

# QFT Project 1 Lectures

## Project 1

(Marks 15)

In this project you will construct a Monte Carlo algorithm and develop computer codes to study a  $d$  dimensional scalar field theory on the lattice. In the next project you will use these codes to study the theory in  $d = 2, 3$  and  $4$  dimensions.

The theory you will study is defined by the lattice action

$$S_E^\ell = \sum_x \left\{ \frac{1}{2} \chi_x^2 - \kappa \sum_\mu \chi_x \chi_{x+\hat{\mu}} + g \chi_x^4 \right\},$$

where  $\kappa$  and  $g$  are called “bare” couplings. We will define  $\kappa^{-1} = 2d + (m_0 a)^2$ , where  $m_0$  is defined as the bare lattice mass term and  $a$  is the lattice spacing. When  $g = 0$  we can verify that the physical mass of the scalar particle is  $M_{\text{phys}} = m_0$ , however once we turn on interactions this connection between  $M_{\text{phys}}$  and  $m_0$  is no longer true and we will treat  $\alpha \equiv (m_0 a)^2$  as a parameter that defines  $\kappa$ .

The label  $x = (\mathbf{n}, \tau)$  represents the site on a square, cubic, or hyper-cubic lattice depending on  $d$  with periodic boundary conditions. Let  $L$  be the number of lattice sites in each direction. Your Monte Carlo algorithm should help you compute three quantities

$$\sigma = \frac{1}{L^d} \sum_x \langle \chi_x^2 \rangle, \quad \chi = \frac{1}{L^d} \sum_{x,y} \langle \chi_x \chi_y \rangle, \quad F = \frac{1}{L^d} \sum_{x,y} \langle \chi_x \chi_y \rangle \cos(2\pi \Delta\tau/L),$$

for a fixed value of  $L$ ,  $\kappa$  and  $g$ . In the definition of  $F$  the quantity  $\Delta\tau$  is the temporal separation in lattice units between  $x$  and  $y$ . Using  $\chi$  and  $F$  let us define a finite size mass  $M(L)$  using the formula

$$M(L) = \frac{2 \sin(\pi/L)}{\sqrt{((\chi/F) - 1)}}.$$

In the next project you will show that in the massive symmetric phase  $M(L \rightarrow \infty) = M_{\text{phys}}a$ ). We will use this to compute the physical mass of the scalar particle in the Monte Carlo method.

1. For  $g = 0$  write a computer program to exactly compute the three observables. Verify that  $M(L) = m_0 a$  independent of  $L$  when  $g = 0$ . This implies  $M_{\text{phys}} = m_0$  in the free theory as expected. Also have a computer code handy to compute the observables when  $\kappa = 0$ . In your report tabulate exact results for  $L = 16$ ,  $g = 0$ , and  $\alpha = 0.25$  and  $L = 16$ ,  $g = 0.1$ , and  $\kappa = 0$  in  $d = 2, 3, 4$ .
2. Using online lectures provided on the webpage or otherwise construct a Monte Carlo algorithm and develop computer codes to study the above  $d$  dimensional lattice field theory. Verify that your Monte Carlo is producing the right results when  $g = 0$  but  $\kappa \neq 0$  and when  $\kappa = 0$  but  $g \neq 0$ . You can use the exact results from above for this purpose.
3. Using your Monte Carlo method compute the three observables and  $M(L)$  for  $L = 8$ ,  $(ma)^2 = -1.5$  and  $g = 0.1$  in  $d = 2, 3$  and  $4$ . Tabulate your results in the form shown below.

$d$	$\sigma$	$\chi$	$F$	$M(L)$

Make sure the errors in your results are around 0.2%.

# Lecture 1

Introducing detailed balance & ergodicity, plus weights, configs, and updates

## Monte-Carlo Methods

Use: Compute Correlation functions  
in QFT.

$$\langle \phi(x) \phi(y) \rangle = \frac{1}{Z} \int [d\phi] e^{-S_E(\phi)} \phi(x) \phi(y)$$

$$Z = \int [d\phi] e^{-S_E(\phi)}.$$

If we define

$$P[\phi] = \frac{e^{-S_E(\phi)}}{Z},$$

Then

$$\langle \phi(x) \phi(y) \rangle = \int d\phi P[\phi] \phi(x) \phi(y).$$

Example:

$$\langle f(x) \rangle = \int dx f(x) P(x)$$

↑  
Prob. dist.

Monte-Carlo method:

Design a procedure to generate

$x_i, i=1, 2, 3, \dots$  density  
such that the probability  $x_i$  is  
produced is  $P(x_i)$ .

Then

$$\langle f(x) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_i f(x_i)$$

A monte carlo method produces  
 $x_i$  according to  $P(x_i)$ .

For a field theory Let us define

$$\epsilon \equiv [\phi] \quad \text{Configuration} = C$$

Then a MC method produces

$$\epsilon_i \quad i=1,2,3,\dots,N$$

such that they are distributed  
according to  $P([\phi])$

We can define the weight of  
a configuration  $W(\epsilon)$ . Weight of config =  $W(C)$   
in our context

$$P(\epsilon) = \frac{w(\epsilon)}{\sum_{\epsilon} w(\epsilon)}$$

$$P(\epsilon) = \frac{w(\epsilon)}{\int_{\phi} w(\epsilon)}$$

Given a weight, I can always calc  
probability distribution, so the  
weights are what's important

$$W(\epsilon) \equiv \mathcal{C}^{-S_E(\phi)} [d\phi]$$

Q: How can we design a MC method for this purpose?

Definition: Update

An update is a probabilistic procedure to generate a new configuration  $\ell_{\text{new}}$  given an

old configuration  $\ell_{\text{old}}$ .

It is usually specified by a transition probability.

$$T_{\ell_i \rightarrow \ell_f} .$$

$$\sum_f T_{\ell_i \rightarrow \ell_f} = 1$$

Detailed balance

If  $T_{\ell_i \rightarrow \ell_f}$  satisfies the relation  $\curvearrowleft$  detailed balance.

$$W(\ell_i) T_{\ell_i \rightarrow \ell_f} = W(\ell_f) T_{\ell_f \rightarrow \ell_i}$$

then we say that the

update satisfies detailed balance!

Ergodicity

Suppose we start with a  $T$  that satisfies detailed balance.

$\ell_{\text{init}} \xrightarrow{\text{update}} \ell_1 \xrightarrow{\text{update}} \ell_2 \xrightarrow{\text{update}} \ell_3 \rightarrow \dots$

and produce configurations such that all configurations can in principle be visited.

then the update is called ergodic.

If updates satisfy detailed  
and are ergodic then

$$C_{\text{init}} \rightarrow C_1 \rightarrow C_2 \rightarrow \dots \rightarrow \overset{\text{`$e$'}}{\underset{\text{$\omega$}}{\lim}} C_\infty$$

$C_\infty$  is distributed according to

$$P(e)$$
!

Suppose we have two or more  
(i.e. 2 or more updates)

$T$ 's such that each obeys  
detailed balance but are not  
ergodic individually, but are  
ergodic when combined then  
again  $C_\infty$  has the correct  
distribution.

Monte Carlo method is a design  
of an update that produces  
 $C_\infty$  distributed correctly.

Q: How do we design updates  
that satisfy detailed balance?

(Usually we don't care about ergodicity  
because we'll have a sufficient # of updates  
that, combined together, give ergodicity)

# Lecture 2

Introducing 2-update procedure and examining the 1<sup>st</sup> update (SPIN)

## Application to Scalar Field Theory

Goal: Develop an update procedure for the scalar field theory on a lattice with the action "I" for lattice

$$S_E^l(x) = \sum_x \left\{ \frac{1}{2} x_x^2 + g x_4^4 \right\} - x \sum_{x, \mu=1,2,\dots,d} x_x x_{x+\hat{\mu}}.$$

↑  
site term      bond term.

$x_x = \text{real\#}$   
↑  
lattice site

If  $\epsilon$  is a configuration then we want to design the  $T_{\epsilon_i \rightarrow \epsilon_f}$  such that

$$W(\epsilon_i) T_{\epsilon_i \rightarrow \epsilon_f} = W(\epsilon_f) T_{\epsilon_f \rightarrow \epsilon_i}$$

What is weight on lattice?

If  $\epsilon$  was  $[x]$  then

$$W(\epsilon) = \exp(-S_E^l(x)).$$

We will devise two such  $T$ 's each of which "may" not be ergodic but together will be ergodic.



design two updates.

→ Motivate why 2 updates

Let us write

$$\chi_x = \rho_x s_x$$

↑  
radial  
variable  
(always  
positive)

sign of  $\chi$ .

$$S_E^l(\chi) = \sum_x \left\{ \frac{1}{2} \rho_x^2 + g \rho_x^4 \right\} - \sum_{\langle x,y \rangle} \chi \rho_x \rho_y \rho_x \rho_y.$$

All site variables are  
indep. Of spin!

All bond terms have  
spin!

The 2 Updates

We will develop 2-updates

- (i) Spin-update . ( hold  $\rho$  fixed and change  $s$  )
- (ii) Regular update ( change  $\chi$  ).

$$\tau \equiv [S, S].$$

For the spin update.  $\rho$  is fixed, we can define

$$W(e) = \exp\left(-\sum_{\langle xy \rangle} k s_x s_y s_x s_y\right).$$

$$\text{Define } \beta_{xy} = k s_x s_y \geq 0$$

↑

bond dependent coupling.

$$e \in [s] \quad (\text{b/c rho is fixed})$$

Spin-update is not ergodic but helps in the efficiency of the algorithm -

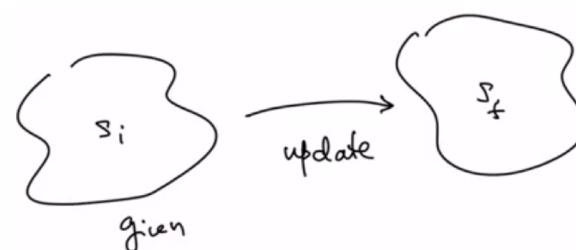
(b/c Chi ( $X$ ) can change from positive to negative, which could be big change, which helps with efficiency in the updates)

$$W(e) = \prod_{\langle xy \rangle} e^{\beta_{xy} s_x s_y}$$

Find a  $T_{[s]; \rightarrow [s_f]}$  such that it obeys detailed balance.

$W(e) \equiv$  weight of a <sup>bond</sup> disordered Ising model.

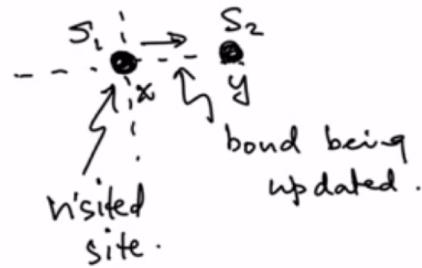
↓  
How should we implement  $T_{s_i \rightarrow s_f}$



update  
(flip of  
a set  
of spins)

A procedure to flip a spins for set of clusters in a configuration

$$[s]_i \rightarrow [s]_f \quad \left\{ \begin{array}{l} \text{single cluster} \\ \text{Wolff algorithm.} \end{array} \right\}$$



If  $s_1 \neq s_2$  do not put  $s_2$  on your visited site list.

- Algorithm to implement T is as follows:

keep two lists  $\begin{cases} \rightarrow \text{sites visited} \\ \rightarrow \text{sites visited flag} \end{cases}$

(i) Pick a site at random.

↑  
first site visited.  
↑  
Put this on the list  
and flag the site as  
visited.

(ii) Start going over all  
visited sites and look over  
all neighbouring directions

(iii) repeat this for all the  
sites on the visit list.

(iv) When all sites on the  
visit list is exhausted then  
flip ( $s \rightarrow -s$ ) on all the  
sites on the visit list!

# Lecture 3

Examining the regular update and observables

Discuss updates for the lattice Model:

$$S_E^L = \sum_x \left( \frac{1}{2} \chi_x^2 + g \chi_x^4 \right) - k \sum_{\langle xy \rangle} \chi_x \chi_y$$

$\underbrace{\qquad\qquad\qquad}_{\text{depends only on}}$        $\underbrace{\qquad\qquad\qquad}_{\text{all neighboring bonds}}$

$$\chi_x = S_x S_x . \quad \begin{matrix} \uparrow \\ \text{ignore in the spin update} \end{matrix}$$
$$\sum_{\langle xy \rangle} \beta_{xy} S_x S_y$$
$$\beta_{xy} = \begin{matrix} \uparrow \\ \text{Spin update weights} \end{matrix} \times S_x S_y .$$

We designed a single cluster algorithm  
to flip a set of spins.

### Regular Update

Idea is to go to each lattice site  
and update  $\chi_x$  there! We can do  
this sequentially and visit all  
lattice sites.

Q: What is the single site update.

Let us consider  $e$  to be  $\underline{\underline{x}}_x$ . Then

$$W(e) = \exp\left(-\frac{1}{2}x_x^2 - g x_x^4 + k x_x \sum_{\langle xy \rangle} x_y\right)$$

$\sum_{\langle xy \rangle} x_y$  = Sum over all values of the field  $x$  at nearest neighbor sites of  $x$ .

$$= \sum_{x, \hat{\mu}} (x_{x+\hat{\mu}} + x_{x-\hat{\mu}})$$

$$k \cdot \sum_{x, \hat{\mu}} (x_{x+\hat{\mu}} + x_{x-\hat{\mu}}) = \alpha$$

fixed while we update  $x_x$

$$W(e) = \exp\left(-\frac{1}{2}x_x^2 + \underbrace{\alpha x_x}_{\text{The weight}} - g x_x^4\right)$$

$$= \exp\left(-\frac{1}{2}(x_x - \alpha)^2 - g x_x^4\right) e^{\frac{\alpha^2}{2}}$$

The weight

Const, doesn't change with updates, thus can ignore

Complete the square

I need  $\chi_x$  that is distributed according the weight

$$\exp\left(-\frac{1}{2}(\chi_x - \alpha)^2 - g \chi_x^4\right)$$

If  $g=0$  then this is a Gaussian distribution.

$$\chi_x^{\text{old}} \xrightarrow{T} \chi_x^{\text{new}} = \chi + \alpha$$

↑  
distributed  
according to  
 $P(\chi) = e^{-\frac{1}{2}\chi^2}$

Heat bath algorithm.

For  $g \neq 0$ , we still produce  $\chi$  according to  $P(\chi)$ . we propose this  $\chi_x^{\text{new}} = \chi + \alpha$  as the new  $\chi$  but we only accept it with probability

$$P = \frac{\exp(-g \chi_x^{\text{new}})}{\exp(-g \chi_x^{\text{old}})}$$

Draw a uniform random number  $r$  between 0 and 1 and if  $r < P$  then  $\chi_x^{\text{new}} = \chi + \alpha$  otherwise

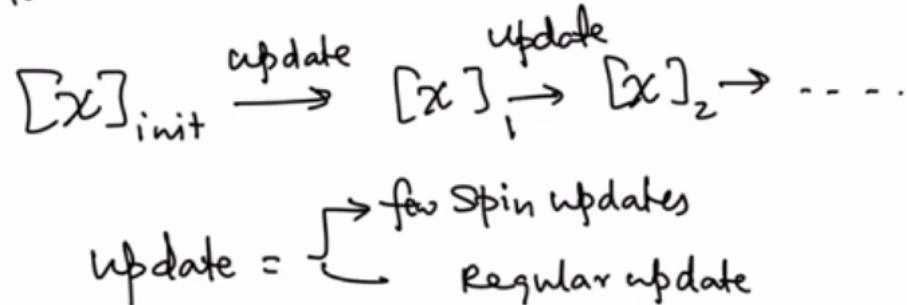
$$\chi_x^{\text{new}} = \chi_x^{\text{old}}$$

(For small  $g$ ,  $r < P$  most of the time)

→ Repeat this sequentially at all sites!

Combining the spin update with the regular update gives you the full algorithm.

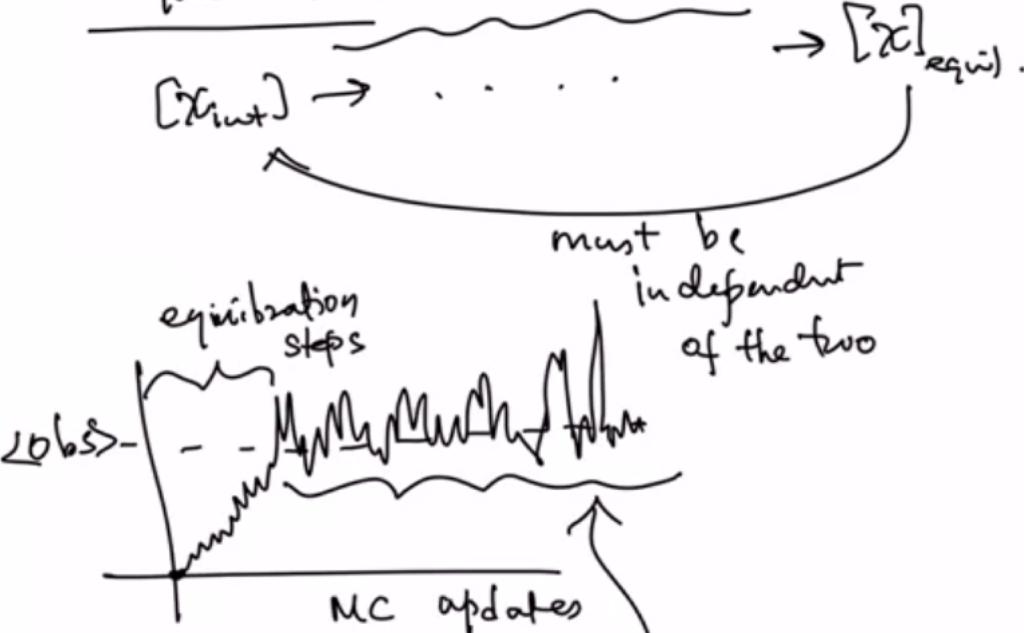
Start with some initial.



Typically, if we do  $\sim 1,000\text{-}5,000$  updates, we have enough

Equilibrate: equilibration steps  $\approx 5000, 10000$

$[x]_{\text{init}} \rightarrow \dots$

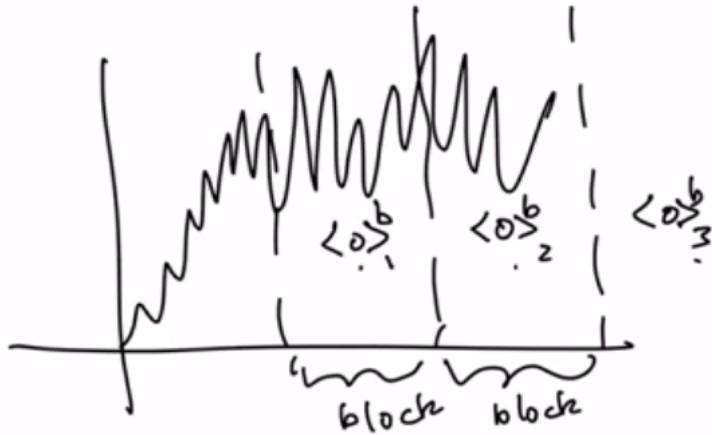


$$\langle O \rangle = \frac{1}{N} \sum_{i=1}^N O_i$$

Once you equilibrate, you can start measuring your observable

$$\delta \langle O \rangle = \frac{1}{\sqrt{N}} \sqrt{\langle O^2 \rangle - \langle O \rangle^2} \quad \text{for large } N.$$

↑  
error in your observable.



More reliable way to calculate error:  
 Use average of the observable over some block  
 of updates as single “observable” point

$$\langle O \rangle = \frac{1}{N} \sum_{i=1}^N \langle O_i^b \rangle$$

$$\delta \langle O \rangle \approx \frac{1}{\sqrt{N}} \sqrt{\langle \langle O_i^b \rangle^2 \rangle - \langle \langle O_i^b \rangle \rangle^2}$$

## Observables

$$C_0^L, \sigma$$

$$\sigma = \frac{1}{L_x^d L_T} \sum_x \langle x^2 \rangle$$

↑ volume  
easy.

$$\sigma_i = \frac{1}{N} \sum_{i=1}^N \sigma_i$$

$\sigma_i = \sigma$  for a given conf that is generated.

# Lecture 4

Measuring observables, doing the algorithm

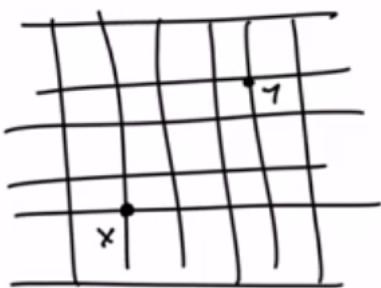
$$\chi = \frac{1}{L^d} \sum_{x,y} \langle x_x x_y \rangle$$

$$F = \frac{1}{L^d} \sum_{x,y} \langle x_x x_y \rangle \cos\left(\frac{2\pi}{L} \Delta\tau\right)$$

$\Delta\tau$  = temporal coordinate difference  
between  $x$  and  $y$ .

Pictorially

" $x_x^i$  (conf)



$x_x^i$

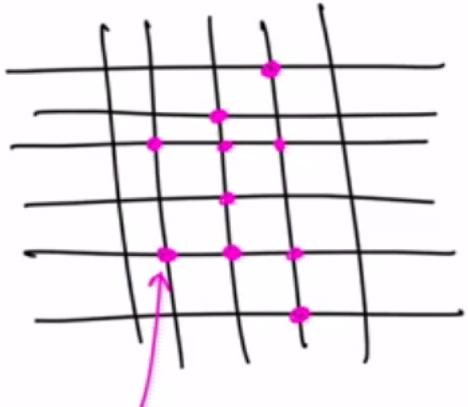
$$\chi_F = \left\langle \frac{1}{L^d} \sum_{x,y} x_x^i x_y^i \cos\left(\frac{2\pi}{L} \Delta\tau\right) \right\rangle$$

Naively  $\Rightarrow$  inefficiës!

$\rightarrow$  regular update.  
 $\rightarrow$  Spin-update Cluster

getsus()  $\rightarrow$   $\chi, F$  using the information  
of the cluster.

Not discuss the proof.



"cluster of spins".  
 radial spin  
 $x_x = \sum_x s_x$

All spins in  
the cluster  
are the same

first site "x"

All other sites including the first site  
will be "y".

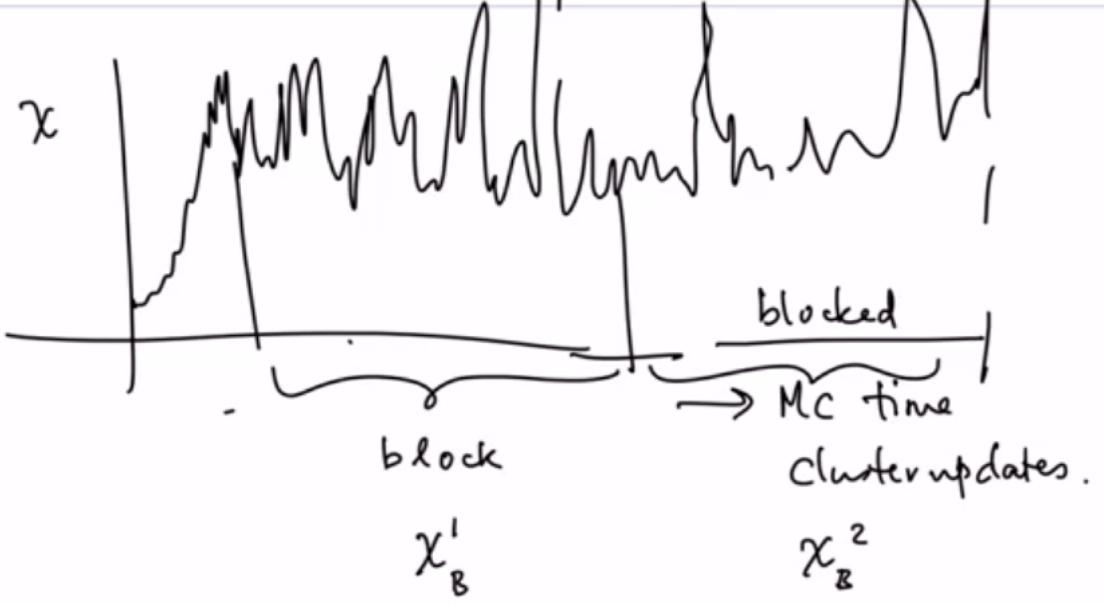
$$|x_x^i| |x_y^i|$$

$$\sum_y \uparrow_x^i \downarrow_y^i$$

If a site is not in the cluster, then it will not contribute to the observable for that particular config

$$x = \left\langle \sum_y \uparrow_x^i \downarrow_y^i \right\rangle$$

$$F = \left\langle \sum_y \uparrow_x^i \downarrow_y^i \cos\left(\frac{2\pi}{L} \Delta \tau\right) \right\rangle$$



$$x_B^1$$

$$x_B^2$$

$$\chi = \frac{1}{N} \left( \sum_{i=1}^N x_B^i \right)$$

$$F = \frac{1}{N} \left( \sum_{i=1}^N F_B^i \right)$$