Lecture#1
Path integral and Mote Carlo simulations

Ken-Ichi Ishikawa

Hiroshima Univ.

1. Lattice Field Theory?

- For example:
- Standard Model
  - QCD: Strong interaction, Hadrons <= quarks, gluons
  - Glashow-Weinberg-Salam: Electroweak
    - QED: electromagnetic interaction: charged particles, photon
    - Weak interaction: Z,W bosons, leptons,…
    - Higgs mechanism
- These are based on Quantum field theory (QFT).
  - Perturbative analysis using the coupling constant expansion.
  - Rely on the smallness of the coupling.
- QCD: at low-energy, coupling expansion fails.
  - Non-preturbative analysis is required.

- To understand the nature of the strong interaction among Hadrons from the dynamics of quarks and gluons, Quantum Chromodynamics (QCD) has been introduced and investigated.
- QCD is well under stood in the high-energy experiments where the asymptotic-free nature of the coupling constant of QCD enables us the perturbative expansion analysis.
- Howerver, at low-energy, the perturbative analysis fails due to the large coupling constant.

- The lattice field theory is one of the non-preturbative analysis method.
- Lattice QCD has been used and developed to understand the low-energy nature of Hadrons.
- The various technique for lattice field theory is common and also has been used in LQCD.
- In this lecture I would like to give some lattice technique and numerical algorithms for LQCD as an example of lattice field theories.

2. Path integral and lattice field theory

- Feynman's path integral quantization is a fundamental basis for lattice field theory.
- Euclidean field is also required to introduce well defined (numerically calculable) path integral formulation.
- Lattice QCD is based on SU(3) gauge theory defined on a Euclidean 4Dim lattice universe.

2-1 Feynman's path integral quantization

- A quantum field theory :
  - $S[\phi]$: Action.
  - $\phi(x)$: Field to be quantized (real scalar for simplicity).
  - x : space-time corrdinate.
- Feynman's path integral quantization.
  - Generating functional for Green's functions (correlation func.)

$$Z[\eta] = \int D\phi \exp\left[\frac{i}{\hbar}\left(S[\phi] + \eta \cdot \phi\right)\right]$$

  - N-point Green's function of the theory.

$$\left\langle T[\hat{\phi}(x_1)\hat{\phi}(x_2)\cdots\hat{\phi}(x_n)]\right\rangle = \frac{\hbar^n}{i^n Z[0]} \frac{\delta^n Z[\eta]}{\delta\eta(x_1)\delta\eta(x_2)\cdots\delta\eta(x_n)}\bigg|_{\eta=0}$$

$$= \frac{1}{Z[0]}\int D\phi\left(\phi(x_1)\phi(x_2)\cdots\phi(x_n)\right)\exp\left[\frac{i}{\hbar}S[\phi]\right]$$

  - We can extract various information from Green's functions basically….

- However, the analytic integration of the path-integral is not always available except for free field theories.
- The integral also has a difficulty in Minkowski metric. The integral is a kind of Fresnel integrals and the integrant oscillates. This may prevents us to evaluate it numerically….
- In order to evaluate this integral:
  - Introduce Euclidean path integral
    - Needs validation : Minkowski⇔ Euclid relation. Experimentally or constructive field theory, Osterwalder-Schrader axioms…
  - Discretize Space-Time => Lattice space-time
    - Needs validation: lattice spacing error
- Here we assume:
  - there is a Euclidean field theory for a target Minkowski field theory.

2-2 Euclidean path integral

- $S_E[\phi_E]$: Euclidean action.
- $\phi_E(x_E)$: Euclidean field. Real valued.
- $x_E = (x, y, z, \tau)$: Euclidean 4D coordinate.
  - They are usually obtained from Minkowski versions after Wick's rotation. $t = i\tau$

- Generating functional for Euclidean Green's functions.

$$Z_E[\eta] = \int D\phi_E \exp\left[-\frac{1}{\hbar}\left(S_E[\phi_E] - \eta \cdot \phi_E\right)\right]$$

- If the Euclidean action is real valued, the integral has a better property than the Mikowski version. A chance to evaluate them by numerical integration?
- The physics information can be obtained from Euclidean Green's functions by inverse Wick's rotation or investigating the tau dependence.

$$\left\langle \phi_E(\vec{x},\tau)\phi_E(\vec{0},0)\right\rangle = \frac{1}{Z_E[0]} \frac{\delta^2 Z_E[\eta]}{\delta\eta(\vec{x},\tau)\delta\eta(\vec{0},0)}\bigg|_{\eta=0} = \frac{1}{Z_E[0]}\int D\phi_E(\phi_E(\vec{x},\tau)\phi_E(\vec{0},0))\exp\left[-S_E[\phi_E]/\hbar\right]$$

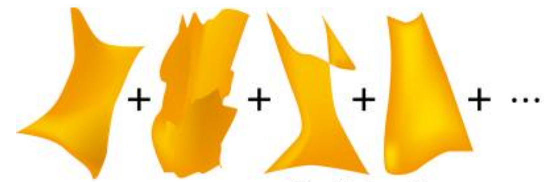
$$\int d\vec{x}\left\langle \phi_E(\vec{x},\tau)\phi_E(\vec{0},0)\right\rangle e^{-i\vec{p}\cdot\vec{x}} \xrightarrow[\tau\to+\infty]{} Ce^{-E(\vec{p})\tau}$$

$E(\vec{p})$: lowest energy in this channel (intermediate state).

2-3 Euclidean path integral and lattice

- Path integral measure $\int D\phi_E$
  - Integration by field shape (configuration)

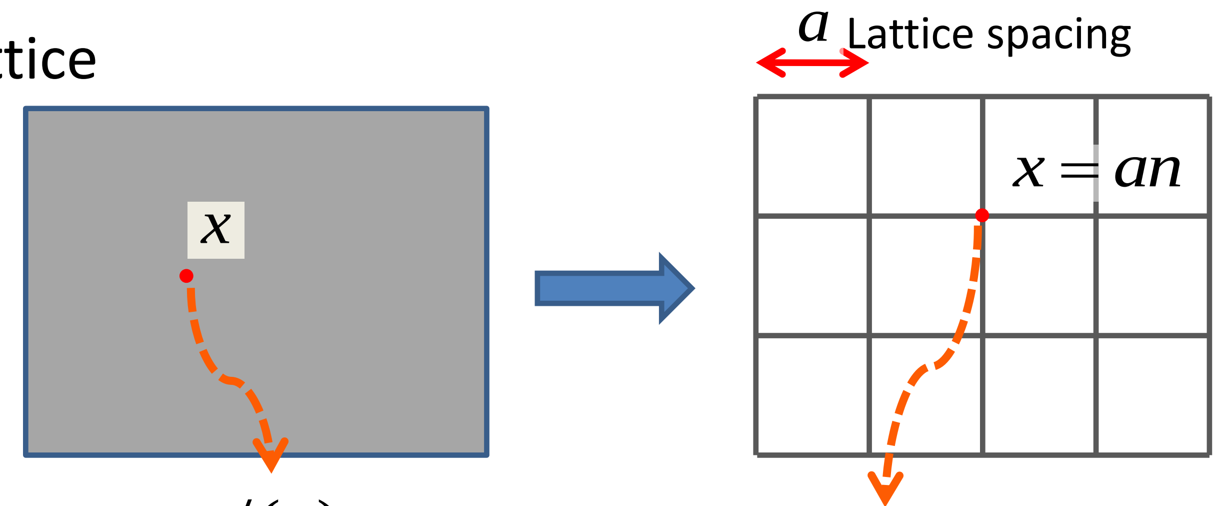
$$\int D\phi_E \approx$$



sum over field shape

  - Euclidean space time, $x_E = (x, y, z, \tau)$ is continuous. Difficult to maintain $\int D\phi_E$ for numerical evaluation. This will cause UV divergences. The renormalization and regularization is required.
  - Introduce the lattice discretization:
    - As a regularization.
    - As a well defined integration measure.
      - Degree of Freedom (DoF) is still finite. IR regulator by limiting system size (finite volume).

- Lattice



$$\phi(x) \qquad\qquad a\phi(na) \equiv \phi_{Latt}(n)$$

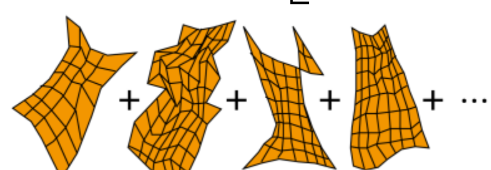
- Lattice regularized path integral

$$n = (n_x, n_y, n_z, n_\tau) \in Z^4$$

$$Z_E[\eta] = \int D\phi_E \exp\left[-\frac{1}{\hbar}(S_E[\phi_E] - \eta\cdot\phi_E)\right]$$

$$Z_{Latt}[\eta] = \int D\phi_{Latt} \exp\left[-\frac{1}{\hbar}(S_{Latt}[\phi_{Latt}] - \eta\cdot\phi_{Latt})\right]$$

$$\int D\phi_{Latt} \equiv \prod_n \int d\phi_{Latt}(n) =$$



sum over lattice field shape

Multiple integration on the vector $\vec{\phi} = (\cdots\phi_{Latt}(n_1), \phi_{Latt}(n_2), \phi_{Latt}(n_3), \cdots)^T$

- Lattice regularized path integral

$$Z_{Latt}[\eta] = \int D\phi_{Latt} \exp\left[-\frac{1}{\hbar}\left(S_{Latt}[\phi_{Latt}] - \eta\cdot\phi_{Latt}\right)\right] \qquad n = (n_x, n_y, n_z, n_\tau) \in Z^4$$

$$\Rightarrow \quad Z(\vec{\eta}) = \int d\vec{\phi} \exp\left[-\left(S(\vec{\phi}) - \vec{\eta}\cdot\vec{\phi} \right)\right]$$

Multiple integration on the vector $\vec{\phi} = \left(\cdots \phi_{Latt}(n_1), \phi_{Latt}(n_2), \phi_{Latt}(n_3), \cdots\right)^T$

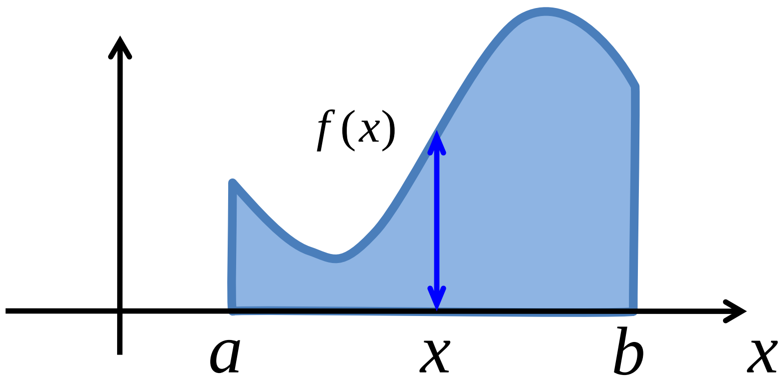
- How to evaluate this integral?
  - Similar to Canonical partition function in statistical mechanics.
  - Dimension of $\vec{\phi}$ is very large. For real scalar on a 4D lattice with the size (16x16x16x16), $\dim(\vec{\phi}) = 16^4 = 65{,}536$
  - If the weight exp(…) is real and non-negative, we can evaluate it using Monte Carlo Methods.
  - Note: Lattice action should be designed appropriately. (based on Symmetry, spectrum, relation Minkowski⇔ Euclid, …..)
  - When no real and non-negative weight is derived, we encounter the sign problem in the Monte Carlo method. Ex. System in finite density.

2-4 Integration using Monte Carlo Methods.

- Monte Carlo
  - Ex. Integration with a single variable.

$$I = \int_a^b f(x)dx$$

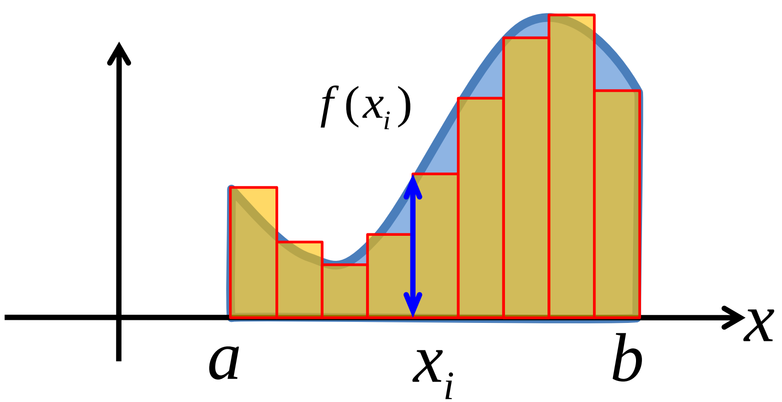
$$f(x) \geq 0, \quad \text{and real valued.}$$



  - Rectangular integration

$$I = \lim_{N \to \infty} f(x_i)\Delta x,$$

$$\Delta x = (b-a)/N$$



- Random sampling

(1) Pick up a number $x_i = x$ from the interval $[a,b]$ randomly.

(2) Evaluate function as $f_i = f(x_i)$.

(3) Repeat (1)-(2) N times, then we get samples $\{f_1, f_2, \cdots, f_N\}$.

We can estimate the integral as

$$I \approx \frac{b-a}{N}\sum_{i=1}^{N} f_i$$

The random number sequence $\{x_1, x_2, \cdots, x_N\}$ has a uniform distribution in [*a*,*b*]. This means that the random variable *x* has the following probability density:

$$P(x) = \begin{cases} Const & x \in [a,b] \\ 0 & x \notin [a,b] \end{cases}$$

Thus the statistical averaging for $f_i = f(x_i)$ means

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} f_i = \int_{-\infty}^{\infty} f(x)P(x)dx \Big/ \int_{-\infty}^{\infty} P(x)dx$$

Denominator is for the probability normalization.

- A defect and inefficient property of the simple rectangular and random sampling integration.
  - If the target function f(x) has a keen peak with narrow width *W*,

The integration may fail until

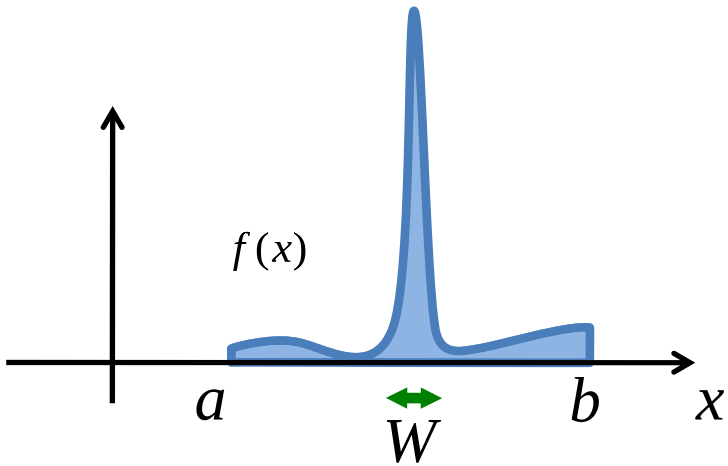
$$(a-b)/N << W$$

is satisfied.

One sample in the peak.

Most of samples are unimportant.

sample ratio = *1/N*.



  - In multi dimensional integrations, the situation becomes more worse.
  - D-dimension

$$I = \iiint \dots \iint f(\vec{x}) d^D \vec{x}$$

$$(a-b)_i / N_i << W_i \quad \text{for} \quad i\text{-th direction}$$

$$\text{Total Sample Number} = \prod_{i=1}^{D} N_i \approx N^D$$

Only one sample is in the peak.
Ratio = $1/N^D$ << 1.

- Importance Sampling (Monte Carlo)
  - As seen before uniform sampling is not effective if the integrant has keen peaks.
  - Euclidean Path-integral is a kind of huge-multi dimensional integration.
  - The integral has narrow peaks in general, and the highest peak corresponds to the classical solution of the system.

$$\langle \phi_{Latt}(x)\phi_{Latt}(y)\rangle = \frac{1}{Z_{Latt}[0]}\int D\phi_{Latt}(\phi_{Latt}(x)\phi_{Latt}(y))\exp\left[-\frac{S_{Latt}[\phi_{Latt}]}{\hbar}\right]$$

  - In the classical limit ($\hbar$->0), the dominant contribution to the integral comes from:

$$\text{A solution } \phi_{Latt}{}^{*} \text{ which gives the highest peak of } \exp\left[-\frac{S_{Latt}[\phi_{Latt}{}^{*}]}{\hbar}\right].$$

  - This corresponds to the stationary (or minimum) solution of action:

$$S_{Latt}[\phi_{Latt}{}^{*}+\Delta]\approx 0 \quad \text{for any variation } \Delta. \qquad \phi_{Latt}{}^{*}: \text{classical solution.}$$

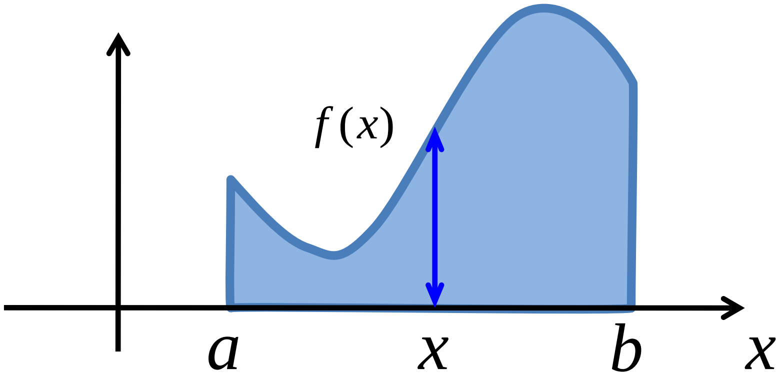
  - We know that the classical solution gives a narrow peak for exp(-S)

Uniform sampling for ϕ is not effective to evaluate path integrals.

- Importance Sampling (Monte Carlo) cont'd

$$I = \int_a^b f(x)dx$$

$$f(x) \geq 0, \quad \text{and real valued.}$$



- To integrate this function *f*(*x*):
  - If we can generate a sequence / ensemble {*x*} so that the statistical histogram/distribution of {x} is *w*(*x*).
  - We have $\{x^{(1)}, x^{(2)}, x^{(3)}, \cdots, x^{(N)}\}$

    Distribution of $\{x\}$: $w(x)$

    $$I = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{w(x)} w(x)dx = \lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} \frac{f(x^{(i)})}{w(x^{(i)})} = \left\langle \frac{f(x)}{w(x)} \right\rangle \text{ : statistical averaging.}$$

  - The error behaves as 1/Sqrt(N)

    $$I = \int_a^b f(x)dx \approx \left\langle \frac{f(x)}{w(x)} \right\rangle \pm \sqrt{\frac{\langle f^2/w^2\rangle - \langle f/w\rangle^2}{N}} = \left\langle \frac{f(x)}{w(x)} \right\rangle \pm O(1/\sqrt{N})$$

  - The error is minimized when *f*(*x*)=*w*(*x*).
  - The error behaves as 1/Sqrt(N) even for the multi-dimensional integrations.

- Importance sampling for Euclidean path-integrals.
  - For the two-point correlation function:

$$\left\langle \phi_i \phi_j \right\rangle = \frac{1}{Z(\vec{0})} \int d\vec{\phi} (\phi_i \phi_j) \exp\left[-S(\vec{\phi}) \right]$$

  - Generate a sequence/ensemble: $\left\{ \vec{\phi}^{(1)}, \vec{\phi}^{(2)}, \vec{\phi}^{(3)}, \cdots, \vec{\phi}^{(N)} \right\}$
  - So that the sample has the distribition : $w(\vec{\phi}) = C \exp\left[-S(\vec{\phi}) \right]$
  - The two-point correlation function can be estimated as:

$$\left\langle \phi_i \phi_j \right\rangle \approx \frac{1}{N} \sum_{k=1}^{N} \phi_i^{(k)} \phi_j^{(k)}$$

  - The error behaves as 1/Sqrt(N).
  - Note: the dimension of the integral/ $\vec{\phi}$ is $\propto (\text{Lattice sites} = 16^4 \text{ for ex.})$

How to generate such an ensemble ?

- Markov Chain Monte Carlo (MCMC)

(general description)

  - A simple random sampling generation is not effective as seen before.
  - A non-random generation is required.

– MCMC Set up. There exisit

$$\vec{\phi}: \text{ random variable}$$

$$\left\{\vec{\phi}^{(1)}, \vec{\phi}^{(2)}, \vec{\phi}^{(3)}, \cdots, \vec{\phi}^{(t-1)}\right\}: \quad t\text{-1 step sequence generated.}$$

– MCMC adds a new sample to the sequence as

Generate $\vec{\phi}^{(t)}$ with a probability distribution $P(\vec{\phi}^{(t)} \mid \vec{\phi}^{(t-1)})$.

Then add the new sample to the sequence.

$$\left\{\vec{\phi}^{(1)}, \vec{\phi}^{(2)}, \vec{\phi}^{(3)}, \cdots, \vec{\phi}^{(t-1)}, \vec{\phi}^{(t)}\right\}: \quad t \text{ step sequence generated.}$$

Where

$$P(\vec{\phi}^{(t)} \mid \vec{\phi}^{(t-1)}): \quad \text{Transition probability } \vec{\phi}^{(t-1)} \to \vec{\phi}^{(t)} \text{ in MCMC.}$$

- Markov Chain Monte Carlo (MCMC) cont'd
- How to generate the desired distribution from the transition *P?*
  - **Perron-Frobenius theorem.**
    - *P*(ϕ|ϕ') transition probability can be treated as a matrix element which index takes a value of state number.

$$\vec{\phi} : \mathrm{state} \to i : i\text{-th}\,\mathrm{state}, \quad P(\vec{\phi}\,|\,\vec{\phi}') \to P_{ij}$$

$$: \ \mathrm{transition\ probability\ from\ j\text{-}th\ state\ to\ i\text{-}th\ state.}$$

    - The matrix *P* satisfies

$$P_{ij} > 0, \qquad \text{Real positive Probability.}$$

$$\sum_i P_{ij} = 1, \qquad \text{Probability conservation.}$$

    - *P* is called a positive matrix.
  - **Perron-Frobenius theorem:**
    - Any positive real matrix has a unique and largest eigenvalue (with =1), and associated eigenvector with positive components.

$$Pw = w, \qquad w_i > 0$$

- Markov Chain Monte Carlo (MCMC) cont'd
- Using Perron-Frobenius theorem, we have
  - For a given initial state:

    $v^{(0)}$: initial distribution.

    if the system is in i-th state, $v^{(0)}{}_i = 1$, and other components are zero.
  - k-step MCMC corresponds to

$$v^{(0)} = Pv^{(1)}, v^{(1)} = Pv^{(2)}, \cdots, v^{(k)} = Pv^{(k-1)}$$

$$v^{(k)} = P^k v^{(0)}$$

  - The Perron-Frobenius theorem says that

$$\lim_{k\to\infty} v^{(k)} = \lim_{k\to\infty} P^k v^{(0)} = w, \quad \text{where } \underline{Pw = w}.$$

  - The convergence to the fixed distribution is usually exponential. After many MCMC step the distribution is almost identical to the maximum eigen vector *w*.
  - If s has the desired distribution we can generate the desired sequence.

- Markov Chain Monte Carlo (MCMC) cont'd
  - (1) Generate initial state.
  - (2) If the system is in the *j*-th state, generate *i*-th state with the probability P_{ij}
  - (3) Add the new state to the ensemble.
  - (4) Goto (2)
- Where we assumed that the state is discrete and countable, *P* is a positive matrix.
- Extension to Non-negative matrixes, and continuum state is also possible.
- The property that the existence of a unique real maximam eigenvalue and positive eigenvector of the theorem still holds, but some special properties are required on *P*. Here I omit the details of the extension. (irreducible,…)
- Now the problem to the path-integral is
  - How to construct $P(\vec{\phi}\,|\,\vec{\phi}')$ so that the maximum eigen vector is $w(\vec{\phi}) = C\exp\left(-S(\vec{\phi})\right)$?

2-5 Detailed Balance Condition

- How to construct Transition probability $P(\vec{\phi}\,|\,\vec{\phi}')$ to make a desired distribution $w(\vec{\phi}) = C\exp\left(-S(\vec{\phi})\right)$?
- One sufficient condition is the so called detailed balance condition.
  - Recalling that the fixed point distribution is a eigenvector of the transition probability with real unit eigenvalue.

Eigen equation for transition probability matrix *P*

$$Pw = w \Leftrightarrow \sum_j P_{ij} w_j = w_i \qquad \text{for discrete state space}$$

$$\text{or} \quad \int d\vec{\phi}' P(\vec{\phi}\,|\,\vec{\phi}') w(\vec{\phi}') = w(\vec{\phi}) \quad \text{for continuous state space}$$

- The detailed balance condition requires

(Problem-1)

$$\boxed{P_{ij} w_j = P_{ji} w_i \quad \text{or} \quad P(\vec{\phi}\,|\,\vec{\phi}') w(\vec{\phi}') = P(\vec{\phi}'\,|\,\vec{\phi}) w(\vec{\phi})}$$

This is a sufficient condition for eigenvector *w* with unit eivenvalue.
In this case, *P* is a reversible Markov chain (*w* is a simultaneous left and right evec).

- Some MCMC examples that satisfies the detailed balance condition.
- (1) Metropolis-Hastings algorithm

  (Metropolis et al. 1953, Hasitings 1970)

$$\text{(step 0) Given initial state } j^{(0)}, t = 0$$

$$\text{(setp1) Generate a canditate state } i \text{ with probability } q_{ij^{(t)}}.$$

$$\text{(step 2) Take next state } j^{(t+1)} \text{ as}$$

$$j^{(t+1)} = \begin{cases} i & \text{with probablity} \quad \rho_{ij^{(t)}} \\ j^{(t)} & \text{with probablity } 1-\rho_{ij^{(t)}} \end{cases}$$

$$\text{(step 3) } t \Leftarrow t+1, \text{ goto step1.}$$

- Where

$$\rho_{ij} = \min\left(1, \frac{w_i}{w_j}\right) \quad \text{and} \quad q_{ij} = q_{ji}$$

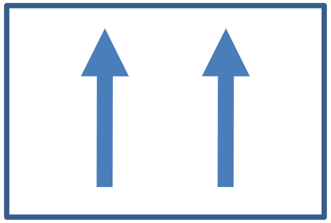
- This algorithm is equivalent to the following transition probability

(Problem-2)

$$P_{ij} = \rho_{ij} q_{ij} + (1 - r_j)\delta_{ij},$$

$$\rho_{ij} = \min\left(1, \frac{w_i}{w_j}\right), \qquad r_j = \sum_k \rho_{kj} q_{kj}$$

- This transition probability matrix satisfies the detailed balance condition.
- A More concrete example for Metropolis algorithm.
  - Ising model with 2 spins.



$$S(\vec{\sigma}) = \beta \sigma_1 \sigma_2$$

$$Z(\vec{\eta}) = \sum_{\sigma_1 = \pm 1, \sigma_1 = \pm 1} \exp\left[-S(\vec{\sigma}) + \vec{\eta} \cdot \vec{\sigma} \right]$$

$$\sigma_1 = \begin{cases} +1 \\ -1 \end{cases} \quad \sigma_2 = \begin{cases} +1 \\ -1 \end{cases}$$

- Ising model with 2 spins.

$$S(\vec{\sigma}) = \beta\sigma_1\sigma_2 \qquad Z(\vec{\eta}) = \sum_{\sigma_1=\pm1,\sigma_1=\pm1} \exp\left[-S(\vec{\sigma}) + \vec{\eta}\cdot\vec{\sigma}\right]$$

- To compute the spin average and spin correlation

$$\langle \bar{\sigma} \rangle = \frac{1}{Z(\vec{0})} \sum_{\sigma_1=\pm1,\sigma_1=\pm1} \frac{\sigma_1 + \sigma_2}{2} \exp\left[-S(\vec{\sigma})\right] = 0$$

$$\langle \sigma_1\sigma_2 \rangle = \frac{1}{Z(\vec{0})} \sum_{\sigma_1=\pm1,\sigma_1=\pm1} \sigma_1\sigma_2 \exp\left[-S(\vec{\sigma})\right] = -\tanh(\beta)$$

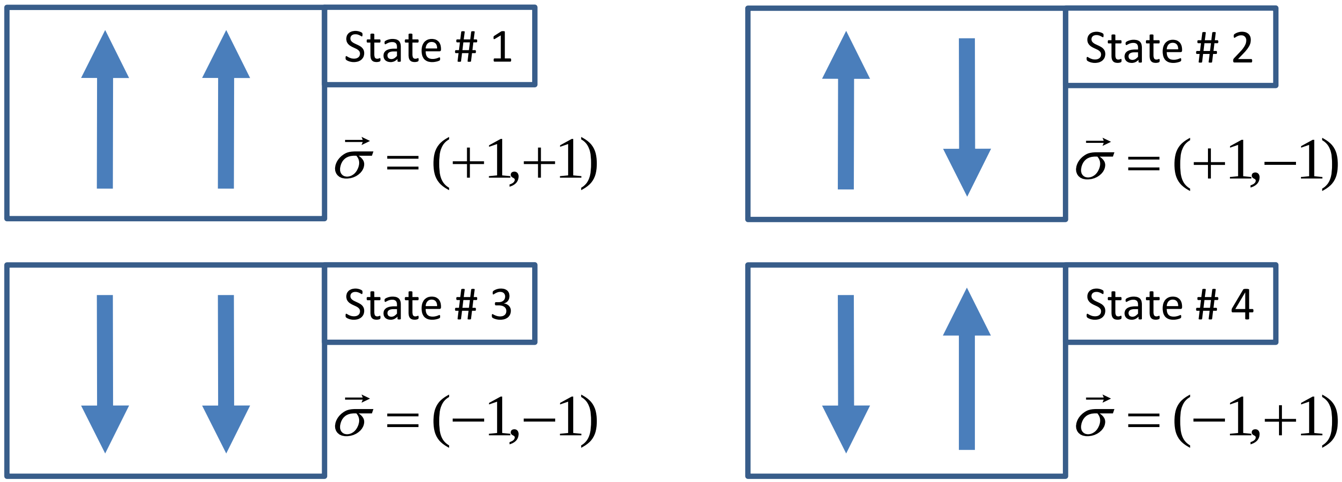
- We generate the ensemble $\left\{\vec{\sigma}^{(0)}, \vec{\sigma}^{(1)}, \vec{\sigma}^{(2)}, \cdots, \vec{\sigma}^{(N)}\right\}$ with the distribution

$$w(\vec{\sigma}) = \frac{1}{Z(\vec{0})} \exp\left[-S(\vec{\sigma})\right]$$

- Then we can estimate the squared spin average by

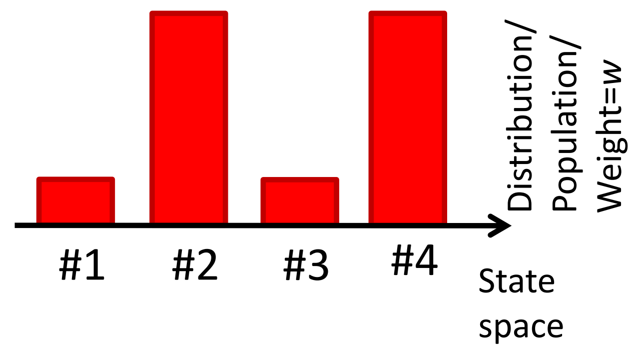
$$\langle \bar{\sigma} \rangle \approx \frac{1}{N} \sum_{j=1}^{N} \frac{\sigma_1^{(j)} + \sigma_2^{(j)}}{2}, \qquad \langle \sigma_1\sigma_2 \rangle \approx \frac{1}{N} \sum_{j=1}^{N} \sigma_1^{(j)}\sigma_2^{(j)}$$

- We have only 4 states.



- The weight(probability) is calculable (*C* is the normalization const = *Z*(0))

$$w(\#1) = C \exp[-\beta], \quad w(\#2) = C \exp[+\beta]$$
$$w(\#3) = C \exp[-\beta], \quad w(\#4) = C \exp[+\beta]$$



- We can generate this distribution with the Metropolis Algorithm

- Metropolis algorithm for the Ising model with 2-spins.

(step 0) Randomly choose initial state $\vec{\sigma}^{(t)}, t=0$

(setp 1) Generate a canditate state $\vec{s}$ with probability $q_{ij^{(t)}} = 1/4$

$\boxed{\begin{array}{l} \text{(step 2) Compute the weight } \rho = \min(1, \exp\left[-S(\vec{s}) + S(\vec{\sigma}^{(t)})\right]) \\ \text{(Step 3) Generate a random real number } U \text{ from } [0,1). \\ \text{(step 4) Take next state } \vec{\sigma}^{(t+1)} \text{ as} \\ \qquad \vec{\sigma}^{(t+1)} = \begin{cases} \vec{s} & \text{when } U \le \rho \text{ (Accept)} \\ \vec{\sigma}^{(t)} & \text{otherwise (Reject)} \end{cases} \end{array}}$

(step 5) $t \Leftarrow t+1$, goto step 1 for desired sample numbers.

- Then we obtain ensemble:

$$\left\{\vec{\sigma}^{(0)}, \vec{\sigma}^{(1)}, \vec{\sigma}^{(2)}, \cdots, \vec{\sigma}^{(N)}\right\}$$

Metropolis test
Metropolis accept/reject step

- Corresponding Fortran Program:
  - [http://theo.phys.sci.hiroshima-u.ac.jp/~ishikawa/ASLFT2010/2SiteIsingMetropolis.tar.gz]

```fortran
use ising_module
implicit none
integer ::  NTHERM      ! first NTHERM samples are dropped
integer :: NSAMPLE      ! measured sample number
integer, allocatable :: seed(:)
integer :: iseed,rand_size

integer :: s0(2)   ! previous state (2-spins)
integer :: s1(2)   !  current state (2-spins)
integer :: j0      ! previous state index
integer :: j1      !  current state index
integer :: istep
real(DP) :: rand_num
real(DP) :: beta
real(DP) :: rho,w0,w1,h0,h1
real(DP) :: spinave,spincorr
integer :: iout

iout=99
open(iout,file="ISING_PARAM",status='old',form='formatted')
read(iout,*)beta
read(iout,*)iseed
read(iout,*)NTHERM,NSAMPLE
close(iout)


!
! Set up pseudo-random number generator
!
call RANDOM_SEED(size=rand_size)
allocate(seed(rand_size))
seed(:) = iseed
call RANDOM_SEED(put=seed)

write(*,'("# BETA=",ES24.15)') beta

write(*,'("# ISEED=",I10," NTHERM=",I10," NSAMPLE=",I10)') &
&           iseed,NTHERM,NSAMPLE

write(*,'("# sample#  state index  spin state",&
&          10X,"spin ave",14X,"spin corr")')

!
! Generate initial state at random
!
call RANDOM_NUMBER(rand_num)
j0 = get_state_index(rand_num)
call set_state(s0,j0)

do istep=1,NTHERM+NSAMPLE
```

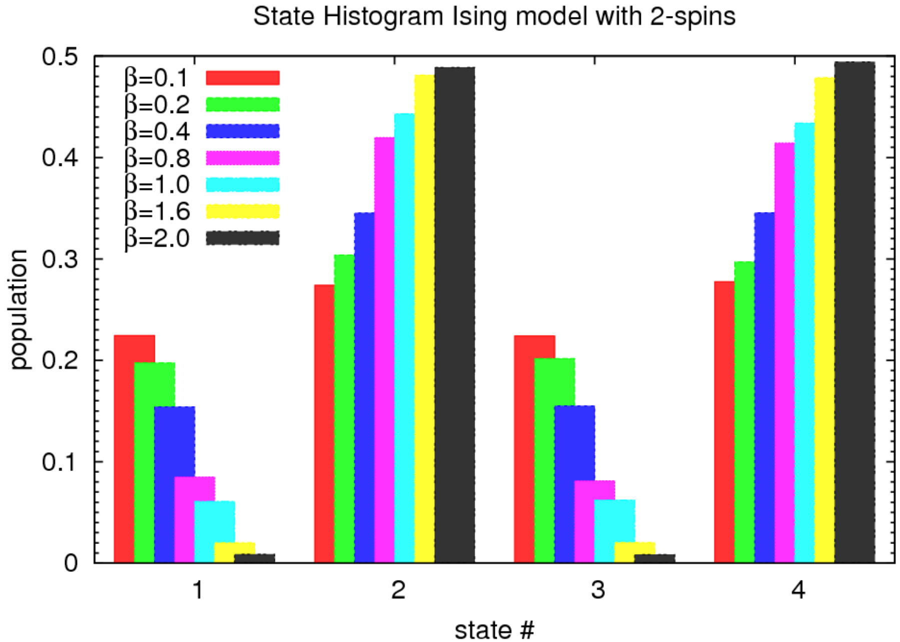
Candidate generation

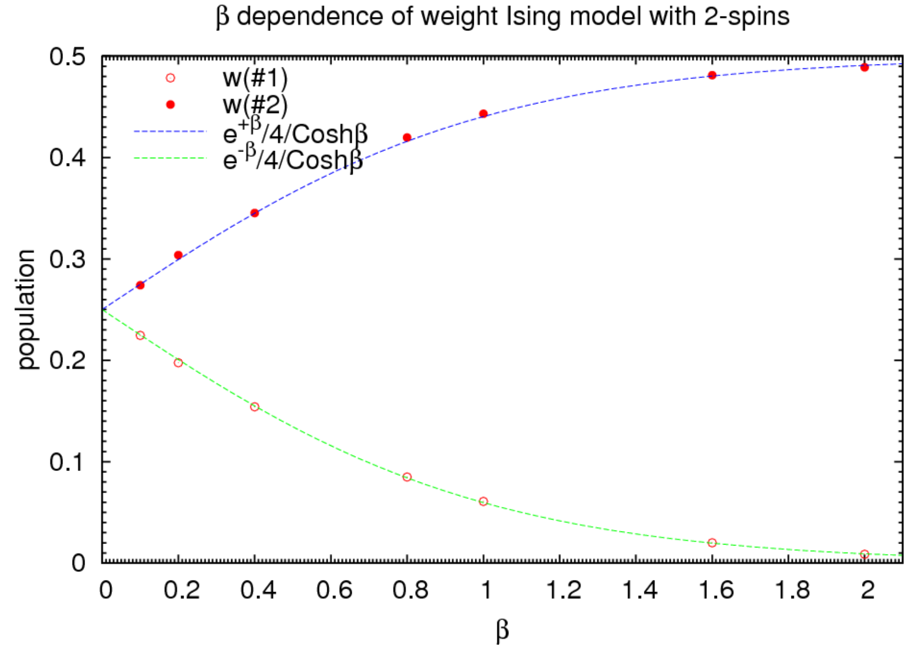
```fortran
  !
  ! Generate candidate state at random
  !
  call RANDOM_NUMBER(rand_num)
  j1 = get_state_index(rand_num)
  call set_state(s1,j1)
```

Metropolis test

```fortran
  !
  ! Compute Metropolis test weight
  !
  h0 = hamil(beta,s0)
  h1 = hamil(beta,s1)
  w0 = exp(-h0)
  w1 = exp(-h1)
  rho = MIN(1.0_DP,w1/w0)

  !
  ! Metropolis test Accept Reject step
  !
  call RANDOM_NUMBER(rand_num)
  if (rand_num <= rho) then

    ! accept s1 as the new state

    continue

  else

    ! reject s1. s0 is the new state

    s1(:) = s0(:)
    j1    = j0

  endif
  if (istep > NTHERM) then
    !
    ! store current state in the ensemble
    ! and measure observables
    !

    spinave = (s1(1) + s1(2))*0.5_DP
    spincorr = s1(1)*s1(2)

    write(*,'(I10,I10,SP,6X,"(",I2,",",I2,")",2ES24.15)')&
&               istep,j1,s1(1),s1(2), spinave, spincorr

  endif

  !
  ! shift history
  !
  s0(:) = s1(:)
  j0    = j1

enddo

deallocate(seed)
stop
end program
```

- Results
  - Weight histogram
  - Beta dependence of Weight





$$Z = \exp[-\beta] + \exp[+\beta] + \exp[-\beta] + \exp[+\beta] = 4\mathrm{Cosh}(\beta)$$

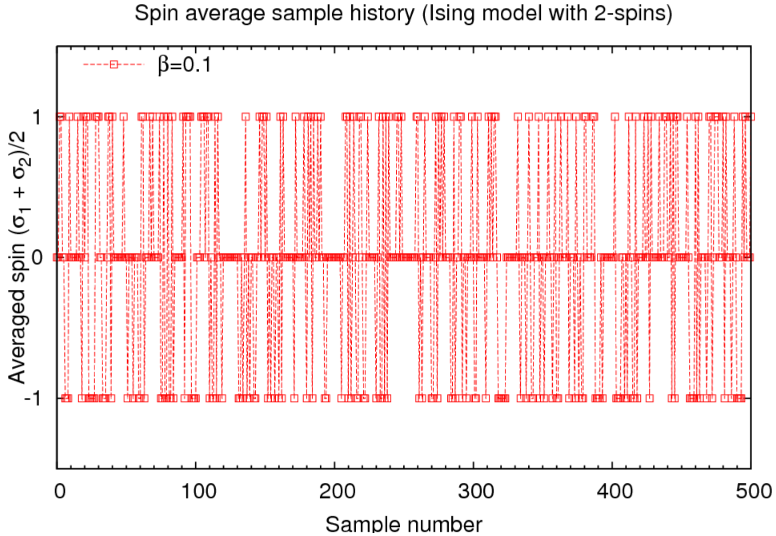
$$w(\#1) = \exp[-\beta]/4/\mathrm{Cosh}(\beta)$$

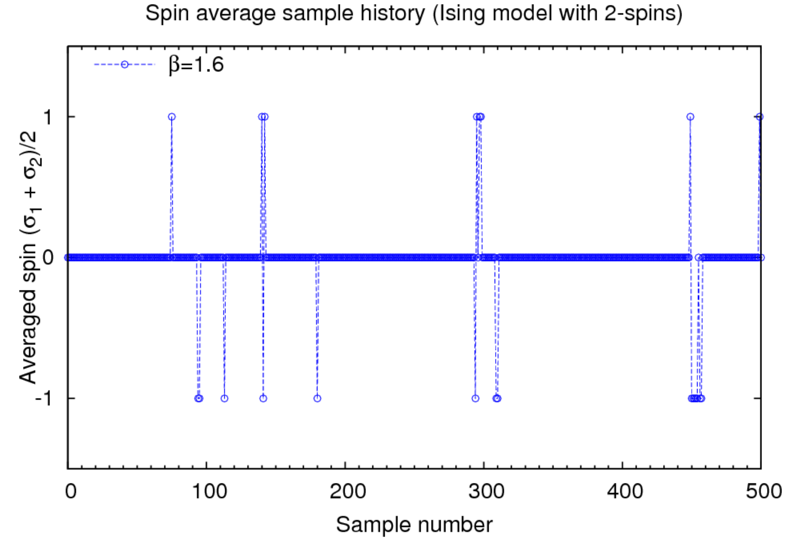
$$w(\#2) = \exp[+\beta]/4/\mathrm{Cosh}(\beta)$$

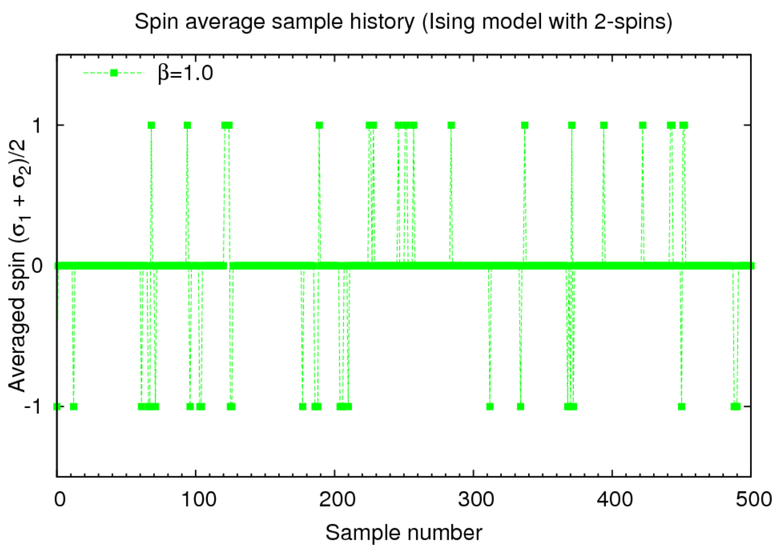
- # Results

$$\frac{\sigma_1^{(j)} + \sigma_2^{(j)}}{2}$$

  - Spin average ensemble history



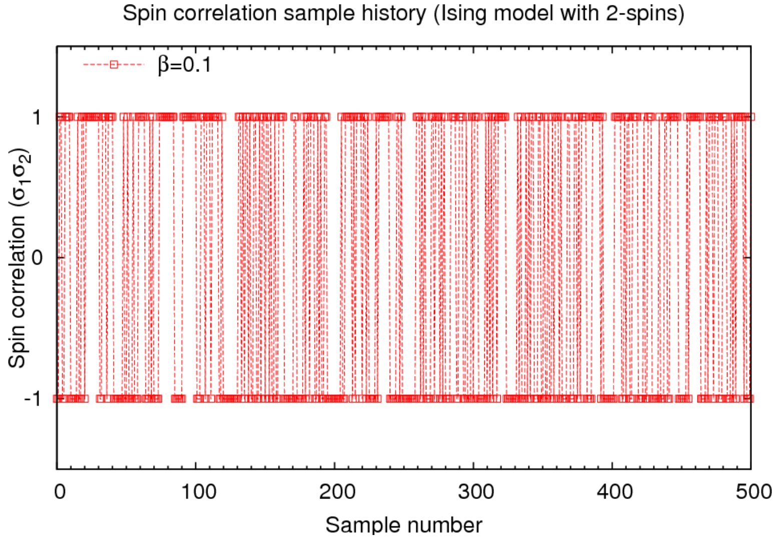


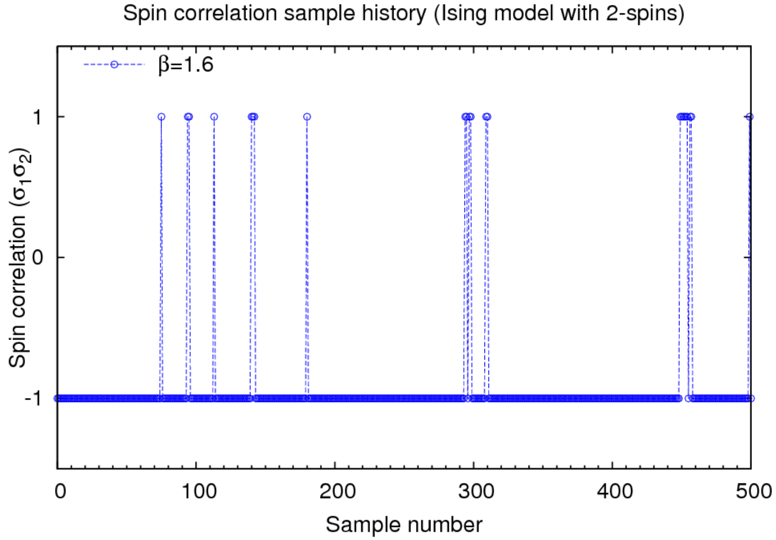


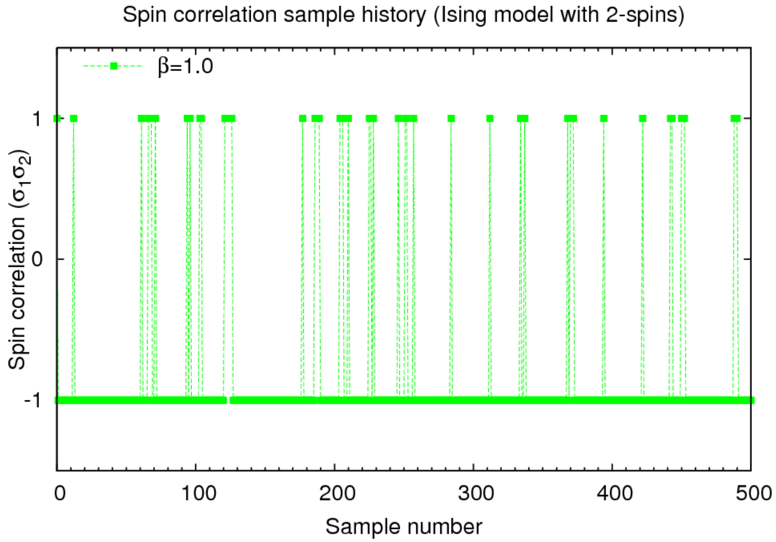
- First 500 samples are plotted.
- Random walking in the state space (4states)
- Spin average can take one of the values (-1,0,1)
- (spin average)=0 can occur for state #2 and #4.
- (spin average)=+1 occurs for state #1.
- (spin average)=-1 occurs for state #3.
- As increasing beta, the state stays at state #2 or #4. spin average = 0 states.

- # Results
  - Spin correlation ensemble history

$$\sigma_1^{(j)}\sigma_2^{(j)}$$

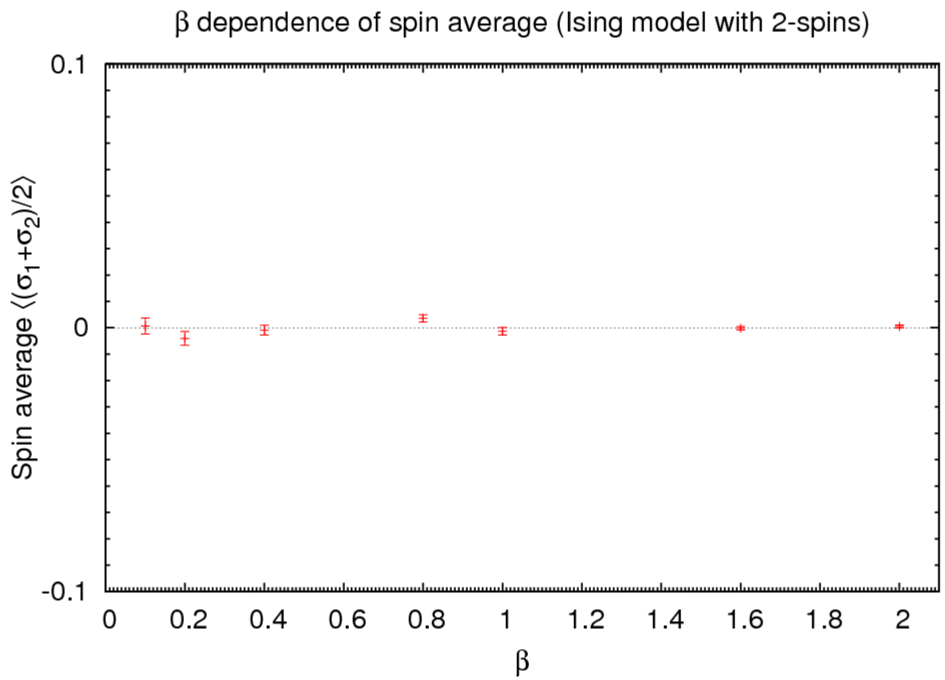






- First 500 samples are plotted.
- Random walking in the state space (4states)
- Spin corr. can take +1 or -1.
- (spin corr.)=+1 can occur for state #1 and #3.
- (spin corr)= -1 occurs for state #2 and #4.
- At small beta population of +1 and -1 is almost same.
- As increasing beta, state with (spin corr.)=-1 dominates. (state #2 and #4)

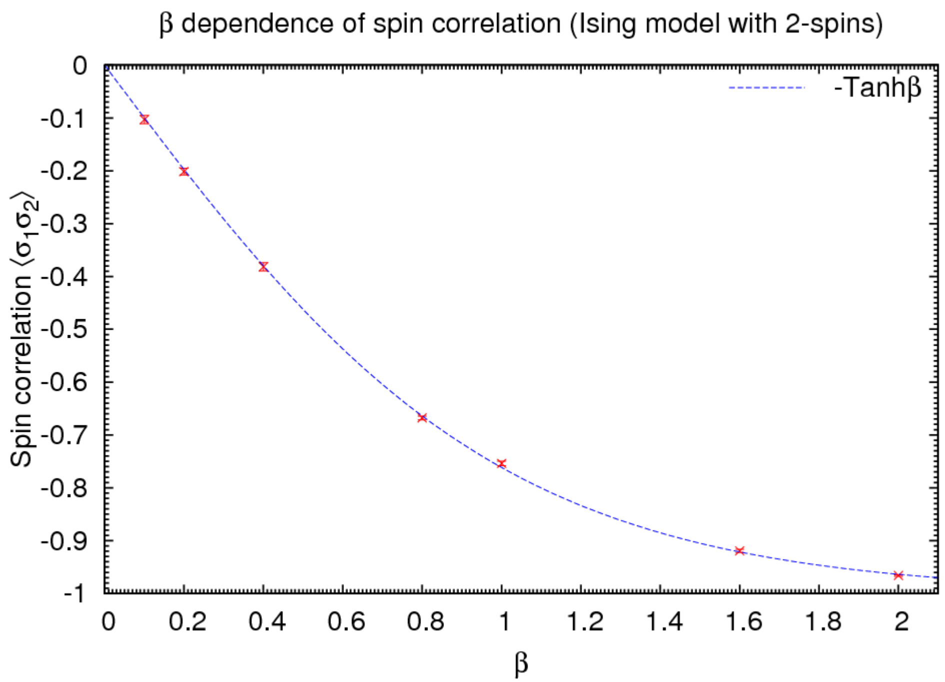
- Results
  - Spin average expectation value



Averaging the history data we obtain zero. This is consistent with the theoretical one.

$$\langle \bar{\sigma} \rangle \approx \frac{1}{N} \sum_{j=1}^{N} \frac{\sigma_1^{(j)} + \sigma_2^{(j)}}{2} \approx 0$$

- Results
  - Spin correration expectation value



Averaging the history data we obtain –Tanh(Beta). This is consistent with the theoretical one.

$$\langle \sigma_1 \sigma_2 \rangle = \frac{+e^{\beta} - e^{-\beta} + e^{\beta} - e^{-\beta}}{4\mathrm{Cosh}\beta} = -\mathrm{Tanh}\beta \approx \frac{1}{N}\sum_{j=1}^{N} \sigma_1^{(j)} \sigma_2^{(j)}$$

- # Difficulties in (Naive) Metropolis Algorithm
  - As seen before the new candidate state(configuration) is really added when the Metropolis test accept.
  - In Statistical Mechanics (Canonical ensemble), the exponent of the weight is the energy of the target system.
  - The acceptance ratio is governed by the Energy difference

$$\Delta S = S([\text{Candidate State}]) - S([\text{Previous State}])$$

$$\rho = \min(1, \exp\left[-S([\text{Candidate State}]) + S([\text{Previous State}])\right])$$
$$= \min(1, \exp\left[-\Delta S\right])$$

  - When ΔS is negative, Metropolis test always accept the candidate ($\rho = 1$).
  - When ΔS is positive, the acceptance probability decreases as $\rho = \exp(-\Delta S) < 1$.
  - When the target system has a huge number of d.o.f., the random sampling method to generate the candidate state almost always large positive number for ΔS. This is typical in statistical mechanics and huge multiple dimension integration.
  - Candidate generation method with small energy difference is important.
  - See also 2D-Ising model. (Heat-bath (Gibbs sampler) algorithm)

- Most of MCMC algorithms make use of the Metropolis algorithm and its extension.
  - For LQCD, the system has continuous variables (states).
  - Naïve Metropoils algorithm may fail due to the large energy differece.
- (2)Hybrid Monte Carlo (HMC) algorithm
  (Scalatar, Scalapino, Sugar, PRB34(1986); Duane, Kennedy Pendleton, Roweth, PL195B(1987))
  - This algorithm is useful when the variables are continuous.
  - This is an extension of the Metropolis algorithm with better candidate generation.
  - The HMC algorithm is a de fact standard algorithm for LQCD with dynamical quarks.
  - In the next lecture I will describe the details of the HMC algorithm.

Problems

(1) Check that the detailed balance condition is a sufficient condition of the eigenvector (stationary distribution) of the transition matrix. [page 22]

(2) Check that the transition matrix for the Metropolis algorithm satisfies the detailed balance condition. [page 24].

(3) Complete the transition matrix for the Ising model with 2-spins in a 4x4 matrix form and Check the eigenvector. [page 24-33]

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{34} & P_{44} \end{pmatrix}$$

(4) Get and compile the Ising model with 2-spins program. Check the result numerically. [page 24-33] (this needs gfortran and gnuplot on Linux)

(5) Evaluate the averaged acceptance rate of the Metropolis test when the energy difference is a random variable from the Gaussian distribution with mean= μ and variance= $\sigma^2 = 2\mu$. [Hint: Complementary error function]

$$p(\Delta S) = \frac{1}{\sqrt{4\pi\mu}} \exp\left(-\frac{(\Delta S - \mu)^2}{4\mu} \right)$$

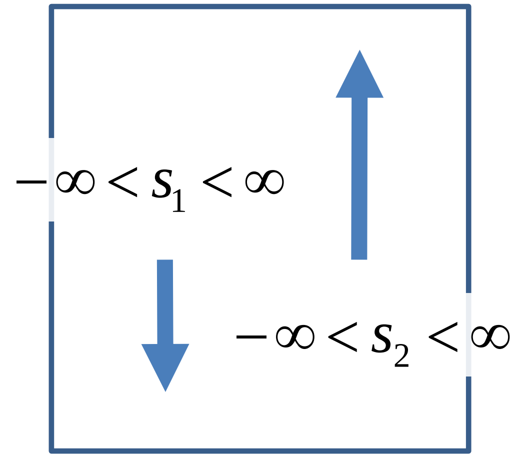
(6) [Advanced] Consider a N-sites 1D Ising model with periodic boundary condition.

Backup (2-site Scalar model)

- For a continuous state model. I show the 2-site scalar model. (a toy model for lattice scalar field theories)

$$\beta < 1$$

$$Z(\vec{\eta}) = \int_{-\infty}^{\infty} d\vec{s} \exp\left[-\left(\beta s_1 s_2 + \frac{s_1{}^2 + s_2{}^2}{2} \right) + \vec{\eta} \cdot \vec{s} \right]$$



$$\left\langle \frac{s_1 + s_2}{2} \right\rangle = \frac{1}{Z(\vec{0})} \int_{-\infty}^{\infty} d\vec{s} \, \frac{s_1 + s_2}{2} \exp\left[-\left(\beta s_1 s_2 + \frac{s_1{}^2 + s_2{}^2}{2} \right) \right] = 0$$

$$\langle s_1 s_2 \rangle = \frac{1}{Z(\vec{0})} \int_{-\infty}^{\infty} d\vec{s} \, (s_1 s_2) \exp\left[-\left(\beta s_1 s_2 + \frac{s_1{}^2 + s_2{}^2}{2} \right) \right] = -\frac{\beta}{1 - \beta^2}$$

• Metropolis algorithm

(step 0) initial state $\vec{s}^{(t)}, t = 0$ from Gaussian Distribution $\mathrm{N}[0, \mathrm{var}]$

(setp 1) Generate a canditate state $\vec{s}$ from Gaussian Distribution $\mathrm{N}[\vec{s}^{(t)}, \mathrm{var}]$

$$\text{This corresponds to } q(\vec{s}\,|\,\vec{s}^{(t)}) = \frac{1}{2\pi\,\mathrm{var}} \exp\left[-\frac{(\vec{s}-\vec{s}^{(t)})^2}{2\,\mathrm{var}}\right].$$

(step 2) Compute the weight $\rho = \min(1, \exp\left[-S(\vec{s}) + S(\vec{s}^{(t)})\right])$

(Step 3) Generate a random real number U from $[0,1)$.

(step 4) Take next state $\vec{s}^{(t+1)}$ as

$$\vec{s}^{(t+1)} = \begin{cases} \vec{s} & \text{when } U \le \rho \ \text{(Accept)} \\ \vec{s}^{(t)} & \text{otherwise (Reject)} \end{cases}$$

(step 5) $t \Leftarrow t + 1$, goto step 1 for desired sample numbers.

- We do not have a finite distribution at beta=1 with this model.
- We can not use uniform sampling for candidate generation because $-\infty < \vec{s} < \infty$.

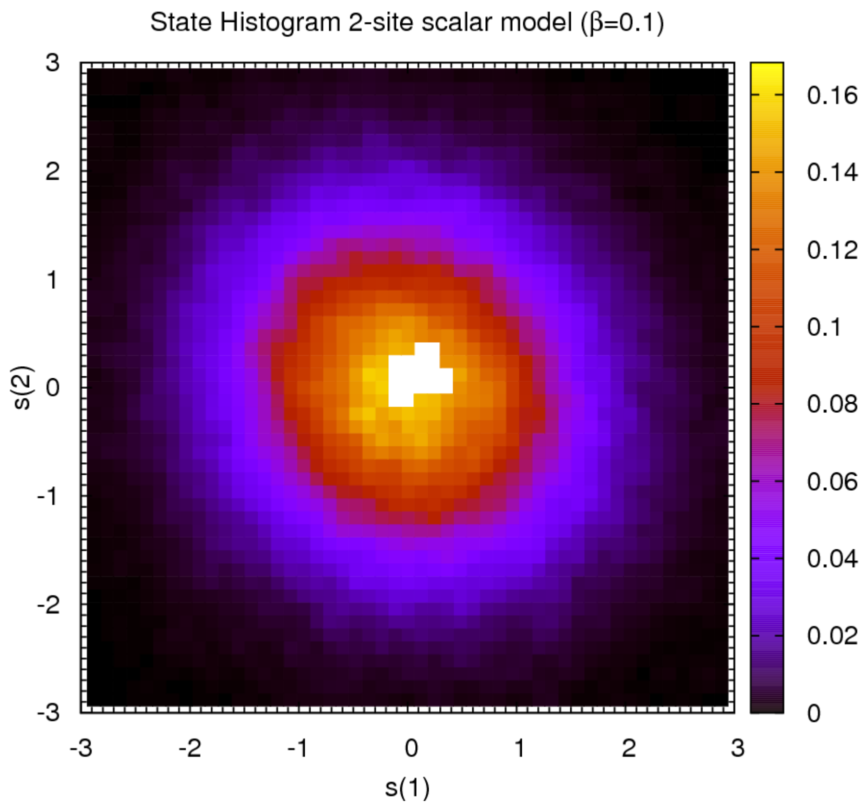
- Fortran program:

  [http://theo.phys.sci.hiroshima-u.ac.jp/~ishikawa/ASLFT2010/2SiteScalarMetropolis.tar.gz]

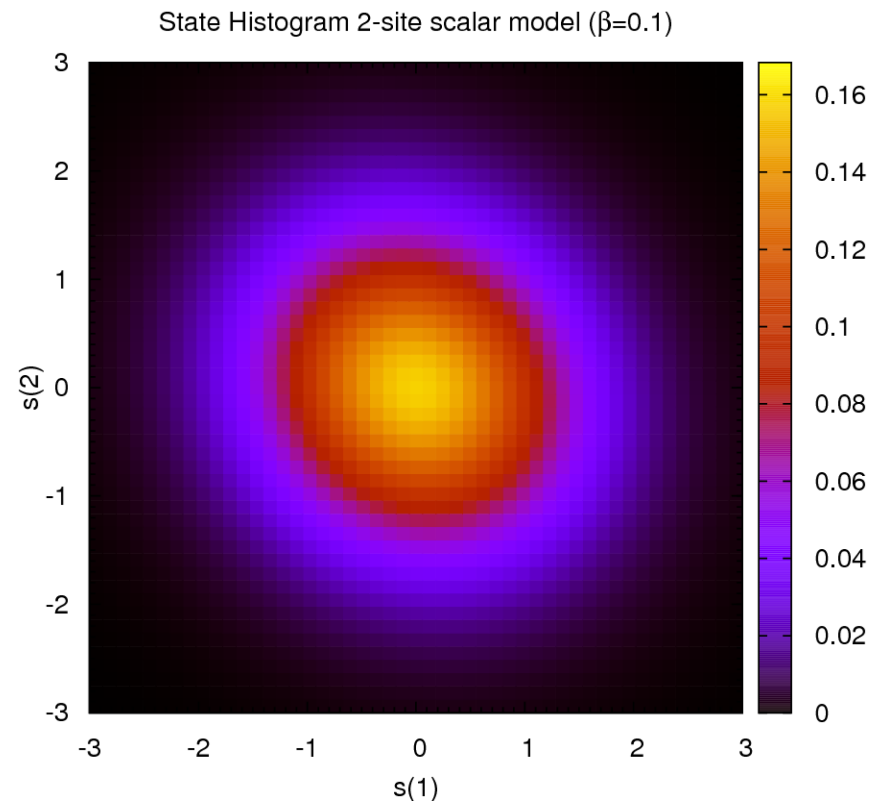
  – 10,000,000 samples are generated. But we save 10,000 samples with interval 100. We use var=1 for candidate generation.

- State weight/histogram generated via Metropolis algorithm

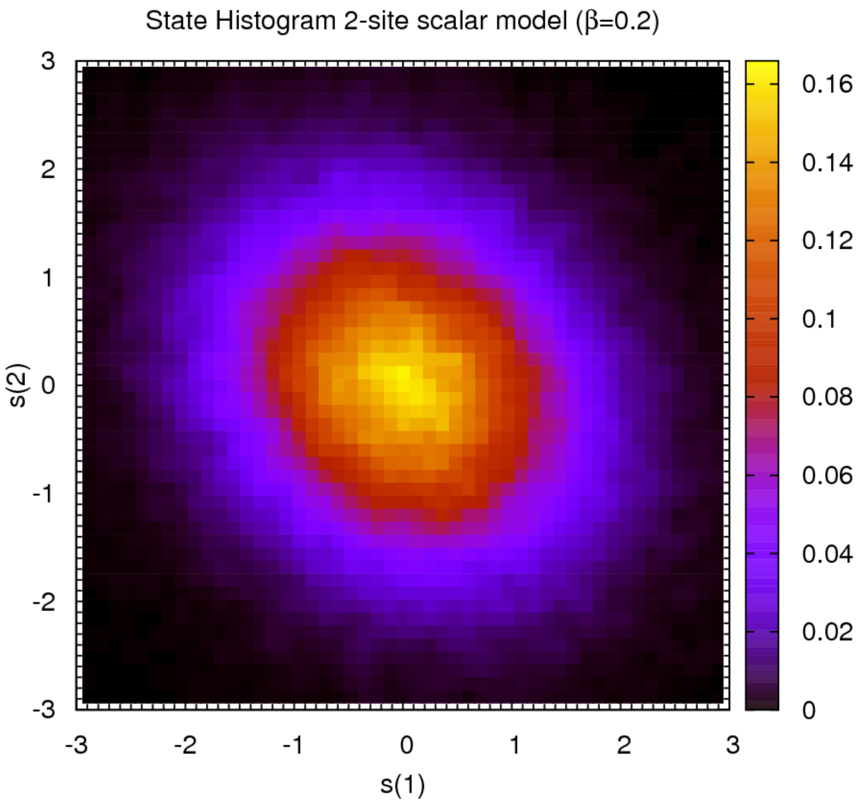
Measured

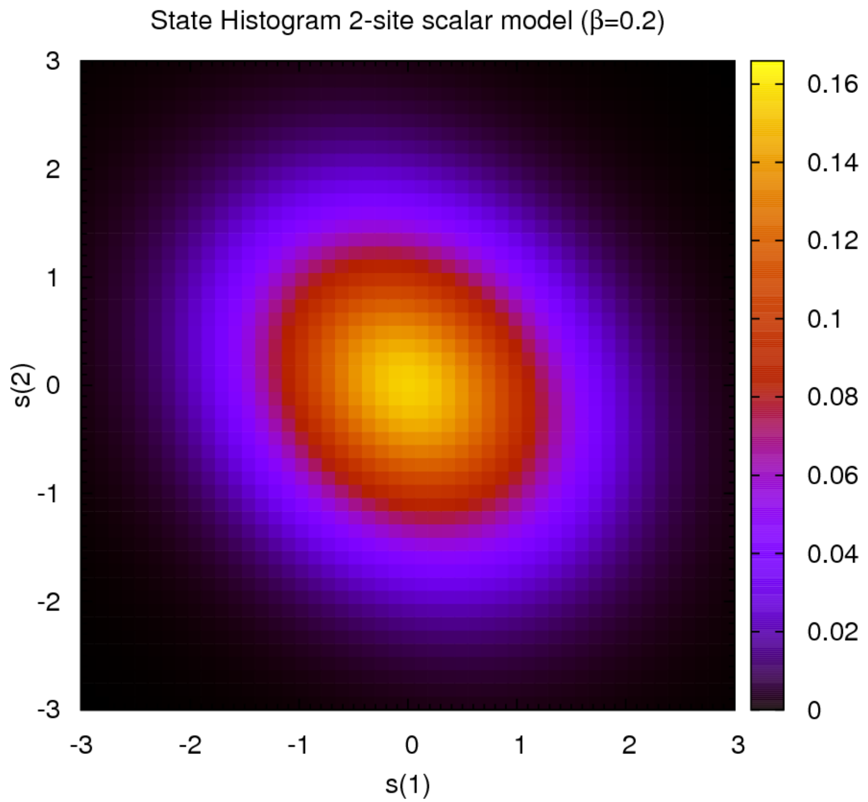


Theoretical

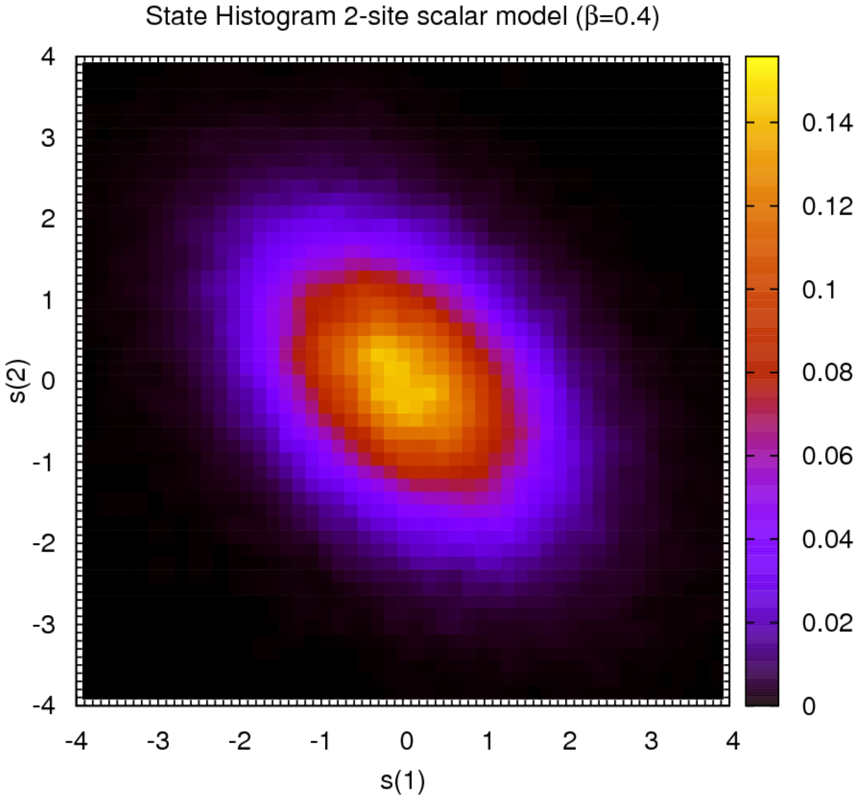


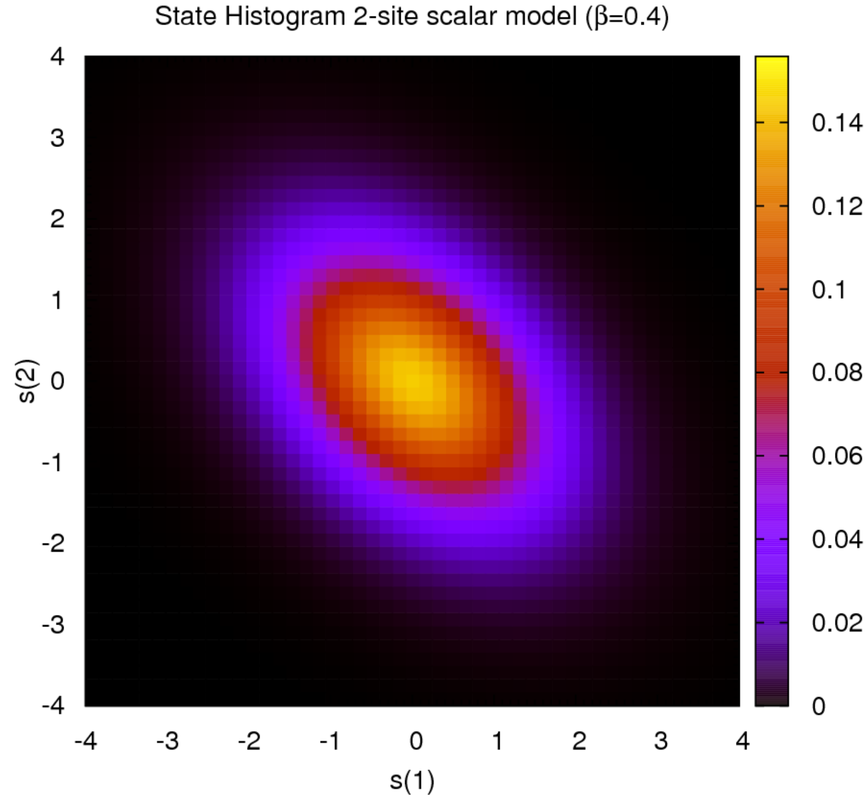
- State weight/histogram generated via Metropolis algorithm



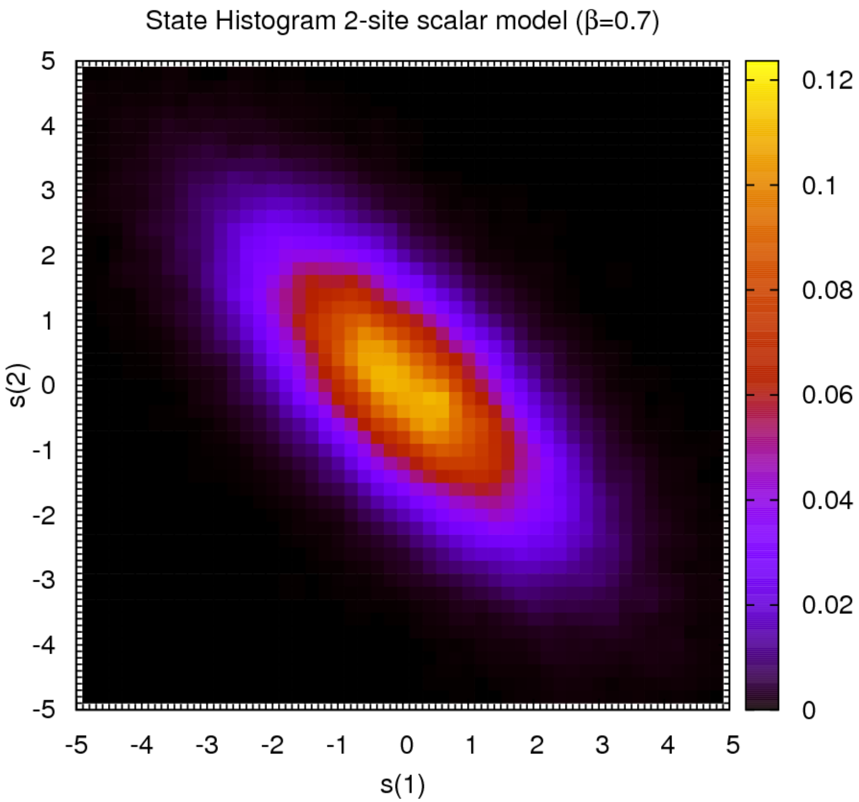


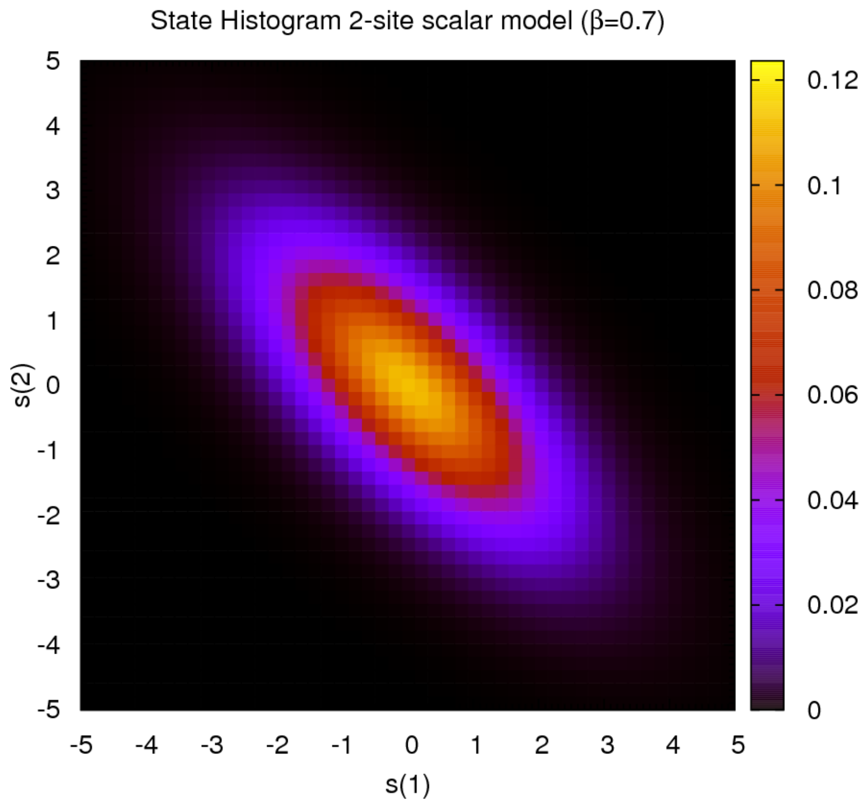
- State weight/histogram generated via Metropolis algorithm



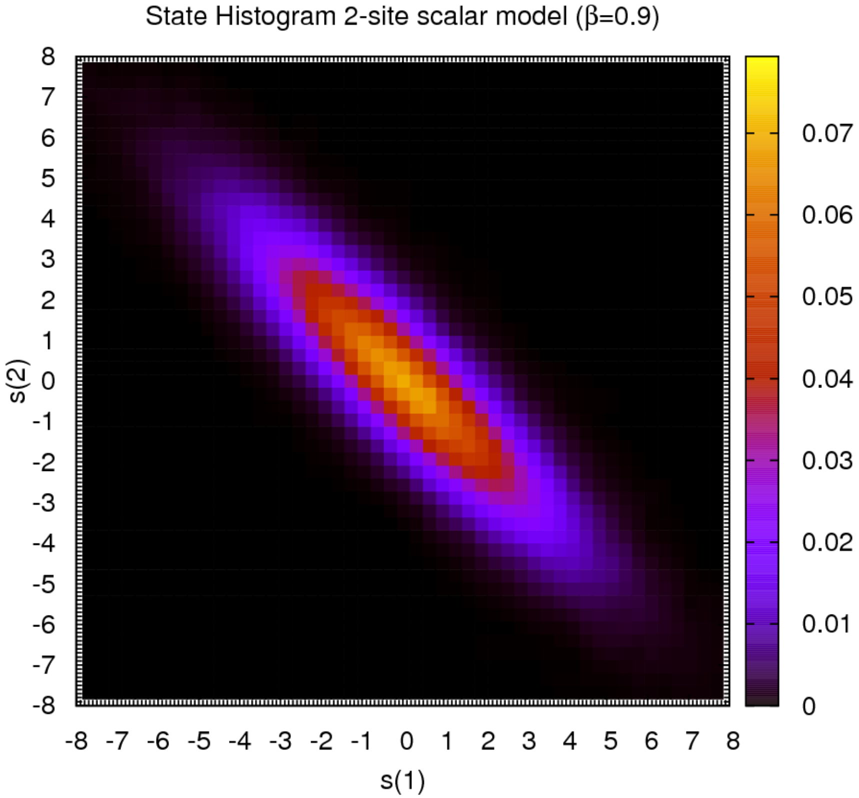


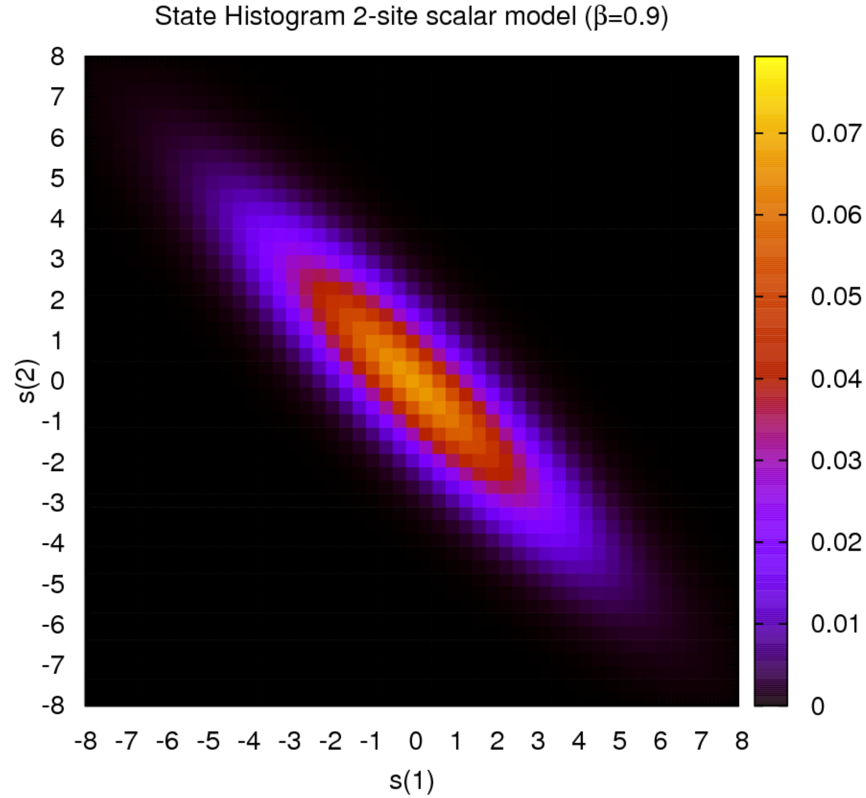
- State weight/histogram generated via Metropolis algorithm





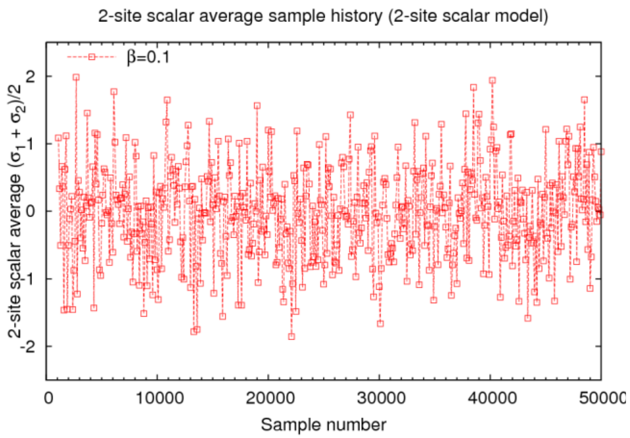
- State weight/histogram generated via Metropolis algorithm

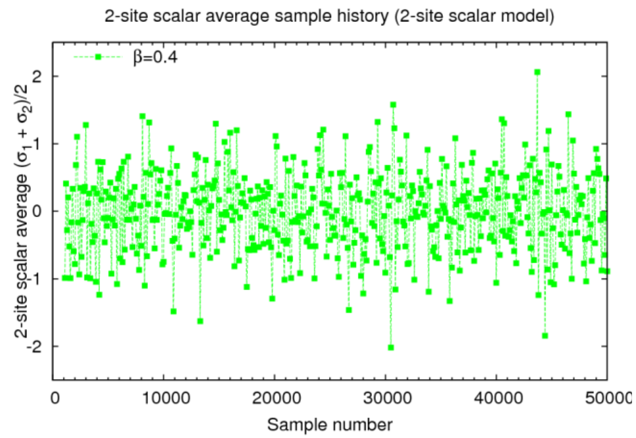


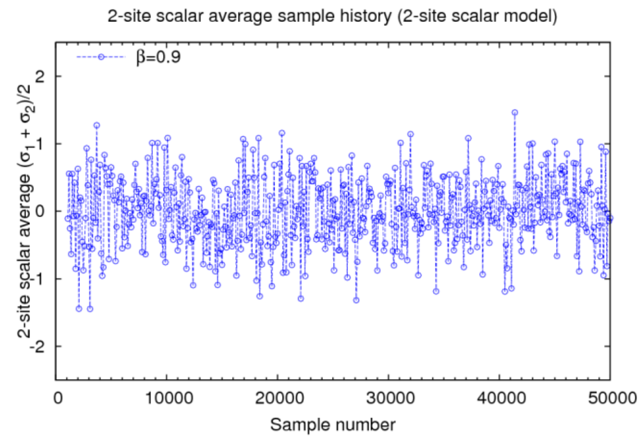


- Spin average and Spin correlation history generated via Metropolis algorithm

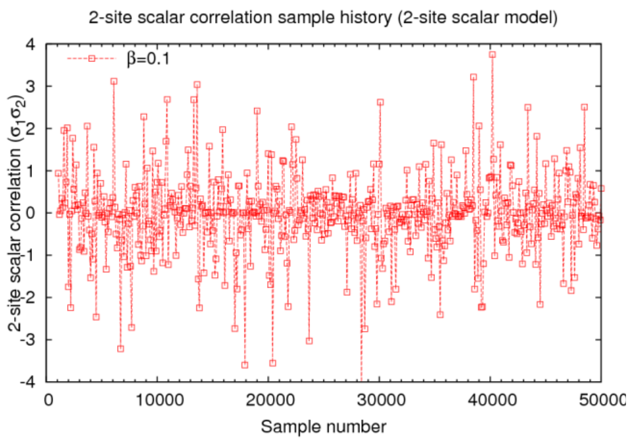
Spin average

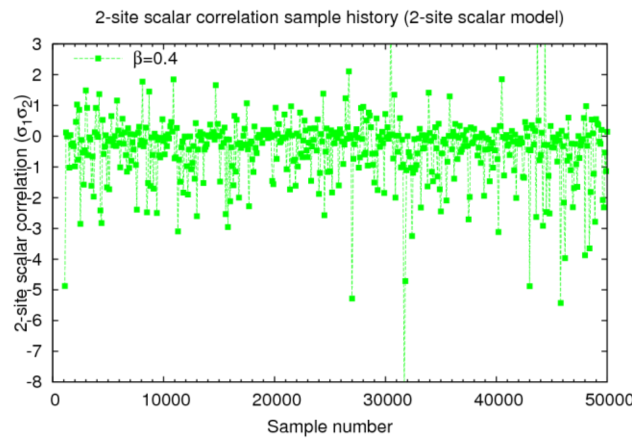


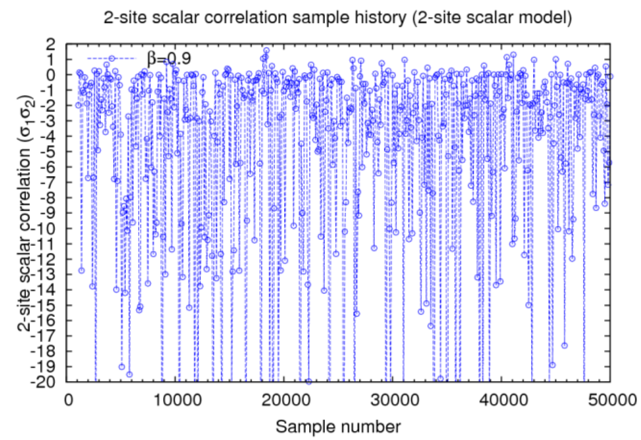




Spin correlation

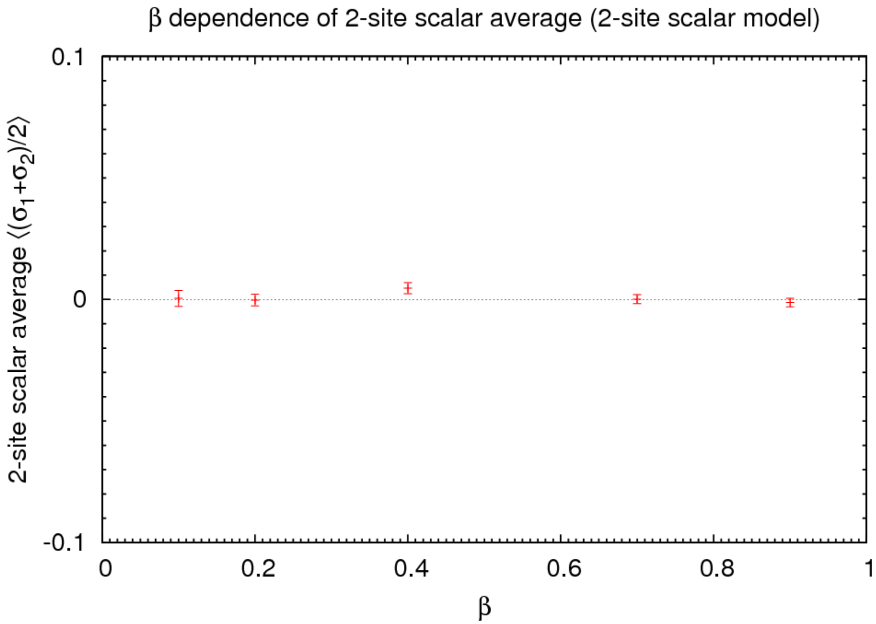




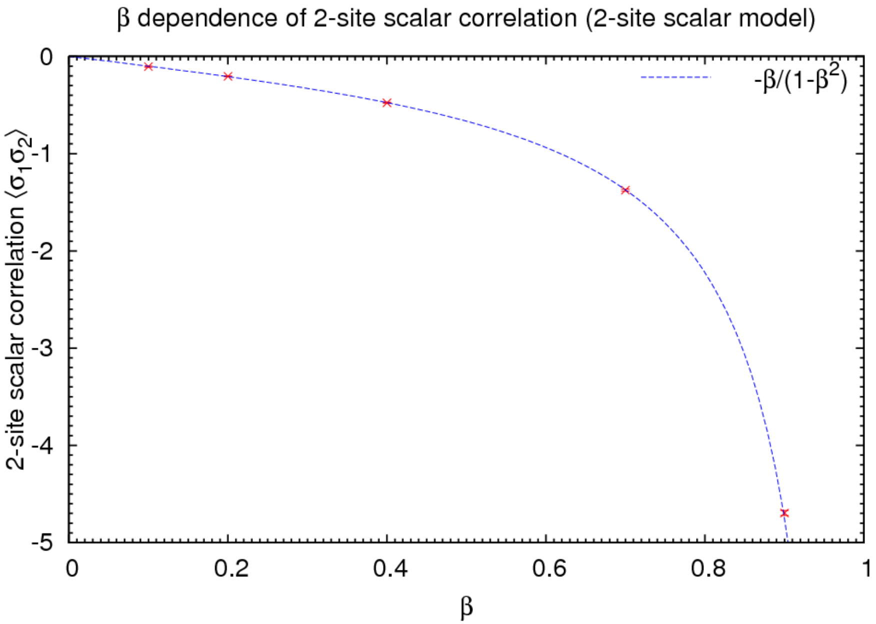


- Beta dependence of Spin average and Spin corr.

Spin average



Spin correlation



All programs are NO WARRANTY.

- Metropolis algorithm transition probability for $\vec{s}\,' \to \vec{s}$

$$P(\vec{s}\,|\,\vec{s}\,') = \rho(\vec{s},\vec{s}\,')q(\vec{s}\,|\,\vec{s}\,') + (1 - r(\vec{s}\,'))\delta(\vec{s} - \vec{s}\,')$$

$$\rho(\vec{s},\vec{s}\,') = \min(1, \exp\left[-S(\vec{s}) + S(\vec{s}\,')\right]), q(\vec{s}\,|\,\vec{s}\,') = \frac{1}{2\pi\,\mathrm{var}} \exp\left[-\frac{(\vec{s} - \vec{s}\,')^2}{2\,\mathrm{var}}\right]$$

$$r(\vec{s}\,') = \int_{-\infty}^{\infty} d\vec{s}\,\rho(\vec{s},\vec{s}\,')q(\vec{s}\,|\,\vec{s}\,')$$

$$\delta(\vec{s}) = \text{Delta function}$$

Problem answers

- (1) $$\sum_j P_{ij} w_j = \sum_j P_{ji} w_i \to \sum_j P_{ij} w_j = w_i \sum_j P_{ji}$$
$$\to \sum_j P_{ij} w_j = w_i \quad \text{because} \quad \sum_j P_{ji} = 1$$
- (2)

$$\rho_{ij} = \min\left(1, \frac{w_i}{w_j}\right)$$
$$\text{and} \quad q_{ij} = q_{ji}$$

$$f_{ij} \equiv \log(\frac{w_i}{w_j}) \to \rho_{ij} = \min\left(1, \frac{w_i}{w_j}\right) = \Theta(f_{ij}) + \Theta(-f_{ij})\frac{w_i}{w_j}$$
$$\to f_{ij} = -f_{ji}$$
$$\rho_{ij} w_j = \left(\Theta(f_{ij}) + \Theta(-f_{ij})\frac{w_i}{w_j}\right) w_j = w_j \Theta(f_{ij}) + \Theta(-f_{ij}) w_i$$
$$= \left(\frac{w_j}{w_i}\Theta(f_{ij}) + \Theta(-f_{ij})\right) w_i = \left(\frac{w_j}{w_i}\Theta(-f_{ji}) + \Theta(f_{ji})\right) w_i$$
$$= \rho_{ji} w_i$$

- Similarly $$P_{ij} w_j = P_{ji} w_i$$

- (3) I show beta>0 case only.

$$P=\frac{1}{4}\begin{pmatrix}1 & y & 1 & y\\ 1 & 3-2y & 1 & 1\\ 1 & y & 1 & y\\ 1 & 1 & 1 & 3-2y\end{pmatrix}\quad \text{with}\quad y=e^{-2\beta}\quad (\text{for } \beta>0)$$

- Eigenpairs

$$w_{\lambda=1}=\begin{pmatrix}y\\1\\y\\1\end{pmatrix}\frac{1}{2(2+y)},\quad \text{with}\quad \lambda=1.$$

Desired distribution

$$w_{\lambda_3}=\begin{pmatrix}-1\\1\\-1\\1\end{pmatrix},\quad \text{with}\quad \lambda_3=0$$

$$w_{\lambda_2}=\begin{pmatrix}0\\-1\\0\\1\end{pmatrix},\begin{pmatrix}-1\\0\\1\\0\end{pmatrix}\quad \text{with}\quad \lambda_2=\frac{1-y}{2}<1.$$

- Thus MCMC converges to the desired distribution.

$$\lim_{N\to\infty} P^N v = w_{\lambda=1}$$

The convergence rate is governed by the difference between 1 and next largest eigenvalue.

- (5)
- (complementary) error functions:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \qquad \text{error function.}$$

$$\mathrm{erfc}(x) = 1 - \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \quad \text{complementary error function.}$$

- Averaging acceptance probability:

$$\left\langle P_{acc} \right\rangle = \int_{-\infty}^{\infty} \min(1, e^{-\Delta S}) \frac{1}{\sqrt{4\pi\mu}} e^{-\frac{(\Delta S - \mu)^2}{4\mu}} = \mathrm{erfc}\left(\frac{\sqrt{\Delta S}}{2} \right)$$

History

- 2013/01/31: Metropolis test probability is corrected.

$$\rho_{ij} = \min\left(1, \frac{w_j}{w_i}\right)$$

and $q_{ij} = q_{ji}$

Wrong

$$\rho_{ij} = \min\left(1, \frac{w_i}{w_j}\right)$$

and $q_{ij} = q_{ji}$

Right