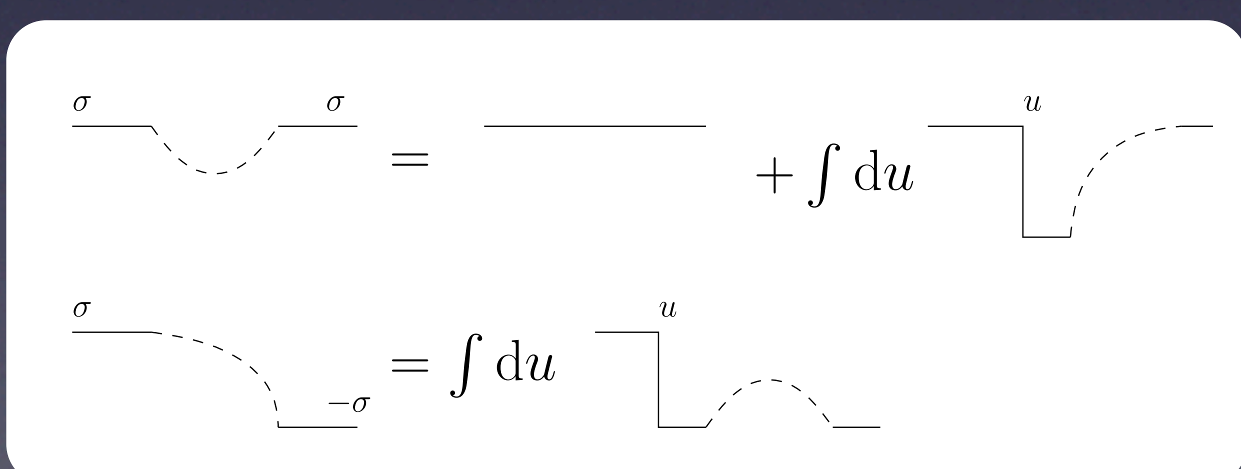
A useful recursion



$$
\begin{aligned}
W(s \to s, h, \lambda) &= e^{sh\lambda} + \Gamma \int_0^{\lambda} du\; e^{shu}\; W(-s \to s, h, \lambda - u)\;, \\
W(s \to -s, h, \lambda) &= \Gamma \int_0^{\lambda} du\; e^{shu}\; W(-s \to -s, h, \lambda - u)\;.
\end{aligned}
$$

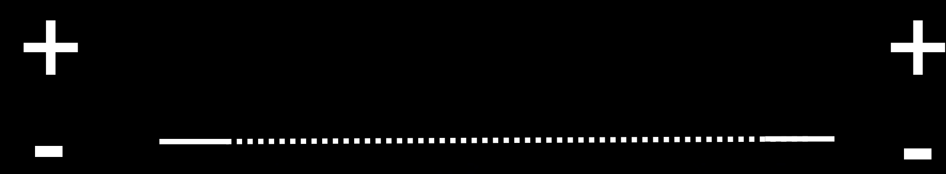
A simple recursive algoritmh





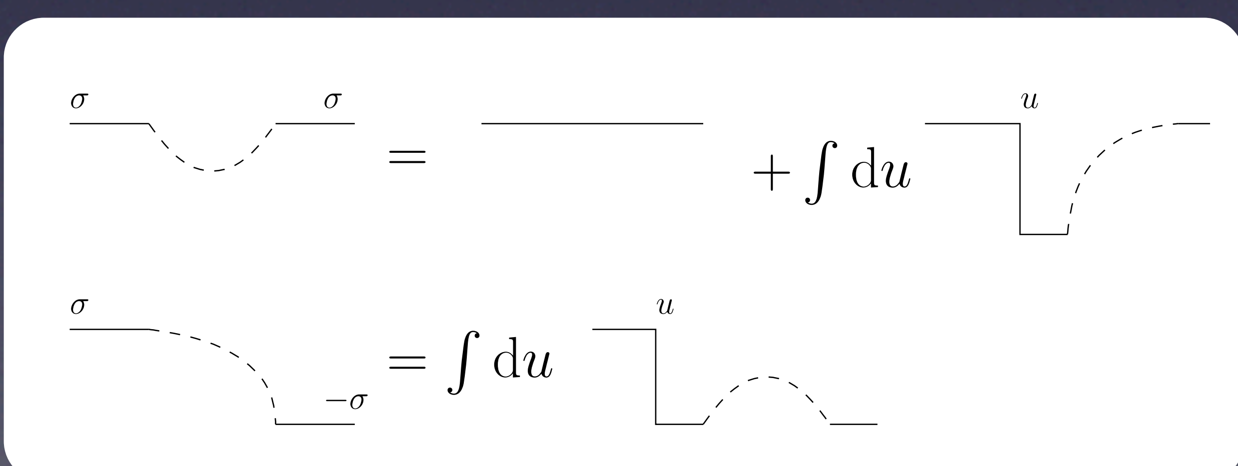
$$W(s \to s, h, \lambda) \quad = \quad e^{sh\lambda} + \Gamma \int_0^{\lambda} du \; e^{shu} \; W(-s \to s, h, \lambda - u) \; ,$$

A simple recursive algoritmh



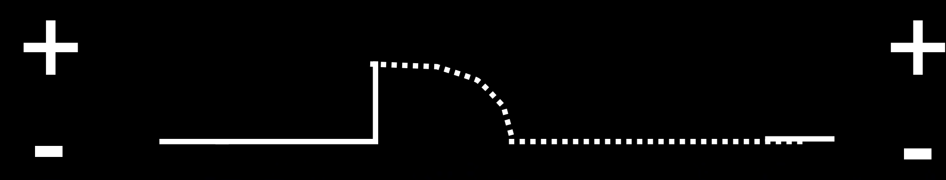
if s = s′ : – with probability $e^{sh\lambda}$ /W (s → s, h, λ), set σ(t) = σ on the whole time interval

– otherwise, draw a random variable u ∈ [0, λ] with density proportional to e^{shu} W (−s → s, h, λ – u) and set s(t) = σ up to time u



$$W(s \to s, h, \lambda) \quad = \quad e^{sh\lambda} + \Gamma \int_0^{\lambda} du\; e^{shu}\; W(-s \to s, h, \lambda - u)\;,$$

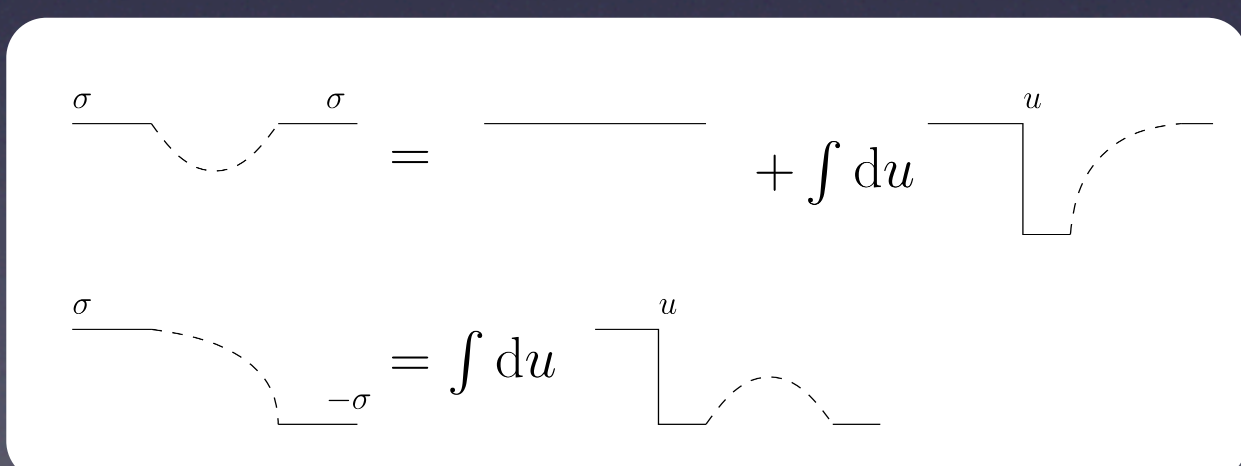
A simple recursive algoritmh



if s = s′ :

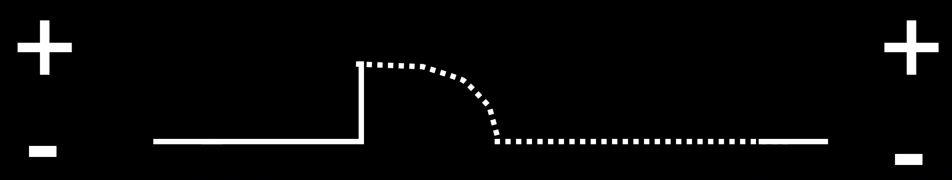
– with probability $e^{sh\lambda} / W(s \to s, h, \lambda)$, set $\sigma(t) = \sigma$ on the whole time interval

– otherwise, draw a random variable $u \in [0, \lambda]$ with density proportional to $e^{shu}\, W(-s \to s, h, \lambda - u)$ and set $s(t) = \sigma$ up to time u



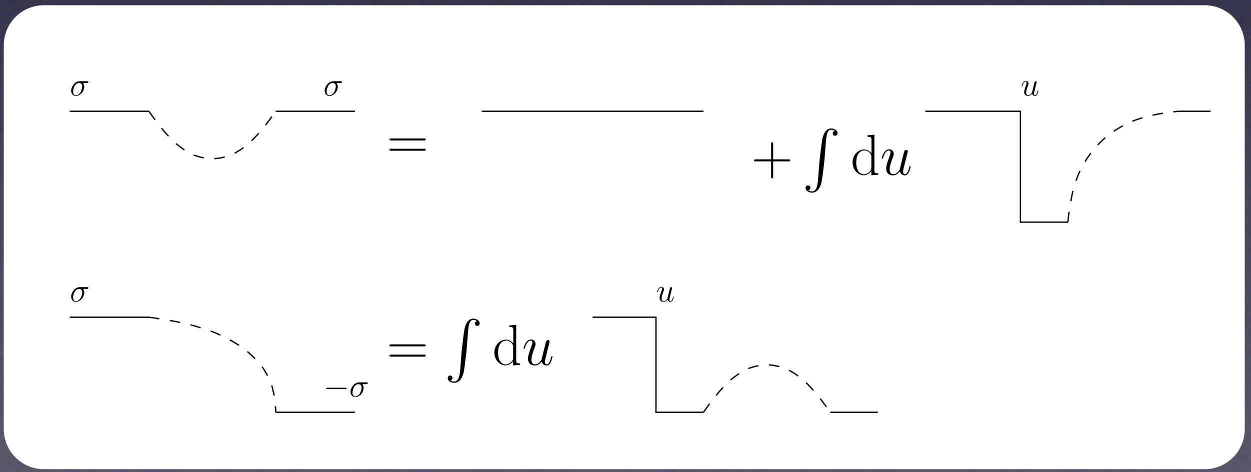
$$W(s \to s, h, \lambda) \;=\; e^{sh\lambda} + \Gamma \int_0^{\lambda} du\; e^{shu}\, W(-s \to s, h, \lambda - u)\;,$$

A simple recursive algoritmh



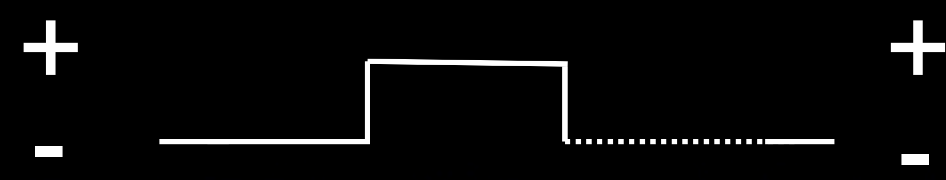
If s = -s′ : – draw a random number with density proportional to $e^{shu}\, W(-s \to -s, h, \lambda - u)$

- set $\sigma(t) = \sigma$ up to time u
- call the previous procedure to generate the remaining trajectory



$$W(s \to -s, h, \lambda) \;=\; \Gamma \int_0^{\lambda} du\; e^{shu}\; W(-s \to -s, h, \lambda - u)\ .$$

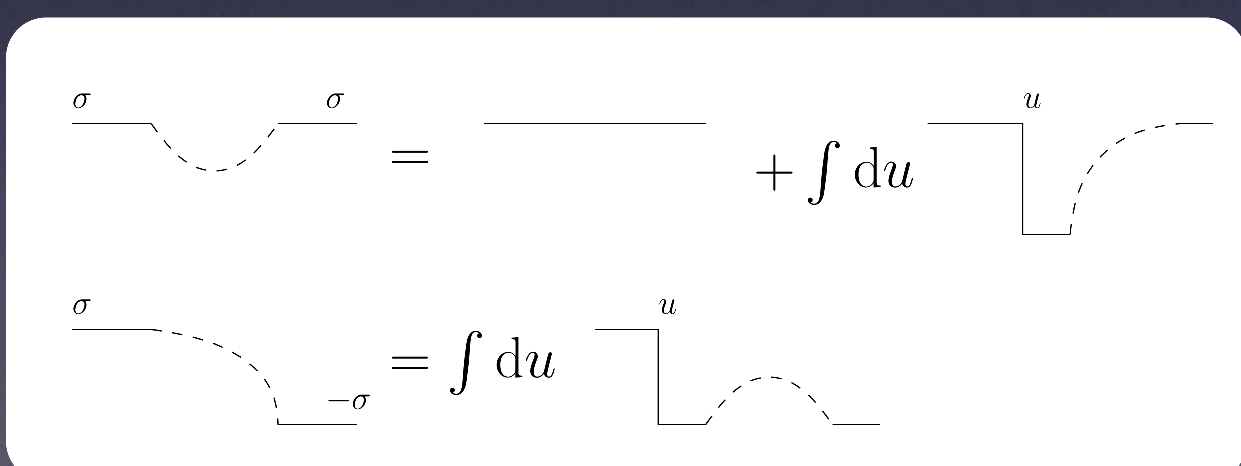
A simple recursive algoritmh



If $s = -s'$: – draw a random number with density proportional to $e^{shu}\, W(-s \to -s, h, \lambda - u)$

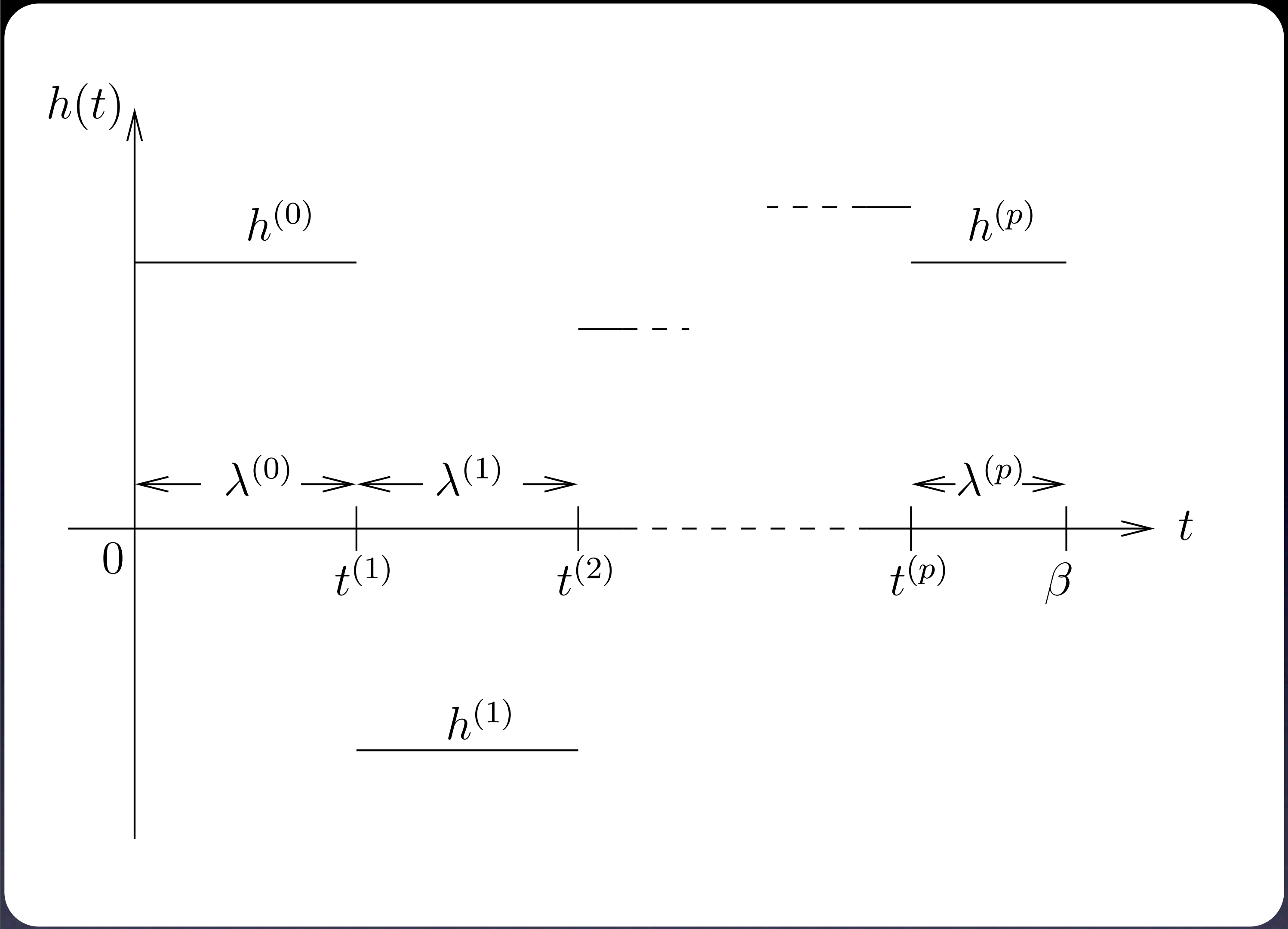
– set $\sigma(t) = \sigma$ up to time u

– call the previous procedure to generate the remaining trajectory



$$W(s \to -s, h, \lambda) \;=\; \Gamma \int_0^{\lambda} du\; e^{shu}\; W(-s \to -s, h, \lambda - u)\;.$$

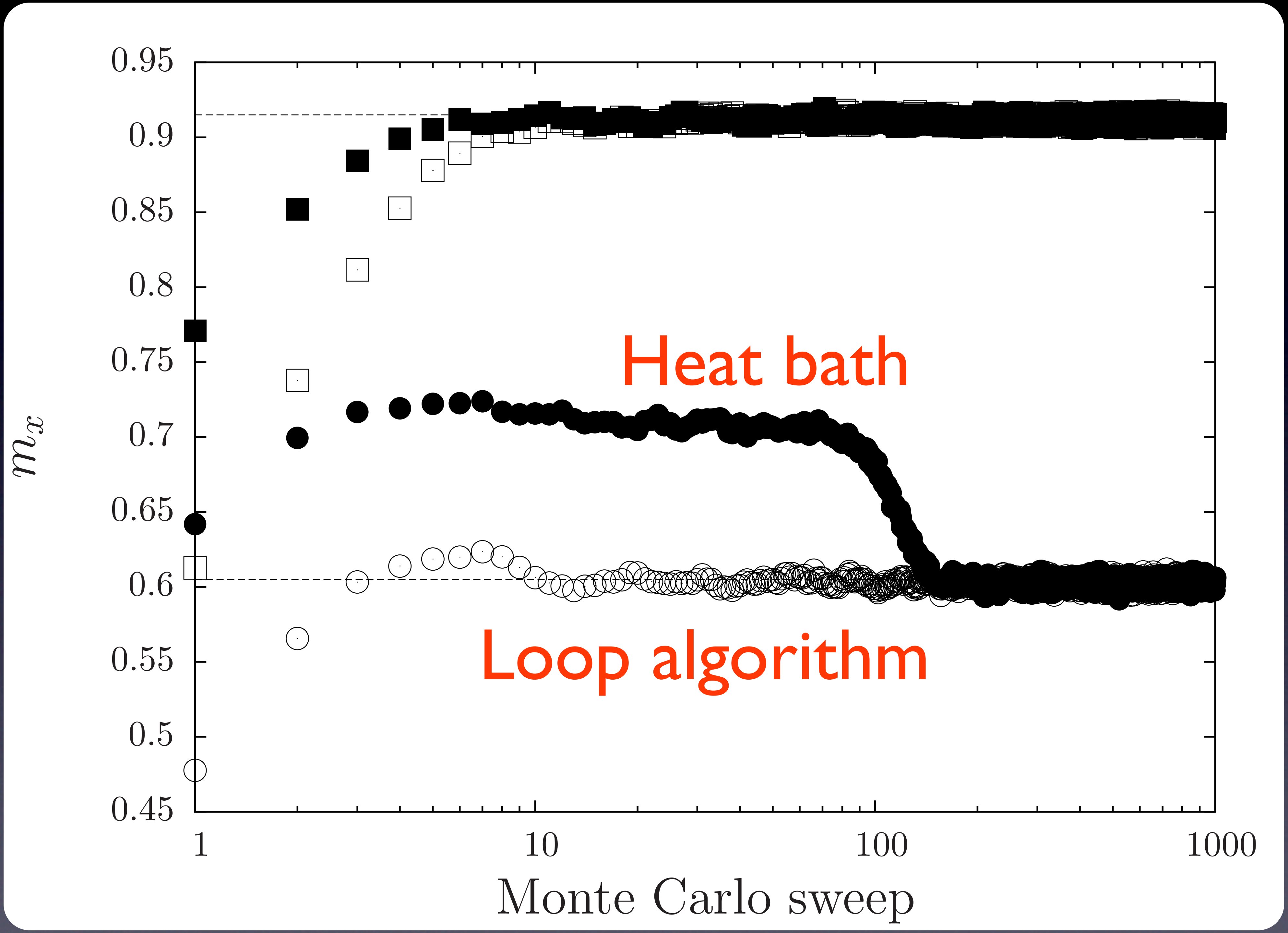
Generating a path in a constant piecewise field



We need to know the spin orientation at time t(1),t(2) ... in order to apply the "constant field algorithm"

$$P(s_1, \cdots, s_p|\mathbf{h}) = \prod_{i=0}^{p} W(s_i \rightarrow s_{i+1}, h^{(i)}, \lambda^{(i)})$$

Some results



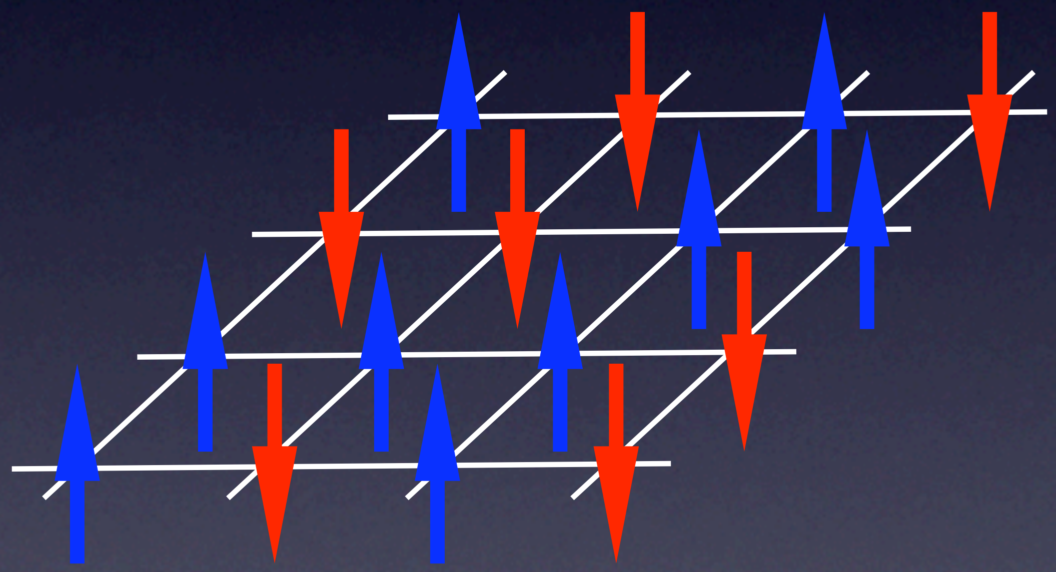
Comparison with the best available algorithm
(Loop Algorithm, Rieger-Kawashima 98')
on a regular random graph

Overview

- Heat bath for classical and quantum spins
- Cavity Method for classical and quantum spins
- Concusions and perspectives

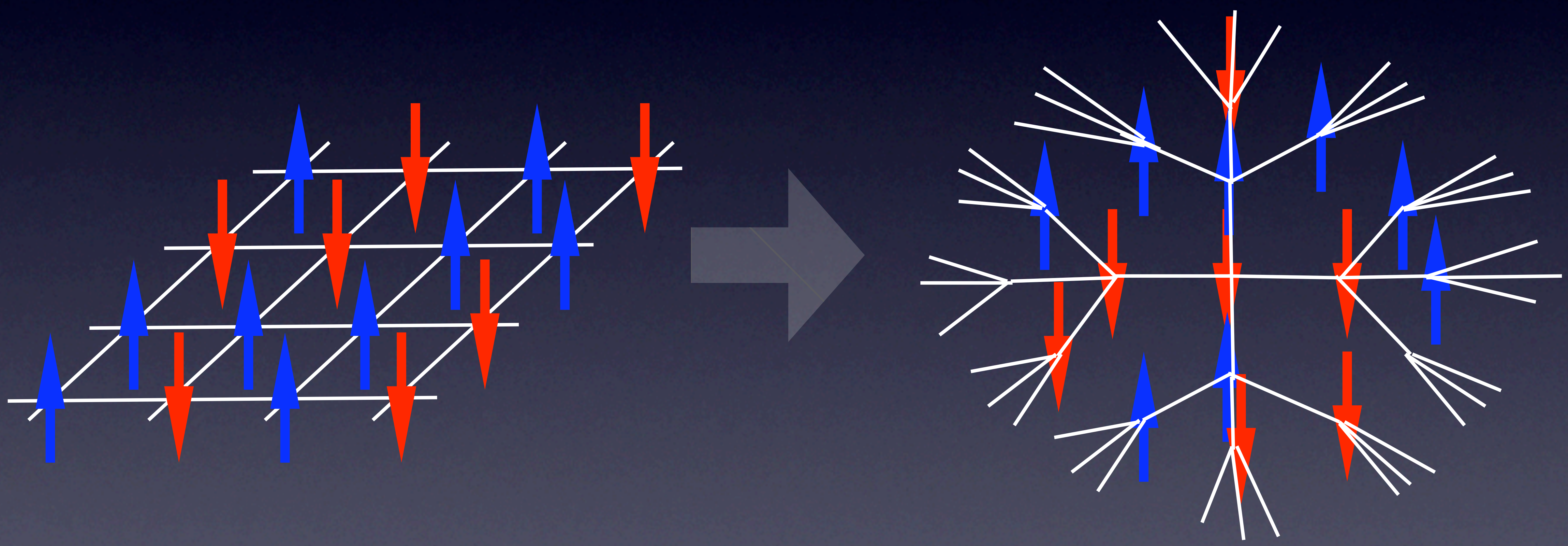
Bethe-Peierls Approximation

(Replica-Symmetric cavity method)



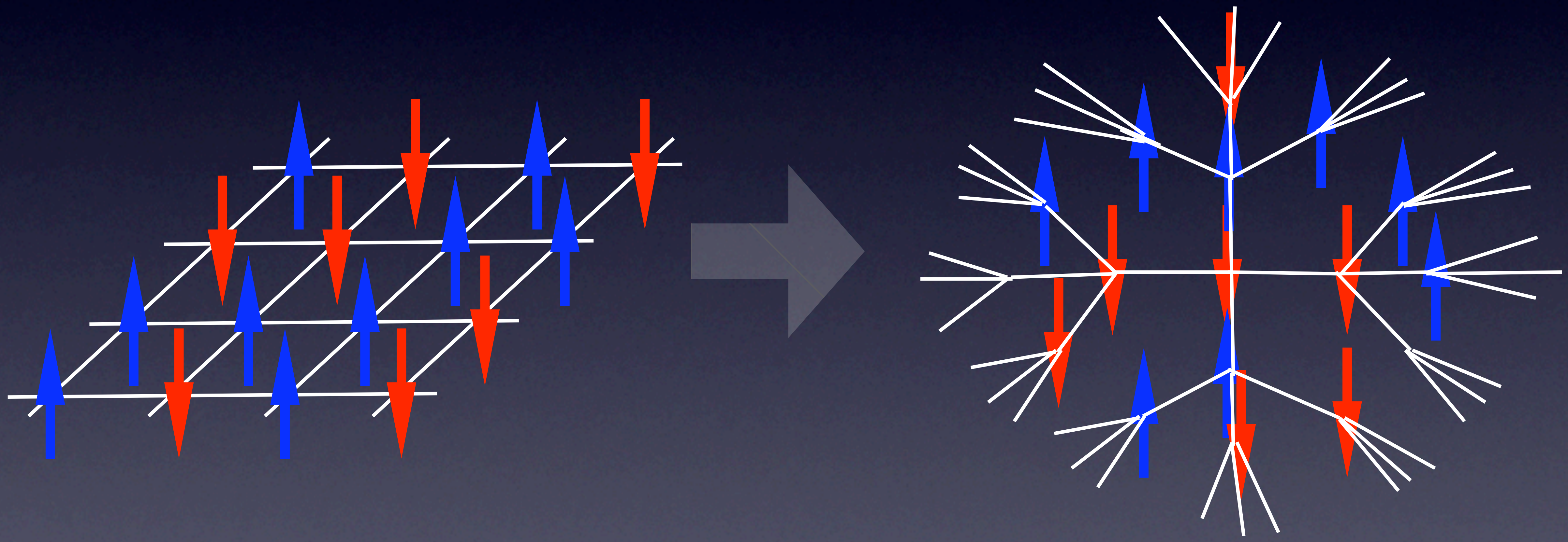
Bethe-Peierls Approximation

(Replica-Symmetric cavity method)



Bethe-Peierls Approximation

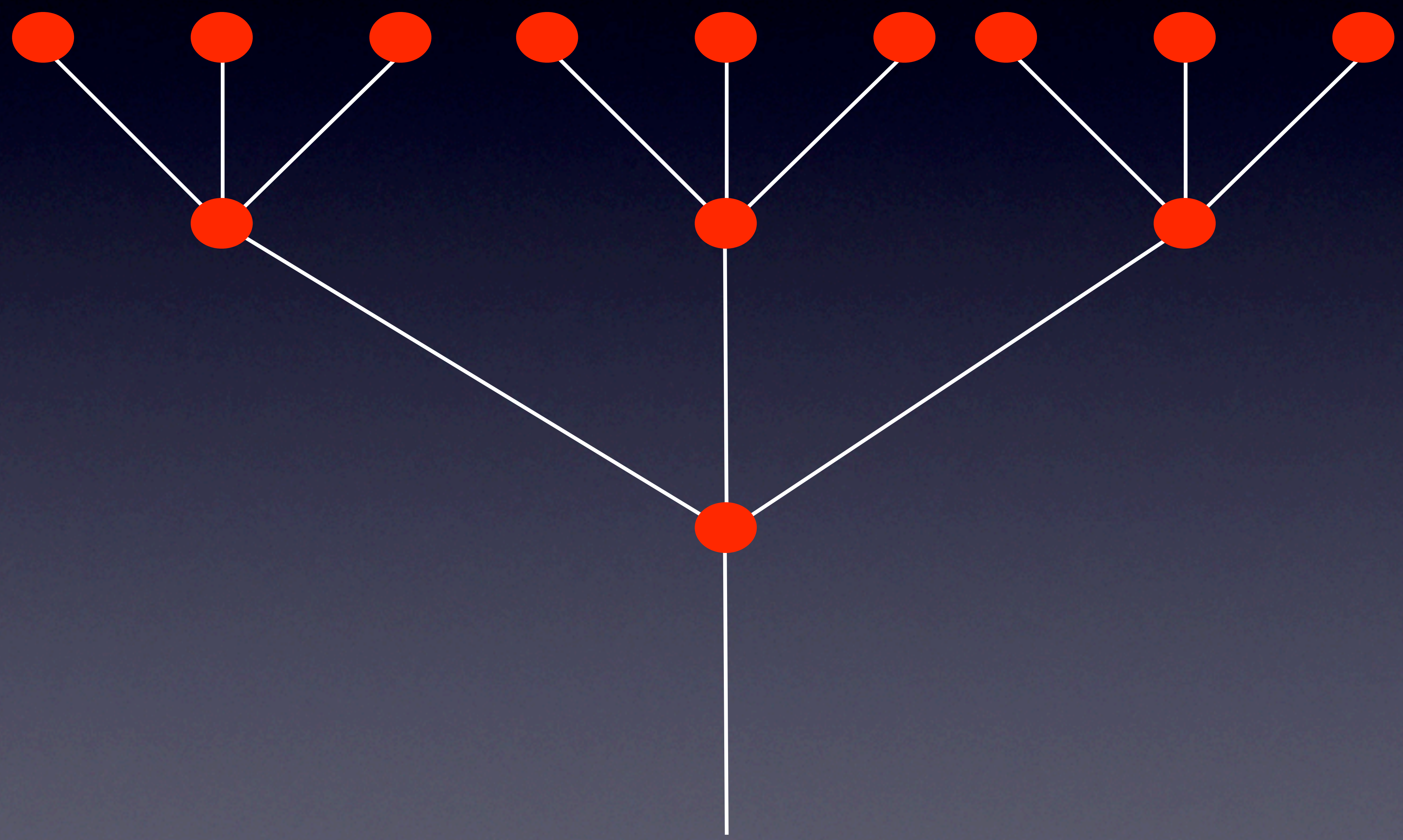
(Replica-Symmetric cavity method)



Solve the model on a tree with the same connectivity

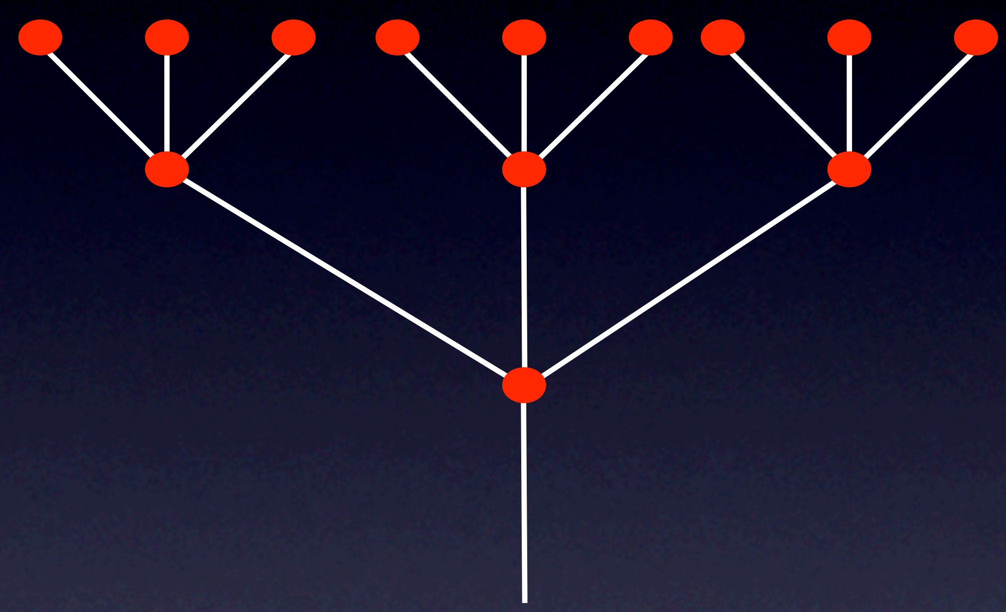
Classical Bethe-Peierls Approximation

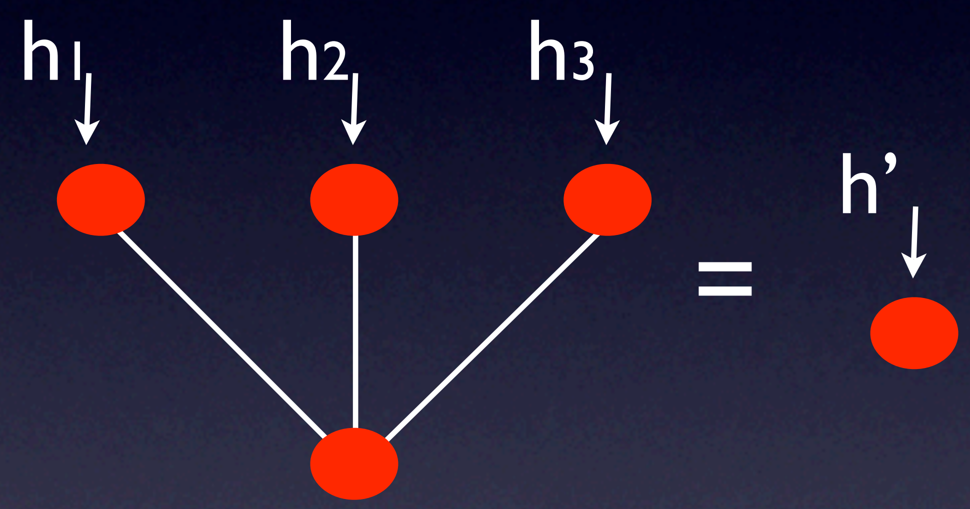
The Cavity Method: solving by recursion



Classical Bethe-Peierls Approximation

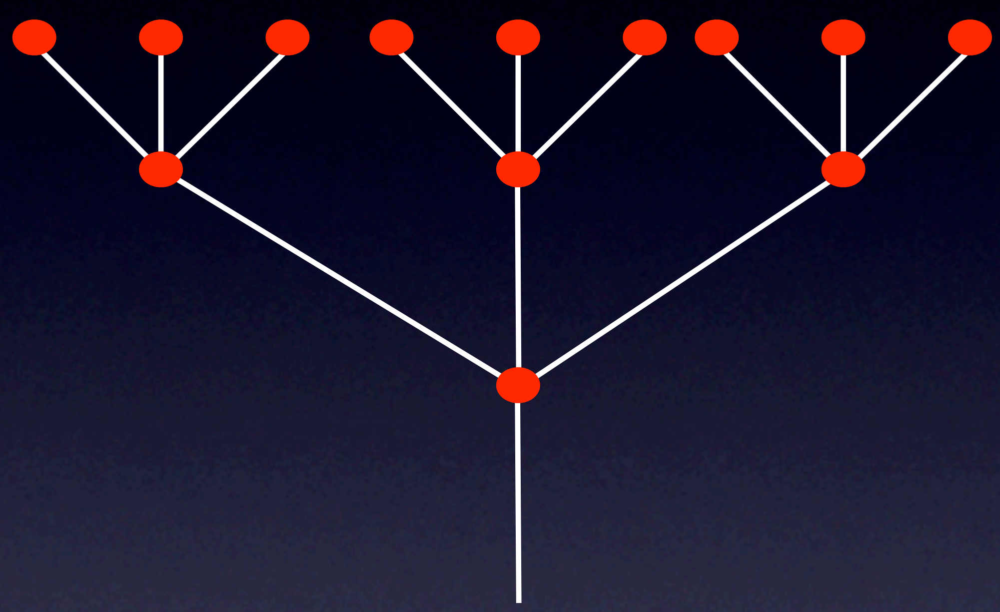
The Cavity Method: solving by recursion

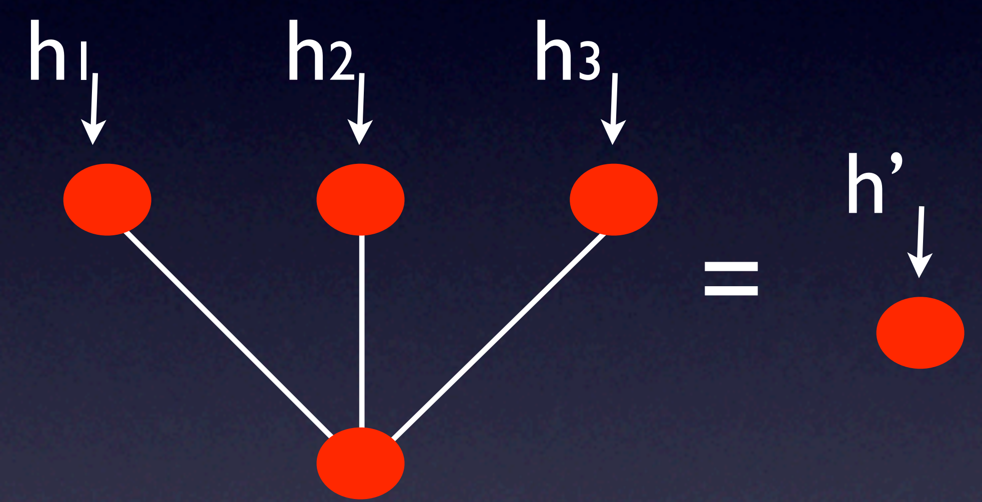




Classical Bethe-Peierls Approximation

The Cavity Method: solving by recursion





$$h' = \sum_{i=1}^{3} \frac{1}{\beta} \tanh^{-1} \left(\tanh \beta h_i \tanh \beta J \right)$$

Classical Bethe-Peierls Approximation

The Cavity Method: solving by recursion

Classical Bethe-Peierls Approximation

The Cavity Method: solving by recursion

Fixed Point

$$h = \frac{c-1}{\beta} \tanh^{-1}\left(\tanh \beta h \tanh \beta J\right)$$

BP

1d=no transition

β(2d)=0.346

β(3d)=0.203

β(4d)=0.144

β(5d)=0.112

Monte-Carlo

1d=no transition

β(2d)=0.44

β(3d)=0.221

β(4d)=0.149

β(5d)=0.114

Classical Bethe-Peierls Approximation

The Cavity Method: solving by recursion

Fixed Point

$$h = \frac{c-1}{\beta} \tanh^{-1} \left(\tanh \beta h \tanh \beta J \right)$$

BP

1d=no transition

β(2d)=0.346

β(3d)=0.203

β(4d)=0.144

β(5d)=0.112

Good quantitative approximation !

Monte-Carlo

1d=no transition

β(2d)=0.44

β(3d)=0.221

β(4d)=0.149

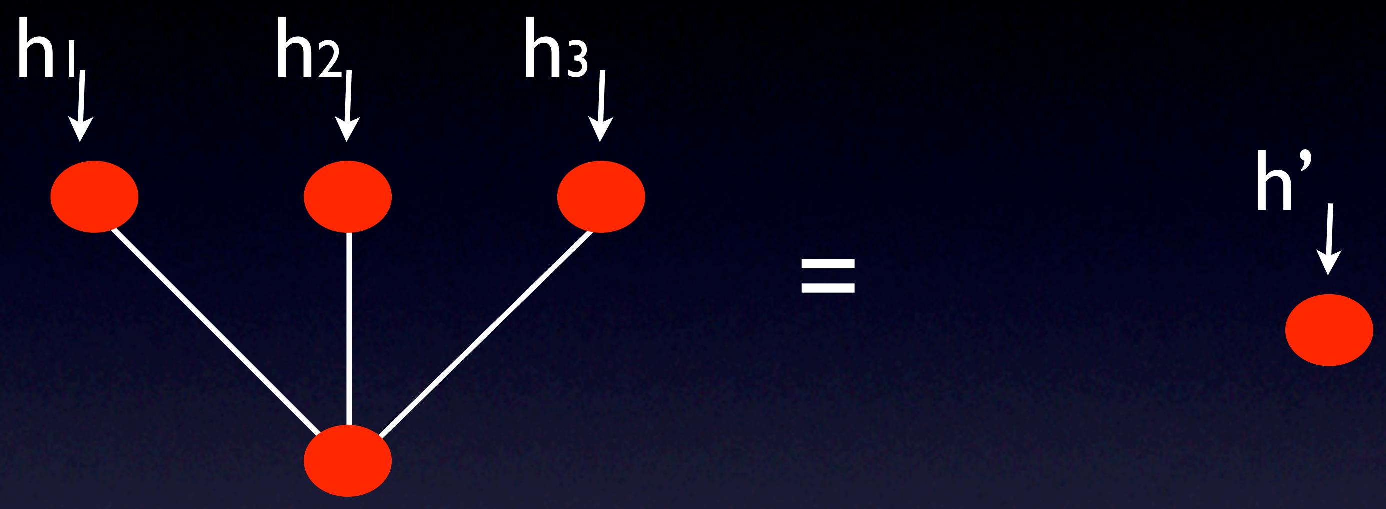
β(5d)=0.114

One field is enough for Ising spins

$$p_{up} = \frac{e^{\beta h}}{Z}$$

$$p_{down} = \frac{e^{-\beta h}}{Z}$$

One field is enough for Ising spins

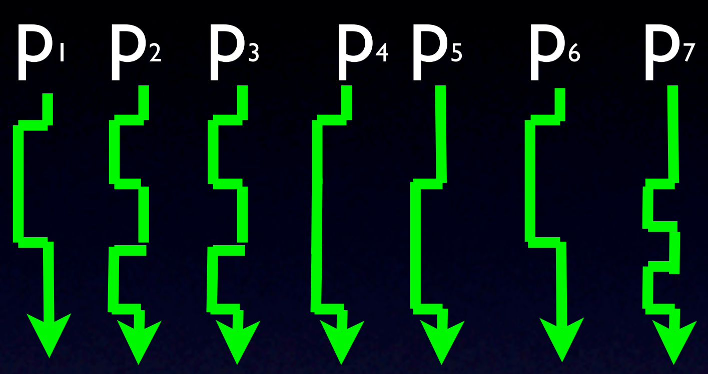


$$p_{up} = \frac{e^{\beta h}}{Z}$$

$$p_{down} = \frac{e^{-\beta h}}{Z}$$

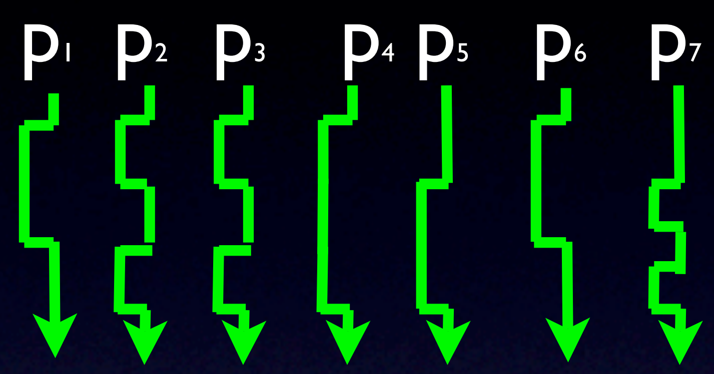
$$h' = \sum_{i=1}^{3} \frac{1}{\beta} \tanh^{-1} \left(\tanh \beta h_i \tanh \beta J \right)$$

But not for quantum spins !!!!



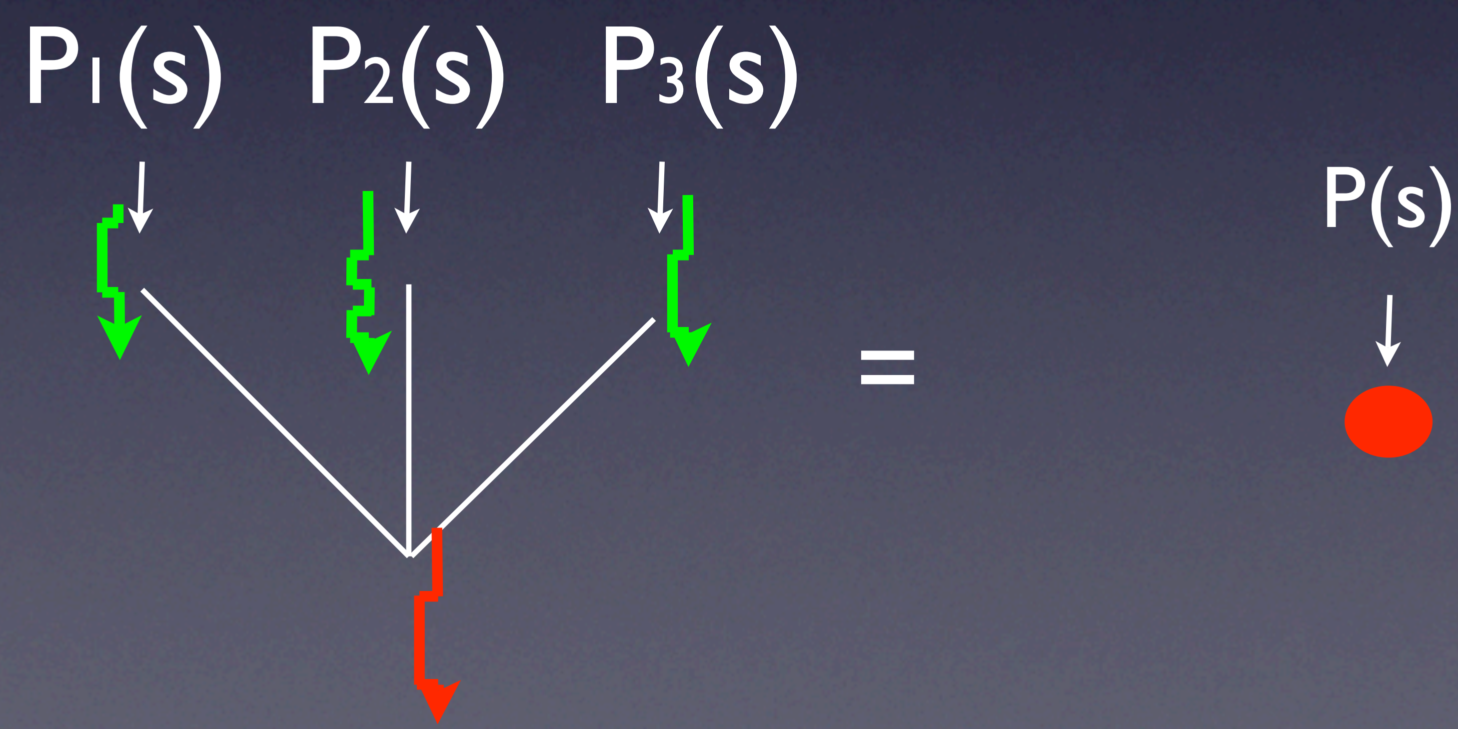
The probability distribution P(s) is a quite complicated object !

But not for quantum spins !!!!

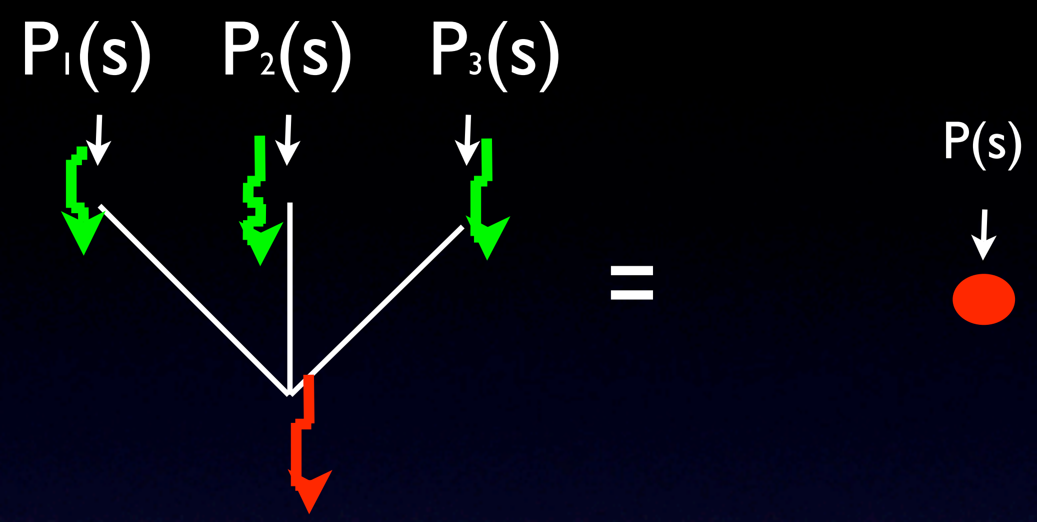


The probability distribution P(s) is a quite complicated object !

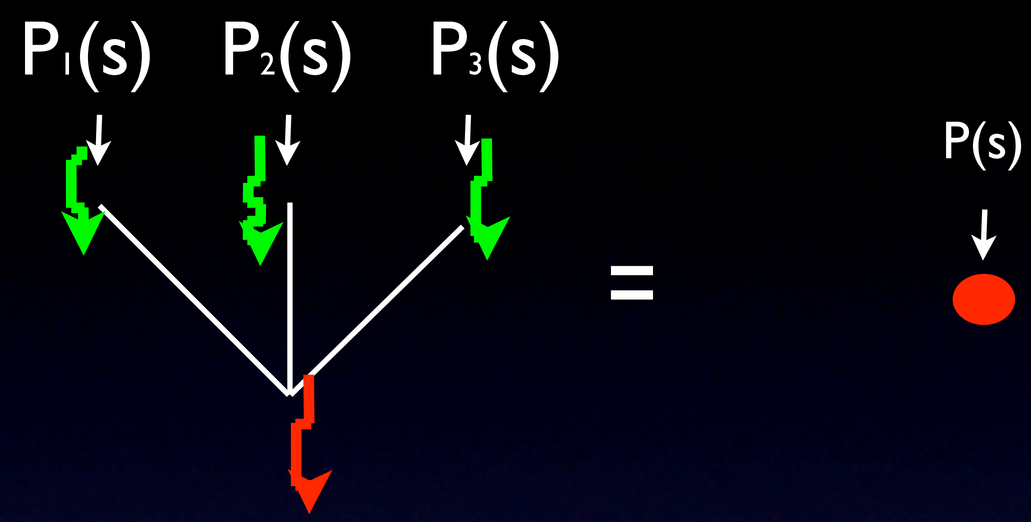
Need for a recursion for P(s) !



Structure of the recurrence equation

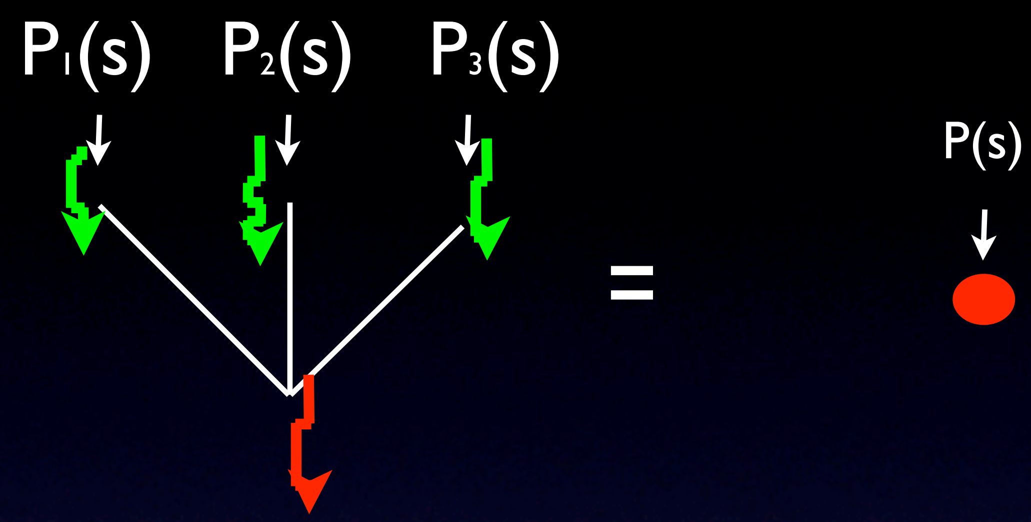


Structure of the recurrence equation



$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)e^{\beta(s_1+s_2+s_3)s}\frac{\omega(s)}{Z}$$

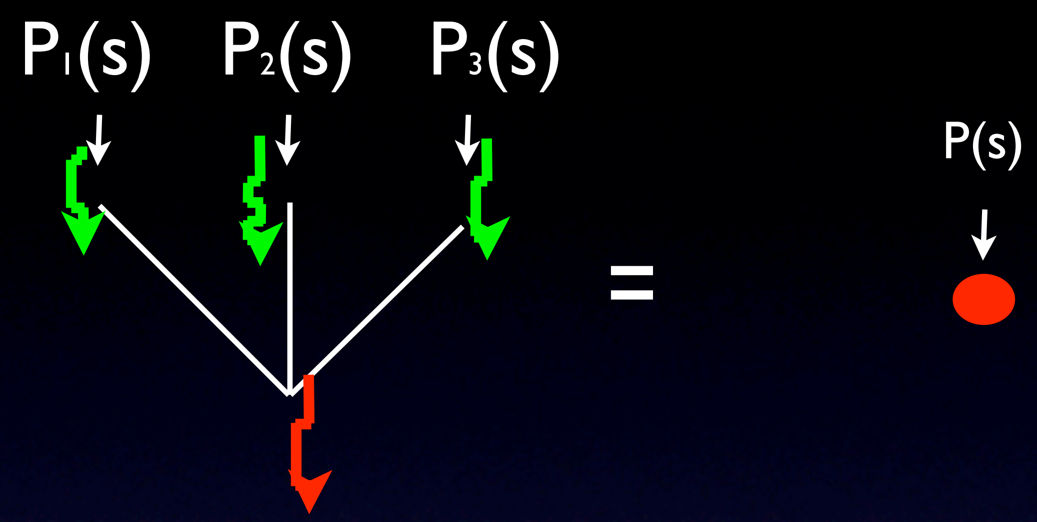
Structure of the recurrence equation



$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)e^{\beta(s_1+s_2+s_3)s}\frac{\omega(s)}{Z}$$

Define the probability distribution given a "field trajectory h" $\longrightarrow$ $p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$

Structure of the recurrence equation



$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)e^{\beta(s_1+s_2+s_3)s}\frac{\omega(s)}{Z}$$

Define the probability distribution given a "field trajectory h" $\longrightarrow$ $p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$

Note h=s1+s2+s3 and rewrite the recursion as

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

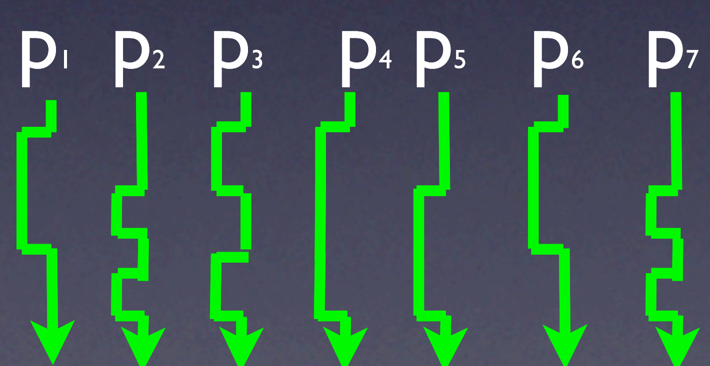
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1, s_2, s_3} P_1(s_1) P_2(s_2) P_3(s_3) p(s | s_1 + s_2 + s_3) \frac{\mathcal{Z}(s_1 + s_2 + s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})} \omega(s) e^{\beta h s}$$

Use the “population” representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



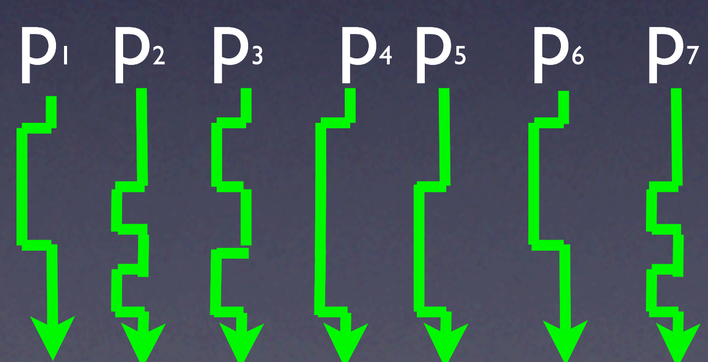
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the "population" representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



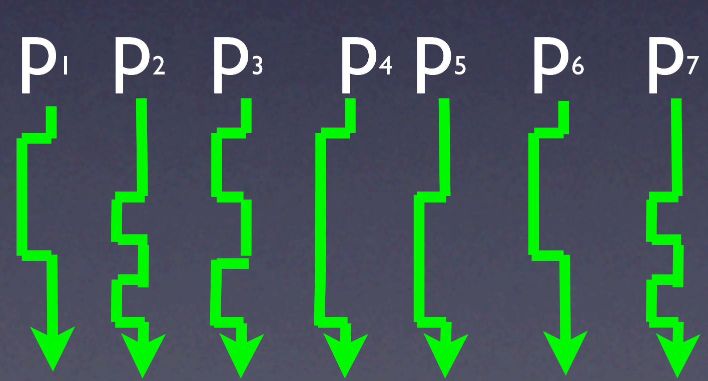
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the "population" representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



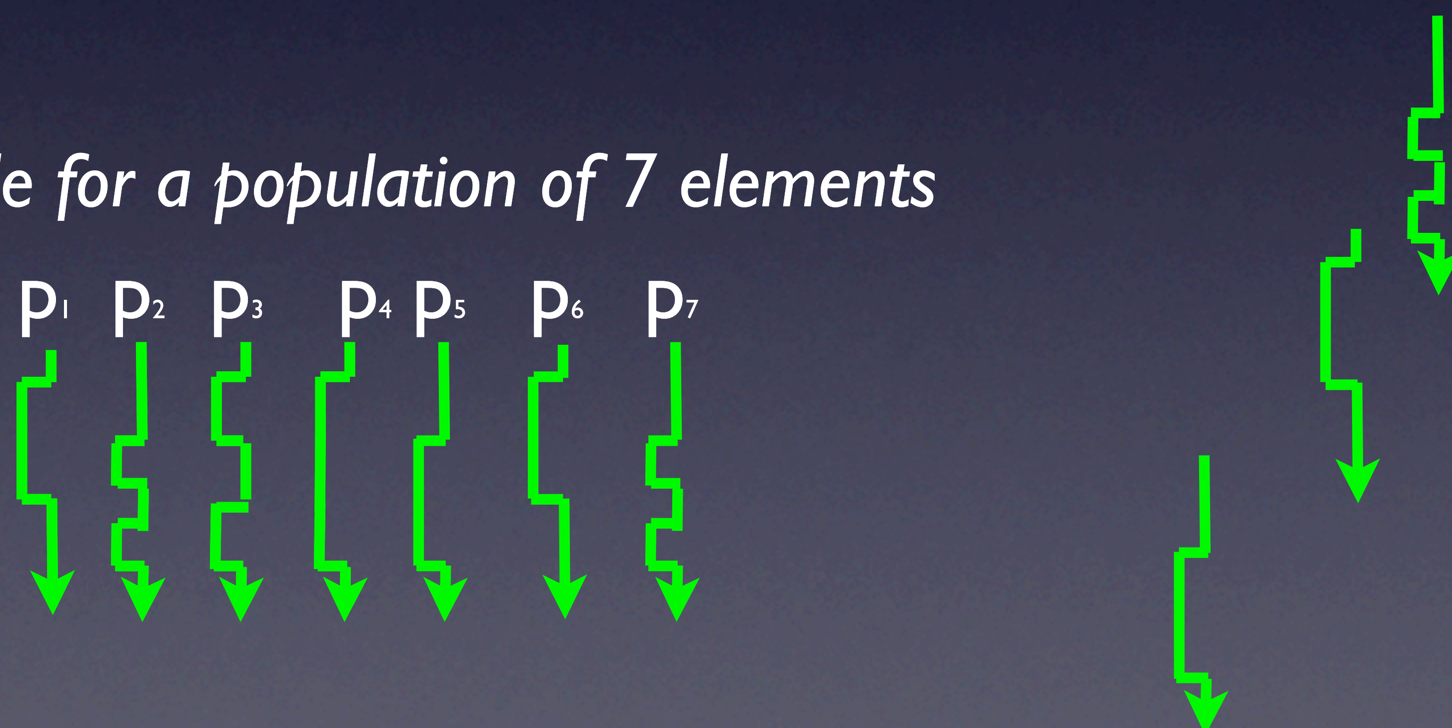
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the “population” representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



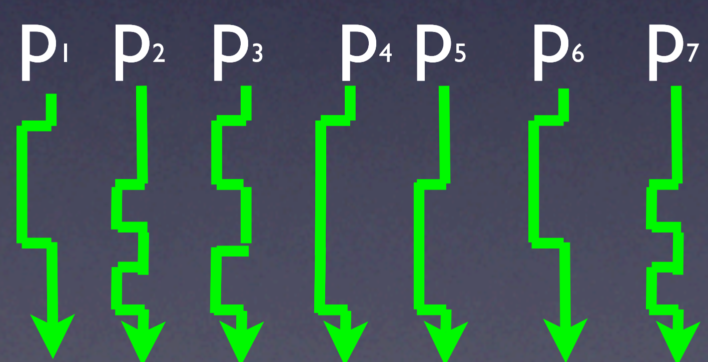
Solving the problem with Population Dynamics

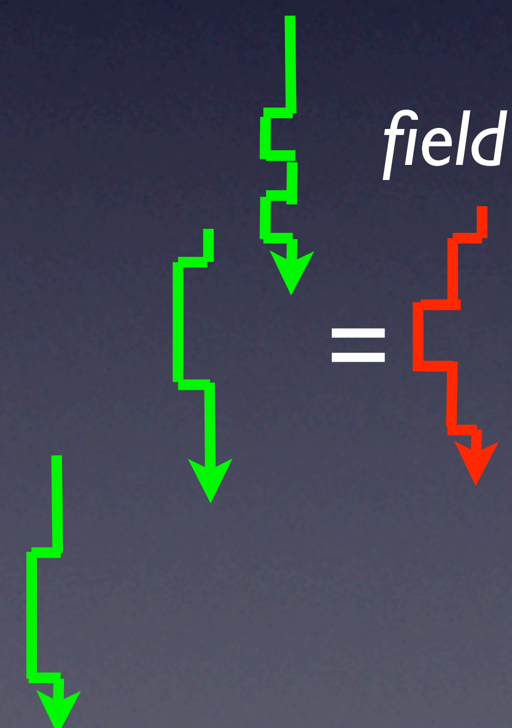
$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the “population” representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements





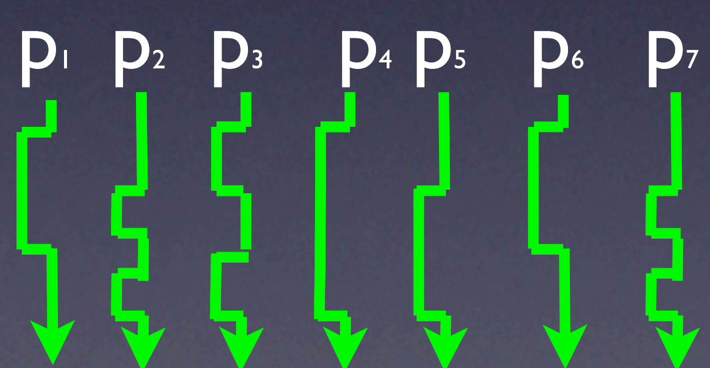
Solving the problem with Population Dynamics

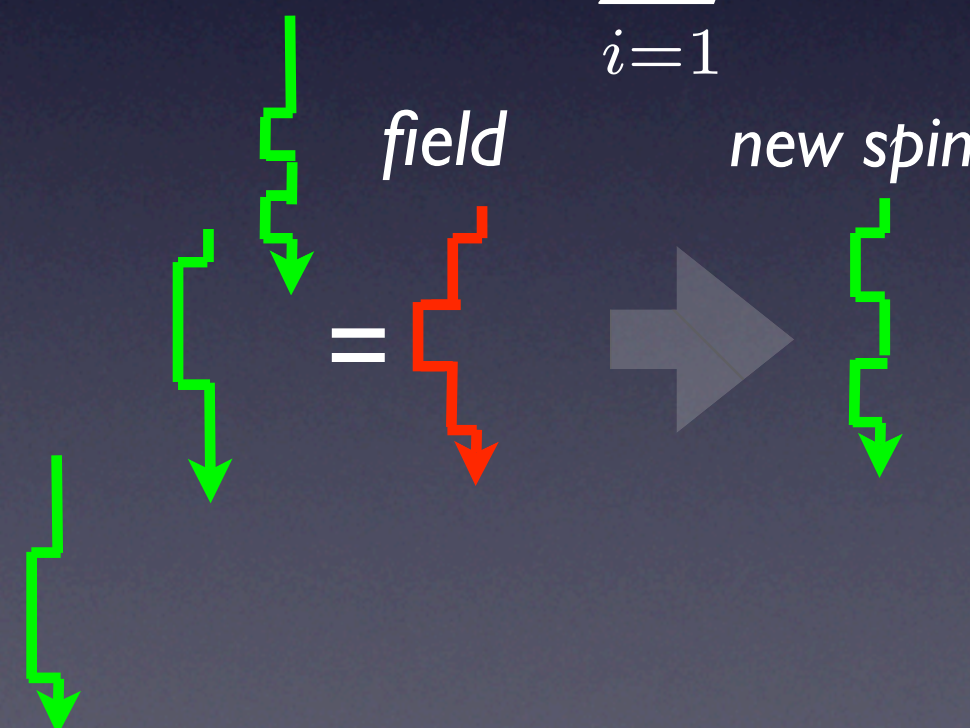
$$P(s) = \sum_{s_1, s_2, s_3} P_1(s_1) P_2(s_2) P_3(s_3) p(s|s_1+s_2+s_3) \frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})} \omega(s) e^{\beta h s}$$

Use the “population” representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements





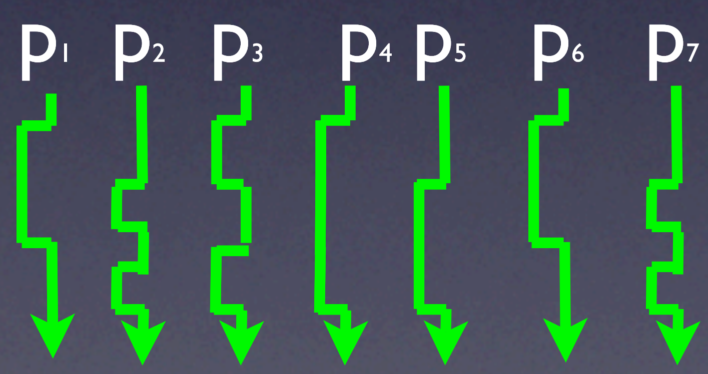
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the “population” representation :
$$P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$$

Example for a population of 7 elements



new population

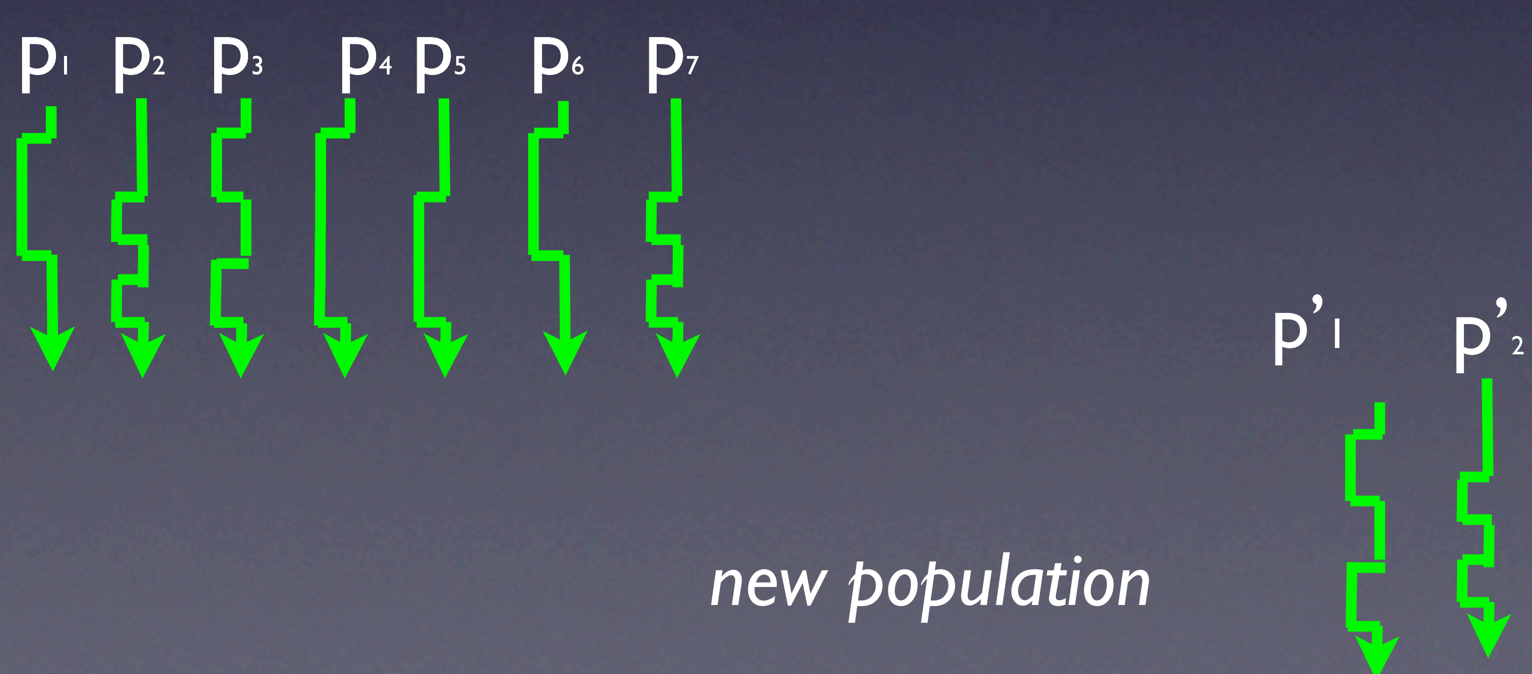
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the "population" representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



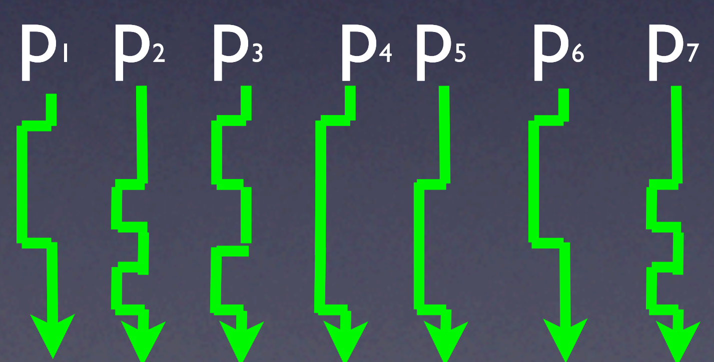
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the “population” representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



new population

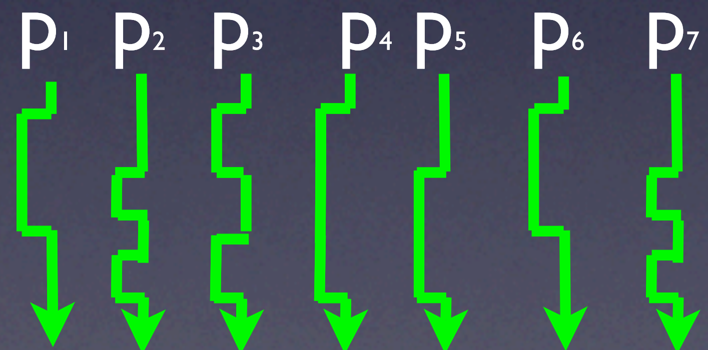
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1) P_2(s_2) P_3(s_3) p(s|s_1+s_2+s_3) \frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

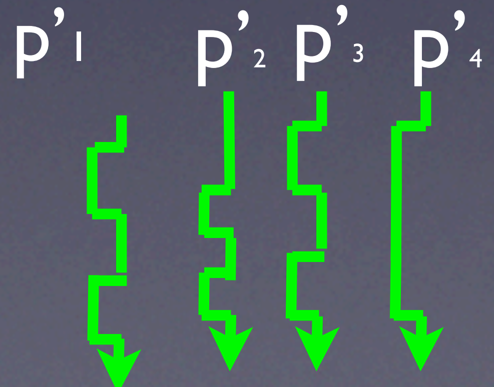
$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})} \omega(s) e^{\beta h s}$$

Use the “population” representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



new population



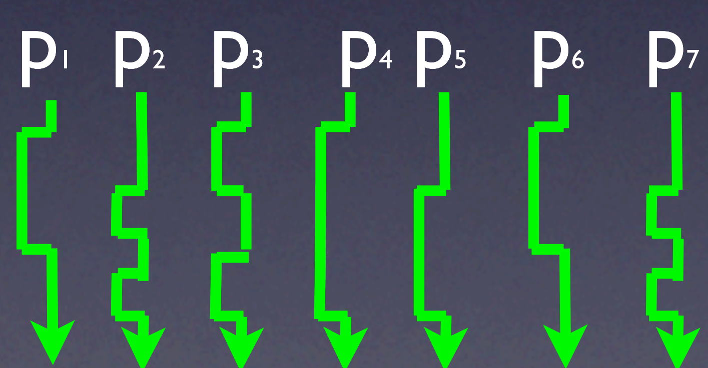
Solving the problem with Population Dynamics

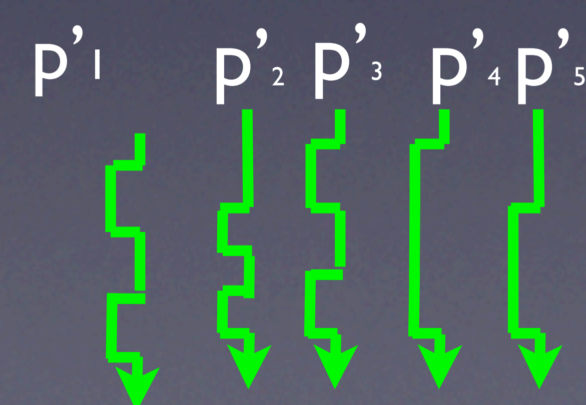
$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the "population" representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements





new population

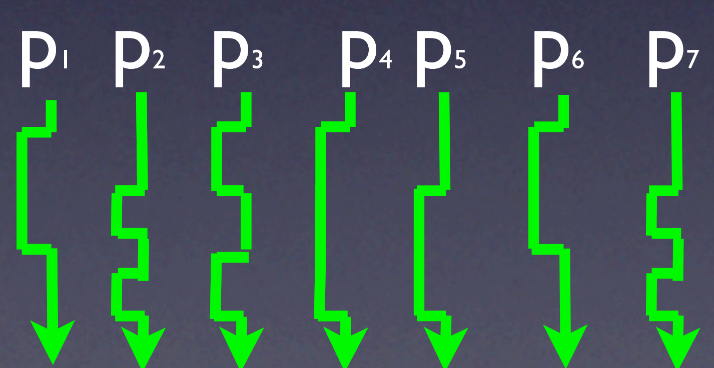
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

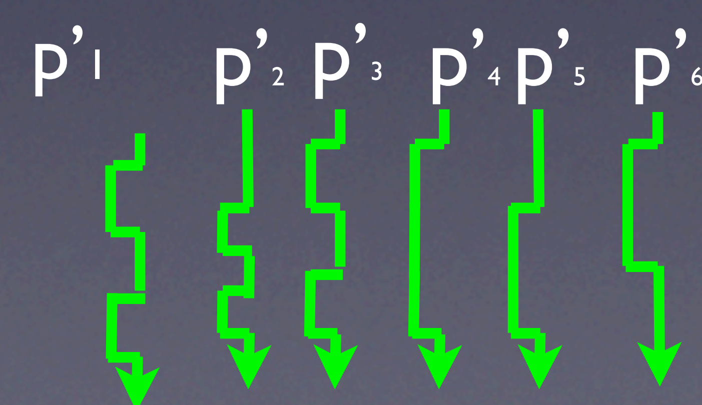
$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the "population" representation : $P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$

Example for a population of 7 elements



new population



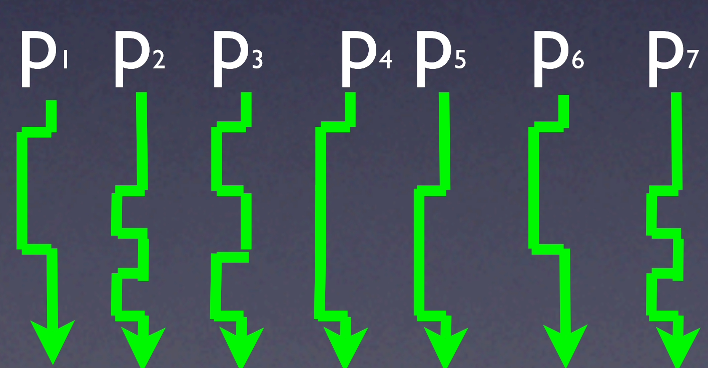
Solving the problem with Population Dynamics

$$P(s) = \sum_{s_1,s_2,s_3} P_1(s_1)P_2(s_2)P_3(s_3)p(s|s_1+s_2+s_3)\frac{\mathcal{Z}(s_1+s_2+s_3)}{Z}$$

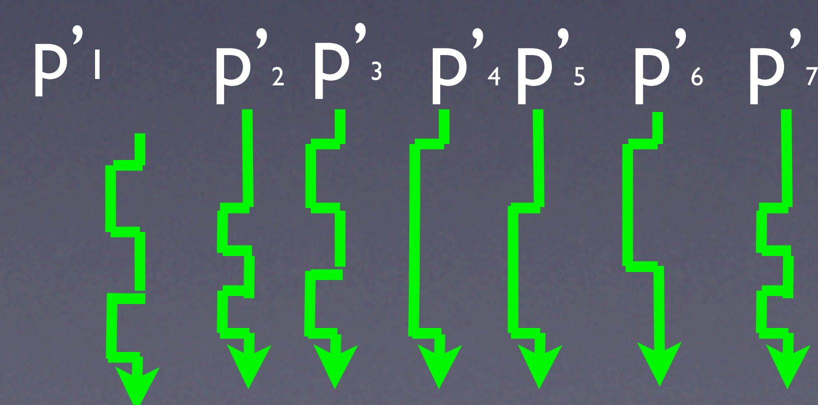
$$p(s|\mathbf{h}) = \frac{1}{\mathcal{Z}(\mathbf{h})}\omega(s)e^{\beta h s}$$

Use the "population" representation : $$P(s) = \sum_{i=1}^{\mathcal{N}} p_i \delta(\sigma - \sigma_i)$$

Example for a population of 7 elements

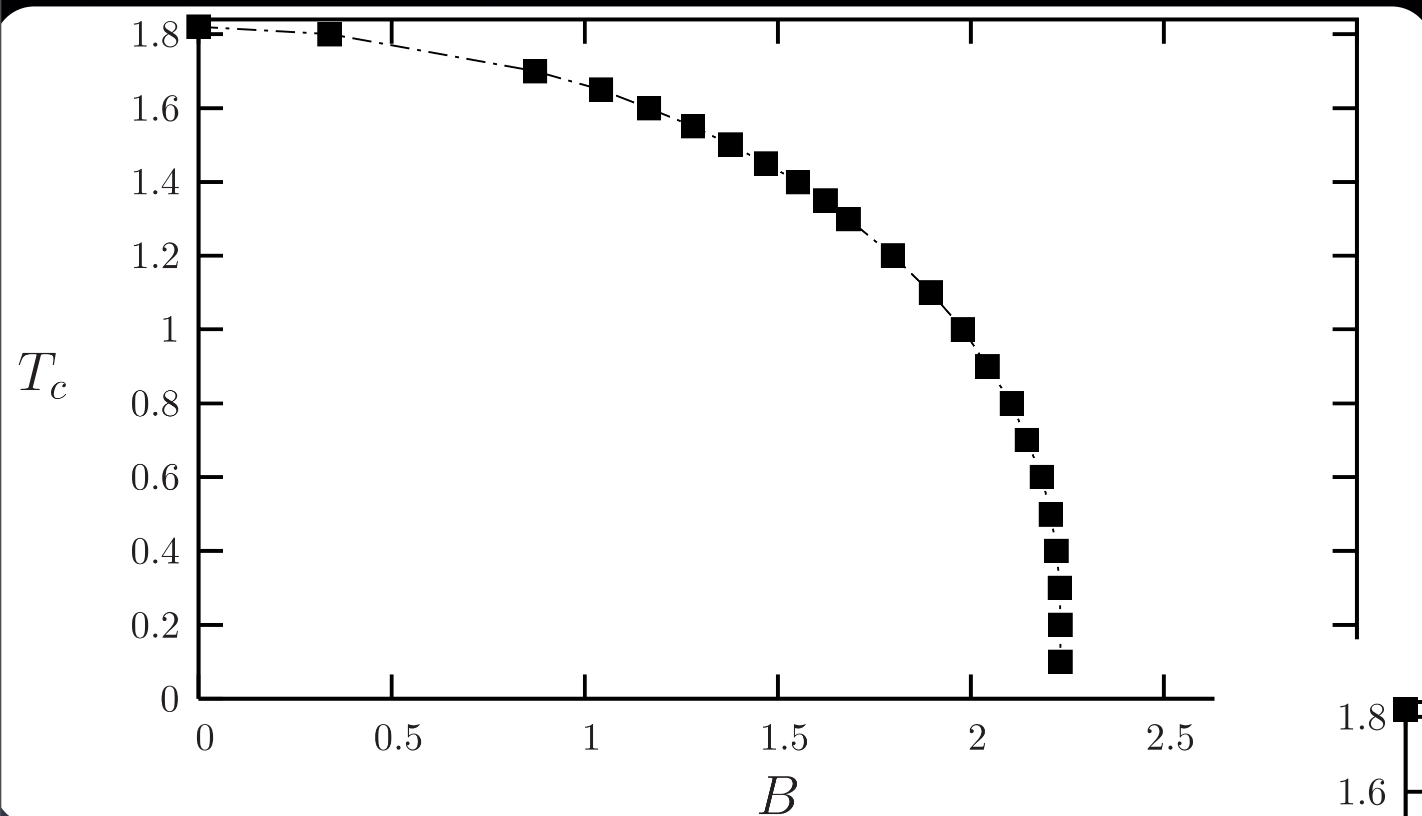


new population

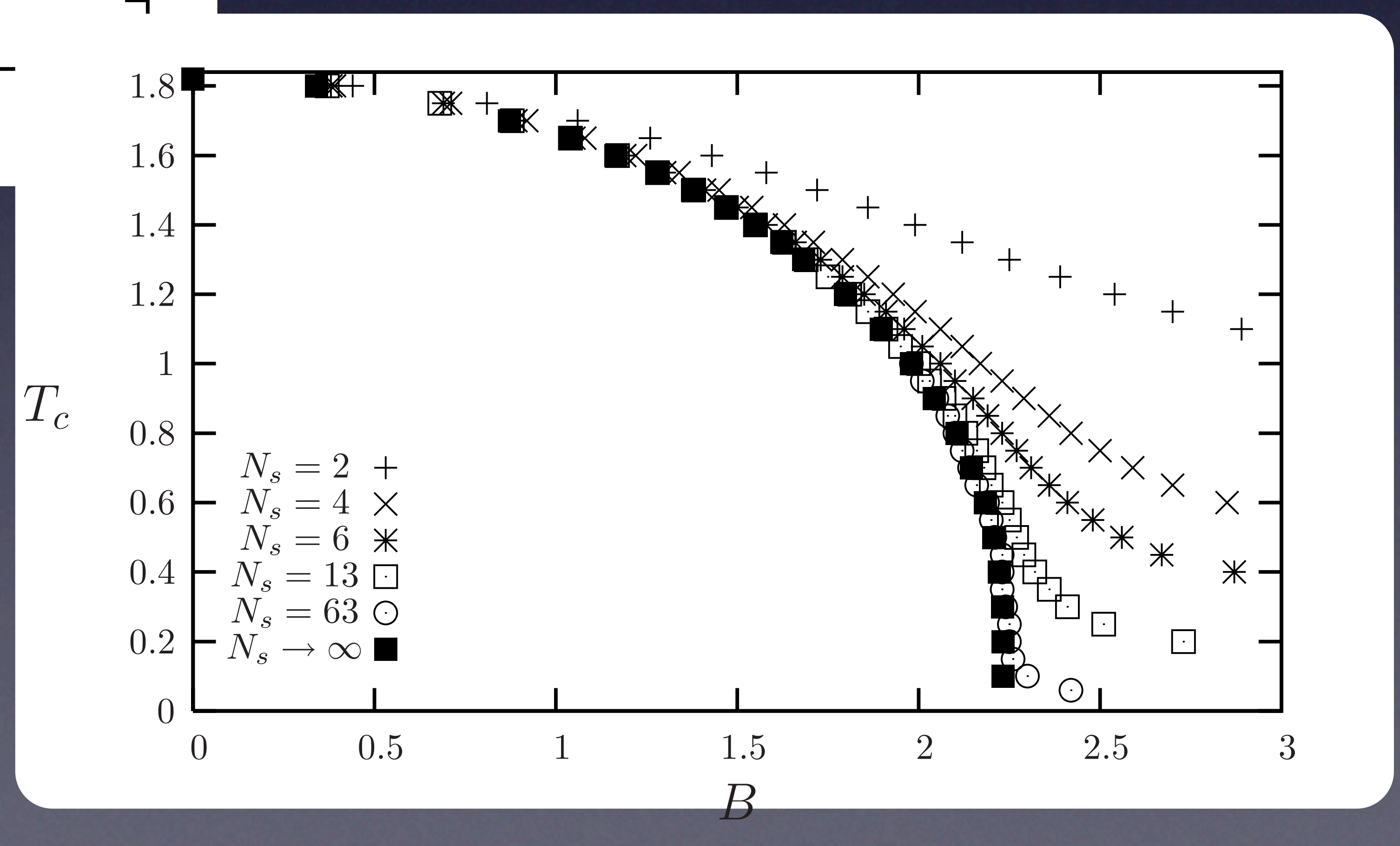


Some Results

Ising ferromagnet in transverse field on a random 3-regular graph

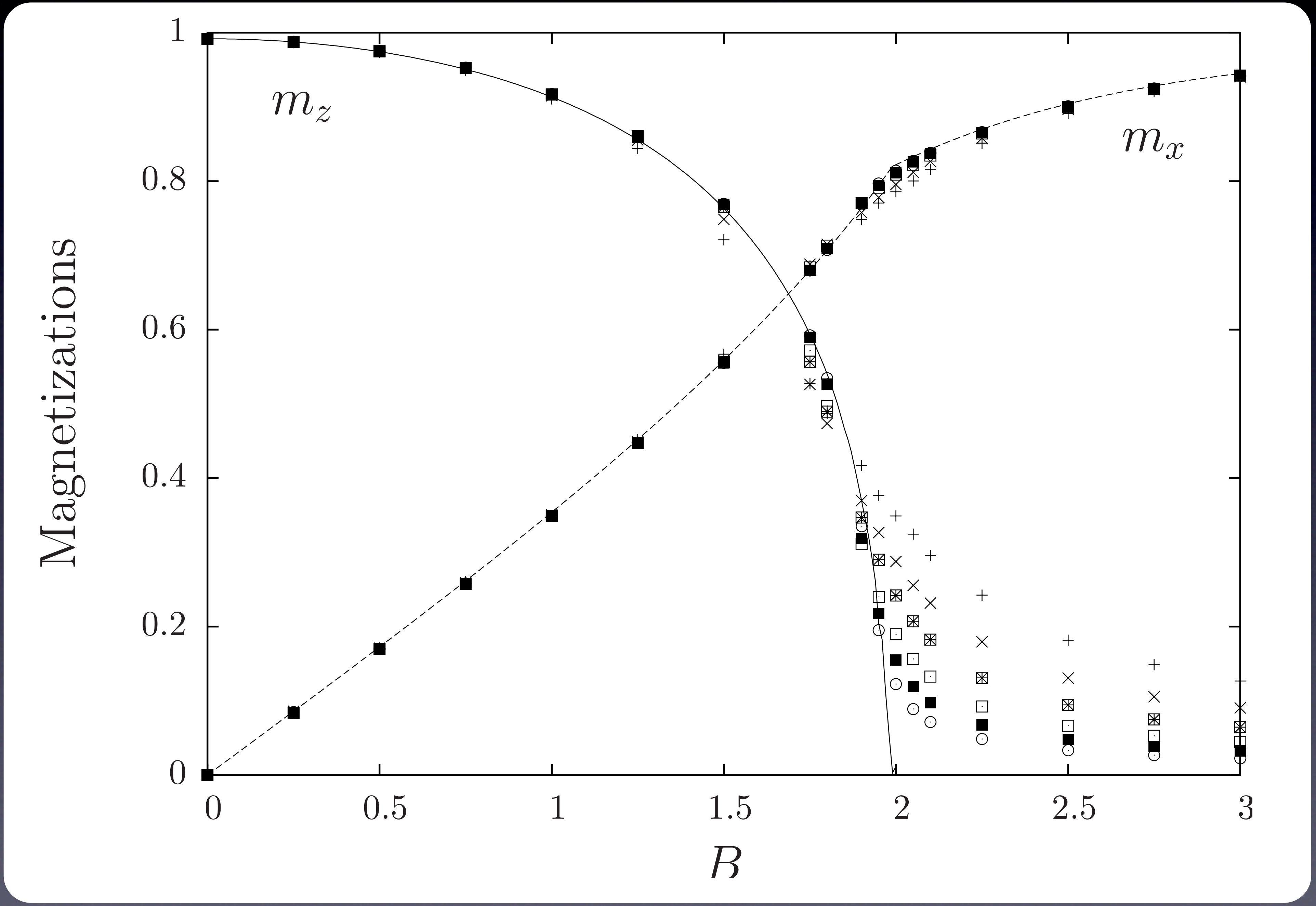


Versus ⇨



Some Results

Ising ferromagnet in transverse field on a random 3-regular graph



T=1

Conclusions...

- A heat bath method for generic quantum spin-1/2 models in transverse field
- Allows to formulate a quantum version of the cavity method to solve the same models on trees (or more generally on random graphs)

... and perspectives

- Simulation of quantum spin-1/2 problem where no loop algorithm is known (Quantum Spin Glasses, Quantum Constraint Satisfaction Problems....)
- Application of the quantum cavity method to the same models on trees/random graphs
- Application to particles systems (Bosonic Hubbard model) e.g. to study glassy phases of cold atoms in disordered potentials
- Application to dynamics of classical models?

Thanks to my collaborators...

Guilhem Semerjian

Alberto Rosso

Florent Krzakala

...and to you for your attention!