Lattice Quantum Field Theory — A Constructive Approach

Daniel Doro Ferrante *

November 18, 2001

Abstract

In this paper it will be reviewed the lattice formulation of Quantum Field Theory (QFT) under a *constructive* approach, i.e., the theory will be developed as if its fundamental formulation were the lattice one.

Contents

*e-mail: danieldf@het.brown.edu

1 Introduction

There are some advantages in using the lattice formulation of QFT, among them we have:

- The lattice regularization provides the only non-perturbative definition of Path Integral and, ∴ of Field Theories;
- It is possible to quantify with precision the error commited in the numerical approximation of the integrals; &
- It is, in principle, possible to reduce arbitrarily this error by approaching the continuum limit (reducing the lattice spacing) and increasing the statistical samples in the Monte Carlo (MC) integration.

Thus, in order to built the Lattice QFT we have to define our basic objects, such as the lattice itself, lattice differentiation (finite difference operators), lattice integration, classical field theories, lattice path integrals and QFT's.

1.1 Definitions

Let us start by defining what is meant by a lattice.

Definition 1.1. *The [Euclidean] lattice (denoted by $\mathbb{L}^d$, where d is the* dimension *of the lattice) is a [finite] set of points with a neighborhood relationship among them. The type and structure of this neighborhood relationship will determine the dimension of the lattice.*[a] *Each point in this set is called a site and the neighborhood relations are represented by lines connecting these sites, and will be called links.*

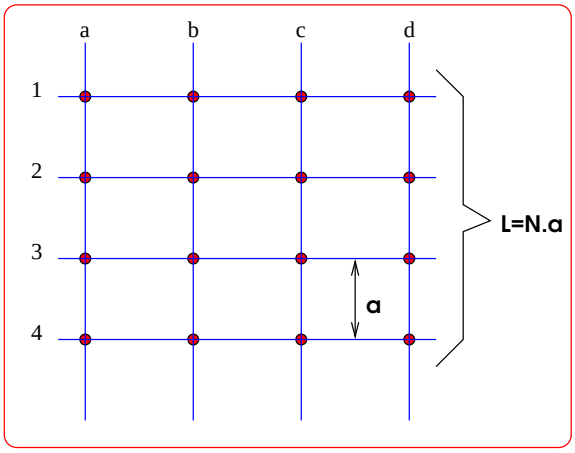

Figure 1: Standard 2-dim lattice.

It is worth noting, at this point, that there is no other additional structure attached to this lattice. Thus, we have something that we could call a *topological space* for now (there is no notion of distance for instance, therefore, there is no notion of geometry). However, once it is desired that this space have a continuum limit of interest, we will introduce the basic distance in this lattice, the *lattice spacing*, usually denoted by "a". Figure 1 should help us picture these

[a] The picture that the reader should have in mind is the one of a "discrete" $\mathbb{R}^d$ (namely, $\mathbb{Z}^d$). (However, once a *constructive* approach is being used, it should be understood that the inverse process will take place, i.e., from this discrete set it will be possible to take a continuum limit that will give us $\mathbb{R}^d$.)

concepts. Now, in light of figure 1, if it is said that in a given direction there are N sites, it is easy to see that the total length (L) of the lattice — in that direction — will be given by $L = N \cdot a$.

Remark. *It has to be noted that, one would not need to introduce neither the lattice spacing "a" nor the lattice length — which follows directly from "a": $L_i = N_i \cdot a_i,\ i = 1, \ldots, d$, where "$d$" is the dimension of the lattice —, granted that these quantities will never enter any of the future calculations, as will be seen. Also, from a more physical perspective, once the field to be simulated is defined, its mass term (assuming it has one) will determine the "natural distance scale" in the lattice, given by the inverse of its mass. (If the simulated field has no mass*[b]*, the answers will be in units of distance, thus, when the continuum limit is desired, one would choose this "unit of distance", "a" — the lattice spacing.)*

The continuum limit is the mathematical difficulty of the theory, as will be seen later on, it contains the main problems and the deep questions of the theory.

It is also notable that a lattice has *boundary conditions*, i.e., one has to specify what happens on the boundaries of this structure. Thus, in this view, one has to worry about periodic or fixed boundary conditions. In periodic boundary conditions, one constrains the end of a certain direction to meet the beggining of that direction, i.e., the given direction is circular. When one has fixed boundary conditions, there is no such thing, therefore, upon reaching the end of a direction, there are "boundary points". This is basically the same difference between a surface without boundary (e.g., a compact surface, like a sphere) and a surface with boundary (e.g., a cube, whose boundaries are the lines joining its vertices).

Now let us move on to understanding what do scalar fields mean on this framework.

Definition 1.2. *A scalar field on the lattice (denoted by φ) is a function such that:*

$$\begin{aligned} \varphi \ : \ \mathbb{L}^d &\to \mathbb{R} \\ \vec{n} \ &\mapsto \varphi(\vec{n}) \end{aligned}$$

That is, for a given point in the lattice (represented by its position "vector", $\vec{n}$) there is a real number associated with it.

Note that this field (φ) is dimensionless, once there are no physical dimensions associated with the field itself.

Remark. *It is being introduced an integer-coordinate system, $n_\mu,\ \mu = 1, \ldots, d$, where each component n_μ is an integer number that enumerates the sites along a given direction in $\mathbb{L}^d$.*

[b] In this case, other "technicalities" will have to be taken into account as well, for instance, the fact the there will be a "zero mode", i.e., the lattice Laplacian will have a vanishing eigenvalue.

Now, just like in the continuum case, one can have functions of this field, which are called *functionals* — because they are functions of a function. One important (for defining a particular theory) functional is called *Action*, denoted by $\mathcal{S}[\varphi]$[c]. Physically speaking, the action determines the dynamics of a given theory (where, by theory, a Lagrangian is understood). Generally speaking, the action should satisfy two basic conditions:

1. **Boundedness** The action functional should be bounded from below, i.e., $\exists\, \mathcal{S}_m \in \mathbb{R}$ such that $\mathcal{S}_m = \inf_{\varphi\in\Phi}(\mathcal{S}[\varphi])$, where $\Phi = \{\varphi : \varphi \in \mathcal{D}\varphi\}$ and $\mathcal{D}\varphi$ is the set of all possible configurations of the field φ taken into account in our simulation[d] (this condition is responsible for the stability of the theory.); &

2. **Locality** The action should be a "local function" of the points in $\mathbb{L}^d$, i.e., it should not involve the product of fields which are not in neighboring sites (this will imply in the continuum limit that one has terms with second order derivatives at most).

A particular form of the action that shall be of further interest will be

$$\mathcal{S}_0[\varphi] = \frac{1}{2}\sum_l (\Delta_l\varphi)^2 + \frac{\alpha}{2}\sum_s \varphi^2(s), \quad \alpha \geqslant 0, \tag{1}$$

where

$$\sum_s \equiv \sum_{\vec{n}} \equiv \sum_{n_1=1}^{N_1}\cdots\sum_{n_d=1}^{N_d},$$
$$\sum_l \equiv \sum_\mu\sum_s \equiv \sum_{\mu=1}^{d}\sum_{n_1=1}^{N_1}\cdots\sum_{n_d=1}^{N_d},$$
$$\Delta_l\varphi \equiv \Delta_\mu\varphi = \varphi(\vec{n}+\hat{\mathbf{n}}_\mu) - \varphi(\vec{n})\ .$$

The above action ($\mathcal{S}_0[\varphi]$) is also known as the *free* action and Δ_l is the lattice version of the partial differential operator. It is not difficult to see that this free action satisfies both of the requisites above — as long as $\alpha \geqslant 0$.

1.2 Classical Theory

For the sake of completeness, let us have a look at what happens with the classical field theory obtained from the above action (free).

[c] The notation is that functionals are denoted by $F[g]$, as opposed to functions, which are denoted by $g(x)$.

[d] It will be seen later that, the set of all the configurations of the field accounted for by our simulation has a *measure*, that shall be denoted by $\mathcal{D}\varphi$. Thus, this dubious notation is to remind the reader of this fact.

Classically, the solution of the field theory described by some action functional, is the field configuration that minimizes the action. It is easy to see that this field configuration for the free theory — as described by the action above — is the one in which the field vanishes at every point of $\mathbb{L}^d$. (The reader should note that periodic boundary conditions are being assumed.)

Now, let's see if the continuum limit can be understood. For such a purpose, it will be introduced a dimensional scale in our theory, i.e., let's choose a particular lattice spacing for lattice given. (With this, a metric structure is being introduced on the lattice.) The continuum limit is taken in such a way that $N \to \infty$, $a \to 0$, however, $L = N \cdot a$ is kept constant. (The theory is kept in a "box" of volume $V = L^d$ and with periodic boundary conditions.) Also, it is being assumed that the lattice remains symmetric and homogeneous (basically, those last two requirements could be substituted by requiring isotropy instead), given that an external scale has just been introduced. Thus, let us rewrite the free action in the following way:

$$\mathcal{S}_0[\varphi] = \frac{1}{2} \sum_s a^d \sum_\mu \left[\frac{a^{(2-d)/2}\, \Delta_\mu \varphi(s)}{a} \right]^2 + \frac{\alpha}{2\, a^2} \sum_s a^d \left[a^{(2-d)/2} \varphi(s) \right]^2 . \tag{2}$$

Now, let us make $\alpha \to 0$ by choosing $\alpha = (m \cdot a)^2$ — for some finite m — and let's define the dimensional version of the field as $\phi = a^{(2-d)/2}\varphi$. Therefore, the action can still be written as

$$\mathcal{S}_0[\varphi] = \frac{1}{2} \sum_s a^d \sum_\mu \left[\frac{\Delta_\mu \phi(s)}{a} \right]^2 + \frac{m^2}{2} \sum_s a^d\, \phi^2(s) .$$

In this fashion, it is not difficult to see that, performing the limiting procedure described above (namely, let $N \to \infty$ and $a \to 0$ — keeping L fixed),

$$\lim_{\substack{N\to\infty \\ a\to 0 \\ L=\text{cte}}} \mathcal{S}_0[\varphi] = S_0[\phi] = \frac{1}{2} \int_V \sum_\mu \left[\partial_\mu\, \phi(\vec{x}) \right]^2 d^d x + \frac{m^2}{2} \int_V \phi^2(\vec{x})\, d^d x, \quad x_\mu = a \cdot n_\mu . \tag{3}$$

As already mentioned, the solutions of such an action are the Euler-Lagrange equations, that can be obtained via the principle of least (extremum, to be specific) action. Therefore, the answer for the particular case above would be

$$\left(-\triangle + m^2 \right) \phi(\vec{x}) = 0, \quad \triangle = \partial^2 = \partial_\mu \partial^\mu .$$

It is easy to see that there exists a lattice counterpart for this by noting that

$$\sum_l (\Delta_l \varphi)^2 = -\sum_s \varphi(s) \triangle \varphi(s) \ ,$$

where $\triangle\varphi(\vec{n}) = \sum_\mu \Big[\varphi(n_\mu - 1) - 2\,\varphi(n_\mu) + \varphi(n_\mu + 1)\Big]$. Thus, the lattice[e] version is given by

$$(-\triangle + \alpha)\,\varphi(s) = 0 \ . \tag{4}$$

1.3 Fourier Transforms in the Lattice

At this point, it is very useful to define what is meant by a *finite* Fourier transform, or a lattice Fourier transform. But, before doing so, Let us just note a couple of facts about what will be the lattice version of the momentum space.

Due to physical interpretation reasons, it is convenient to take k (either a 1-dim vector or the value of a component of the vector $\vec{k} = (k_1, \ldots, k_d)$) as symmetric as possible (around 0), thus it will be adopted that

$$k = \begin{cases} -(N-1)/2, \ldots, 0, \ldots, (N-1)/2, & \text{if } N \text{ is } \underline{\text{odd}}; \ \& \\ -(N/2-1), \ldots, 0, \ldots, (N/2-1), N/2, & \text{if } N \text{ is } \underline{\text{even}}. \end{cases}$$

Thus, the reader can see that k (or $\vec{k}$, for that matter) plays the same role as n (or $\vec{n}$), namely it enumerates the sites of the lattice. In this formulation, it is easy to see that both of them are just basis[f] for the lattice.

Definition 1.3. *Given a hypercubic lattice, with N points per direction and dimension d, that satisfies* periodic *boundary conditions, the Fourier transform and its inverse are defined as*

$$\mathscr{F}[\varphi] \equiv \tilde{\varphi}[\vec{k}] \equiv \frac{1}{N^d} \sum_{\vec{n}} e^{i\frac{2\pi}{N}\vec{k}\cdot\vec{n}}\,\varphi(\vec{n}),$$
$$\mathscr{F}^{-1}[\tilde{\varphi}] \equiv \varphi(\vec{n}) \equiv \sum_{\vec{k}} e^{-i\frac{2\pi}{N}\vec{k}\cdot\vec{n}}\,\tilde{\varphi}(\vec{k}),$$

where

$$\sum_{\vec{k}} \equiv \sum_{k_1=k_m}^{k_M} \cdots \sum_{k_d=k_m}^{k_M} ,$$
$$k_m = \min(k) \ \textit{and} \ \ k_M = \max(k) \ .$$

[e] It is left up to the reader to verify that the equation (4) can be derived by a lattice version of the principle of the least action applied to (1).

[f] And, it will be easy to realize that the Fourier transform is just the operation via which one goes from one basis to another.

It is worth noting a couple of relations:

$$\text{orthogonality: } \sum_{n=1}^{N} e^{i\frac{2\pi}{N}n(k-k')} = N\,\delta_{k,k'}\ ,$$
$$\text{completeness: } \sum_{k=k_n}^{k_M} e^{i\frac{2\pi}{N}k(n-n')} = N\,\delta_{n,n'}\ ,$$

where $\delta_{a,b}$ is the Kronecker[g] one, where $\delta_{a,a} = 1$ and $\delta_{a,b} = 0$, $a \neq b$. Their higher dimensional extensions are straightforward.

1.4 The Lattice Laplacian

Let us take a look at the eigenvalues of the lattice Laplacian. For this, it will be assumed a lattice in the same fashion as the one above (Sub-section 1.3). In this case, one has that

$$\Delta_\mu\, e^{i\frac{2\pi}{N}\vec{k}\cdot\vec{n}} = i\ \overbrace{\left(e^{i\frac{2\pi}{N}\vec{k}\cdot\underset{\sim}{\widehat{\mathbf{n}}}_\mu} - 1\right)}^{i\,2\sin(\frac{\pi}{N}\vec{k}\cdot\underset{\sim}{\widehat{\mathbf{n}}}_\mu)\,e^{i\frac{\pi}{N}\vec{k}\cdot\underset{\sim}{\widehat{\mathbf{n}}}_\mu}}\ e^{i\frac{2\pi}{N}\vec{k}\cdot\vec{n}},$$
$$\rho_\mu(k_\mu) \equiv 2\,\sin\left(\frac{\pi}{N}\vec{k}\,\underset{\sim}{\widehat{\mathbf{n}}}_\mu\right) = 2\,\sin\left(\frac{\pi}{N}k_\mu\right),$$
$$\triangle_\mu\, e^{i\frac{2\pi}{N}\vec{k}\cdot\vec{n}} = -\left[2\,\sin\left(\frac{\pi}{N}k_\mu\right)\right]^2 e^{i\frac{2\pi}{N}\vec{k}\cdot\vec{n}}\ .$$

Thus,

$$\triangle\, e^{i\frac{2\pi}{N}\vec{k}\cdot\vec{n}} = -\rho^2(\vec{k})\, e^{i\frac{2\pi}{N}\vec{k}\cdot\vec{n}},$$

where

$$\rho^2(\vec{k}) \equiv \sum_\mu \rho_\mu^2 = 4\left[\sin^2\left(\frac{\pi}{N}k_1\right) + \cdots + \sin^2\left(\frac{\pi}{N}k_d\right)\right].$$

Thus, the reader should note that $\rho_\mu(k_\mu)$ plays the role of the linear momentum in the lattice. (It also has the correct continuum limit.)

[g] The reader should note that, in the continuum limit that is being considered here, the δ-Kronecker goes to the δ-Dirac: $\lim_{N\to\infty} a^{-d}\delta^d_{\vec{n},\vec{n}'} = \delta(\vec{x}-\vec{x}')$.

1.5 External Sources and Green's Functions

Now, it will be introduced a new concept to this formulation, the *external source*. This is done via a new term in the action, e.g., for the free theory, that has been considered so far

$$\mathcal{S}_0^j[\varphi] = \mathcal{S}_0[\varphi] - \sum_s j(s)\,\varphi(s)\ .$$

The external source, $j(s)$, is simply a function of the lattice sites and, the only restriction that shall be imposed on it is that it be finite.

A *classical* solution for this action can also be obtained via the lattice version of the method of least action. The result turns out to be

$$\delta\mathcal{S}_0^j[\varphi] = \sum_s \delta\varphi(s)\Big[-\triangle\varphi(s) + \alpha\,\varphi(s) - j(s)\Big],$$

therefore

$$\Big[-\triangle + \alpha\Big]\,\varphi(s) = j(s)\ .$$

This is the inhomogeneous version of the equation of motion that was found before. Now, once it is desired to find solutions for the particular boundary conditions in question (periodic), let us make use of the Fourier transforms of the field and the source. Thus, one finds that

$$\begin{aligned}
&\mathscr{F}\Big\{[-\triangle + \alpha]\,\varphi(s) = j(s)\Big\},\\
&\sum_{\vec{k}} e^{-i\frac{2\pi}{N}\vec{k}\cdot\vec{n}}\Big[\tilde{\varphi}(\vec{k})\,(\rho^2(\vec{k}) + \alpha) - \tilde{j}(\vec{k})\Big] = 0,\\
&\qquad\therefore\quad \tilde{\varphi}(\vec{k}) = \frac{\tilde{j}(\vec{k})}{\rho^2(\vec{k}) + \alpha},\\
&\qquad\Rightarrow\quad \varphi(\vec{n}) = \mathscr{F}^{-1}\left[\frac{\tilde{j}(\vec{k})}{\rho^2(\vec{k}) + \alpha}\right].
\end{aligned}$$

Let us now treat a common case, the point-like[h] source (or point-like *charge*, in analogy with its EM counterpart). In this particular case, the source can be written as

[h] This means that the source vanishes everywhere on the lattice, except in one single point.

$$
\begin{aligned}
& j(\vec{n}) = j_0\,\delta_{\vec{n},\vec{n}'}, \\
\Rightarrow\quad & \tilde{j}(\vec{k}) = \frac{j_0}{N^d}\,e^{i\frac{2\pi}{N}\,\vec{k}\cdot\vec{n}'}, \\
\therefore\quad & \varphi(\vec{n}) = \frac{j_0}{N^d}\,\sum_{\vec{k}}\,\frac{e^{-i\frac{2\pi}{N}\,\vec{k}\cdot(\vec{n}-\vec{n}')}}{\rho^2(\vec{k})+\alpha}\;.
\end{aligned}
$$

It is not difficult to spot in the above equation what is known as *Green's function* (or *propagator*, as it will be called in the QFT scenario) and, as usual, it represents the answer of the system to the "perturbation" caused by the external source. Namely, it is defined as (note that $j_0 = 1$)

$$
\begin{aligned}
\tilde{g}(\vec{k}) &\equiv \frac{1}{N^d\left(\rho^2(\vec{k})+\alpha\right)}, \\
\therefore\quad g(\vec{n}-\vec{n}') &= \sum_{\vec{k}}\,\frac{e^{-i\frac{2\pi}{N}\,\vec{k}\cdot(\vec{n}-\vec{n}')}}{N^d\left(\rho^2(\vec{k})+\alpha\right)}\;.
\end{aligned}
$$

Furthermore, for an arbitrary external source, it can be proven that

$$
\begin{aligned}
& \tilde{\varphi}(\vec{k}) = N^d\,\tilde{j}(\vec{k})\,\tilde{g}(\vec{k}), \\
\Rightarrow\quad & \varphi(\vec{n}) = \sum_{\vec{n}}\,j(\vec{n}')\,g(\vec{n}-\vec{n}')\;.
\end{aligned}
$$

Now, the *scaling* relations that had been obtained before can be used at this point in order to determine how will the Green's function behave upon the continuum limit. It is not difficult to see that $(J = a^{-(d+2)/2}\,j)$ and that $(G = a^{2-d}\,g)$[i].

1.6 Monte Carlo Techniques

Monte Carlo (MC) is a statistical method used to compute, numerically, multidimensional integrals and, thus, it will be an important ingredient in one's simulations (c.f. Subsection 2.6).

In order to introduce the method, let us start with the simplest of the integrals

$$
\int_\alpha^\beta\,f(x)\,dx = \int_{\mathcal{D}}\,f\;,\quad \mathcal{D} = [\alpha,\beta]\;.
$$

The condition for Riemann integrability is given by:

[i]Note that G and $\tilde{G}$ have the same dimensions — given the conventions used so far — and, thus, $(\tilde{G} = a^{2-d}\,\tilde{g})$.

1. Pick a partition of $\mathcal{D}$: $\mathcal{P}_{\mathcal{D}}^{(n)} = \{(x_0 = \alpha), x_1, \ldots, x_{n-1}, (x_n = \beta)\}$;
2. Let $I_j = [x_j, x_{j+1}]$:

$$\varepsilon(n) = a \sum_{j=1}^{n-1} \left[\max_{x \in I_j} f(x) - \min_{x \in I_j} f(x) \right], \quad a \equiv \frac{\beta - \alpha}{n} ,$$

3. Thus, $f(x)$ is Riemann integrable[j] iff

$$\lim_{n \to \infty} \varepsilon(n) = 0 .$$

This definition is very practical from the point of view of simulations, because it gives a method of performing numerical integrations, namely:

$$\int_{\mathcal{D}} f \equiv a \sum_{j=1}^{n-1} f(x_j) + O(f'/n^2) .$$

The basic difference among distinct methods for performing numerical integrations resides in its behavior with respect to n: some methods converge faster than others. The "*truncation error*" performed at finite n is called *discretization error*.

1.6.1 Basic Monte Carlo

The MC method is an integration algorithm which has a *statistical* foundation. Here is it:

Algorithm 1: Basic Monte Carlo

1: Generate a set $\mathbb{P} = \{x^{[i]}\}_{i \in \mathcal{I}}$, where $\mathcal{I}$ is an *index* set, namely $\mathcal{I} = \{1, \ldots, n\}$, of N random points with *uniform* distribution in the domain of integration, $\mathcal{D}$;
2: **for all** $x^{[i]} \in \mathbb{P}$ **do**
3: Compute $f(x^{[i]})$;
4: **end for**
5: Compute the average

$$I(N) = \frac{\beta - \alpha}{N} \sum_{i=0}^{N-1} f(x^{[i]}) .$$

If f is Riemann integrable, then

$$\lim_{N \to \infty} I(N) = \int_{\mathcal{D}} f \quad .$$

[j] This condition is valid $\forall\ \mathcal{P}_{\mathcal{D}}^{(n)}$ and, equivalently, one can make $a \to 0$.

It should be noted that, the basic idea behind the method, is that a (several) partition(s) $\mathbb{P}$ be generated and f be computed within it. A good idea is to generate several partitions, compute f for each one of them and then average the result obtained for f.

However, MC integration is *not* very efficient for 1 dimensional integrals, but it becomes more and more efficient (when compared with "conventional" methods) for higher dimensional integrals.

1.6.2 The Metropolis Method

When dealing with k-dimensional integrals of the form

$$\mathscr{I}_k = \int_{\mathcal{D}_1} dx_1 \cdots \int_{\mathcal{D}_k} dx_k \left[f(\vec{x})\, P(\vec{x}) \right] , \quad \mathcal{D} = \bigcup_{j=1}^{k} \mathcal{D}_j , \tag{5}$$

a clever trick can be used in order to modify the algorithm above.

Algorithm 2: Random numbers with density probability P

1: Generate a set of N random points $\mathbb{P}_P = \{\vec{x}^{[i]} : \vec{x}^{[i]} \in \mathcal{D}\}$, using $P(\vec{x}^{[i]})$ for the probability distribution of the point $\vec{x}^{[i]} = (\vec{x}_1^{[i]}, \ldots, \vec{x}_k^{[i]})$;
2: **for all** $\vec{x}^{[i]} \in \mathbb{P}$ **do**
3: Compute $f(\vec{x}^{[i]})$;
4: **end for**
5: Compute the average

$$I_k(N) = \frac{\mathrm{Vol}(\mathcal{D})}{N} \sum_{i=0}^{N-1} f(\vec{x}^{[i]}) ,$$

where $\mathrm{Vol}(\mathcal{D})$ is the volume of the integration domain.

Thus, if $F(\vec{x}) = f(\vec{x})\, P(\vec{x})$ is Riemann integrable, the function $I_k(N)$ converges to the full integral in the limit $N \to \infty$. The factor $P(\vec{x})$ in the integrand has been absorbed into the probability distribution of generating the random points. A slightly different way to look at this would be as a measure change: $d\mu = dx_1 \cdots dx_k \to d\mu' = P(\vec{x}) d\mu$. Now, once integrals are invariant under a change in the measure, our result is guaranteed to appear. (The reader has to note that, the fact that $P(\vec{x})$ is being absorbed in the measure imposes some restrictions on P. On top of it, thinking of P as a probability distribution also imposes restrictions over it. However, those restrictions happen to coincide and they imply that P has to be positive definite.) Therefore, the problem now is the one of generating random points $(\vec{x}^{[i]})$ with a given distribution $P(\vec{x}^{[i]})$. The simplest method to solve this problem is known as the *Metropolis* algorithm (denoted by "MMC" — Metropolis Monte Carlo):

Algorithm 3: Metropolis method

1: Choose a point $\vec{x}^{[i]}$ at random in $\mathcal{D}$ and start at $i = 0$;
2: **while** $i \leqslant N$ **do**
3: Generate another random point, $\vec{y} \in \mathcal{D}$, and a random number $\alpha \in [0, 1)$
4: **if** $P(\vec{y})/P(\vec{x}^{[i]}) \geqslant \alpha$ **then**
5: $\vec{x}^{[i+1]} := \vec{y}$
6: **else**
7: $\vec{x}^{[i+1]} := \vec{x}^{[i]}$
8: **end if**
9: $i := i + 1$
10: **end while**

The succession of points $\{\vec{x}^{[i]}\}$ generated by this algorithm has the required probability distribution. In addition, this type of succession is also known as a *Markov chain*. This fact, actually, completely determines the problem. (The theory of *Markov* chains is a quite powerful one and the reader should familiarize with it.) In particular, each point $(\vec{x}^{[i]})$ of the chain is called a *configuration* and the theory guarantees that the Markov chain is *ergodic*. The MMC method only works with real probability functions (and the constraints imposed above), but f and x can be complex, vector, tensor, etc . . .

Remark. *The **if** condition in the algorithm for the MMC above is equivalent to the condition below:*

Algorithm 4: *Equivalent tests*

1: ***if*** $\{P(\vec{y}) \leqslant P(\vec{x}^{[i]})\}$ OR $\{P(\vec{y}) \geqslant \alpha\}$ ***then***
2: $\vec{x}^{[i+1]} := \vec{y}$
3: ***else***
4: $\vec{x}^{[i+1]} := \vec{x}^{[i]}$
5: ***end if***

2 Free Scalar QFT

Let us, now, try to constructively define a Quantum Theory of Fields, in an analogous way as done before for the classical one. One shall start from the same mathematical framework just laid down. As already mentioned, a *very* important point is that, as opposed to what happened in the Classical case, it will *not* be necessary to introduce an external dimensional scale (the lattice spacing) in the continuum limit.

Definition 2.1. *The Quantum Field Theory is defined as a finite* statistical *model over a given lattice.*

Definition 2.2. *The quantities of physical interest, called* observables*, will be statistical averages of functionals of the field within the model given.*

Thus, in the same way that statistical theories have a Boltzmann factor weighing its configurations, a QFT will also have one, given by the action that defines the theory

$$P[\varphi] \equiv e^{-\mathcal{S}[\varphi]} \ .$$

Thus, given a certain functional of the field, $\mathcal{O}[\varphi]$, its *expectation value* (in a lattice of size N, i.e., it has N sites per dimension) is given by

$$\langle \mathcal{O} \rangle_N \equiv \frac{\int \mathcal{O}[\varphi]\, P[\varphi]\, [d\varphi]}{\int P[\varphi]\, [d\varphi]} \ , \tag{6}$$

where

$$[d\varphi] \equiv \prod_s d\varphi(s) = \prod_{n_1}^{N} \cdots \prod_{n_d}^{N} d\varphi(\vec{n}) \ .$$

Remark. *It is worth noting that, in the continuum limit, the theory of Lebesgue integration —* measure *theory — can be applied here in a way analogous to what was done in sub-subsection 1.6.2. It is realized that $P[\phi] = e^{-S[\phi]}$ satisfies the criteria to be a measure. Thus (using the same notation as used in the aforementioned sub-subsection),*

$$\int_{\mathcal{D}} P[\phi]\, \mathcal{D}\phi = \int_{\mathcal{D}} \mathcal{D}\phi = \mathcal{N} \cdot \mathit{Vol}(\mathcal{D}) \ , \tag{7}$$

where $\mathcal{N}$ is the [total] number of configurations over which the integral is being performed, i.e., $\mathcal{D}\phi \equiv [d\phi] \equiv \prod_{j=1}^{\mathcal{N}} d\phi_j$. Usually, in the continuum limit, it is desired to consider the limit in which $\mathcal{N} \to \infty$, thus the measure is not *defined* a priori*! However, some "tricks" allow to formally define this measure for the* free *QFT. On the other hand, this measure can*not *be defined for any type of* interacting *QFT! Therefore, mathematically speaking, interacting QFT's are not defined at all. In addition, the reader should notice the [no so] subtle difference between the integrals in* (5) *and* (8) *— i.e., between the discrete and continuum cases. This difference*[k] *is what justifies the appearance of this "normalization" factor, $\mathcal{N}$.*

[k] The measures in both of them are different, the finite dimensional one, (5), has a finite — definite — measure, while the infinite dimensional case, (8), has an infinite measure! In other words, that is what having *fields* mean: an *infinite* number of variables.

The observables of greatest interest are those known as *correlation functions* or *functions of n-points*, defined by

$$\begin{aligned} \mathcal{O}[\varphi] &\equiv \varphi(\vec{n}_1)\cdots\varphi(\vec{n}_n), \\ g_N(\vec{n}_1,\ldots,\vec{n}_n) &\equiv \langle\mathcal{O}\rangle_N = \langle\varphi(\vec{n}_1)\cdots\varphi(\vec{n}_n)\rangle_N \ . \end{aligned}$$

Thus, once the QFT is defined in each [finite] lattice, let us define the QFT associated to it in the continuum limit. Once the observables of a QFT completely define it, it suffices to define what is meant by the continuum limit of the expectation values of any observable. Thus,

$$\langle\mathcal{O}[\phi]\rangle \equiv \lim_{N\to\infty} \langle\mathcal{O}[\varphi]\rangle_N \ .$$

The continuum version of a model will be well defined if the above limit exist and is finite for all observables of interest.

Remark. *The reader should note that, once the Fourier transform is, basically, a change of basis on the lattice,*

$$\begin{aligned} \langle\mathcal{O}[\varphi]\rangle &= \frac{\int \mathcal{O}[\varphi]\,P[\varphi]\,[d\varphi]}{\int P[\varphi]\,[d\varphi]} \ , \\ &= \frac{\int \mathcal{O}[\tilde{\varphi}]\,P[\tilde{\varphi}]\,[d\tilde{\varphi}]}{\int P[\tilde{\varphi}]\,[d\tilde{\varphi}]} \ , \\ &\equiv \langle\mathcal{O}[\tilde{\varphi}]\rangle \ . \end{aligned}$$

2.1 Gaussian Integration

The basic integral that one shall have to deal is known as a Gaussian one. The reader should be familiar with those, but for the sake of completeness, let us make a brief summary.

Notation. *The following notations are equivalent:*

$$\langle\vec{y}\,|\mathbb{A}|\,\vec{y}\rangle = \langle\vec{y},\mathbb{A}\vec{y}\rangle = (\vec{y},\mathbb{A}\vec{y}) = \vec{y}\cdot\mathbb{A}\cdot\vec{y} = \vec{y}^T\mathbb{A}\vec{y} = \sum_{i,j=1}^{n} y_i\,[\mathbb{A}]_{ij}\,y_j \ ,$$

where $\vec{y} = (y_1,\ldots,y_n)$ and $\mathbb{A}$ is a real, symmetric, positive definite matrix. Also, note that, for the above $\mathbb{A}$,

$$\det\mathbb{A} = e^{\mathrm{tr}[\log(\mathbb{A})]} \ .$$

Thus,

$$\mathscr{I}(\mathbb{A}) = \int \cdots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\langle \vec{y}, \mathbb{A}\vec{y}\rangle} dy_1 \cdots dy_n = (2\,\pi)^{n/2} \left[\det(\mathbb{A})\right]^{-1/2} , \tag{8}$$

$$\Rightarrow \quad (2\,\pi)^{-n/2} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\langle \vec{y}, \mathbb{A}\vec{y}\rangle} d\vec{y} = e^{-\frac{1}{2}\,\mathrm{tr}[\log(\mathbb{A})]} \quad .$$

And, a slight generalization, gives us:

$$(2\,\pi)^{-n/2} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\langle \vec{y}, \mathbb{A}\vec{y}\rangle + \langle \vec{\rho}, \vec{y}\rangle} d\vec{y} = e^{-\frac{1}{2}\,\mathrm{tr}[\log(\mathbb{A})]}\, e^{-\frac{1}{2}\langle \vec{\rho}, \mathbb{A}^{-1}\vec{\rho}\rangle} \quad .$$

Remark. *The analogous relations for the continuum limit (Path Integrals) would be:*

$$\int \mathcal{D}\phi\, e^{-\frac{1}{2}\iint \phi(x)\, A(x,x')\, \phi(x') dx\, dx'} = e^{-(1/2)\,\mathrm{tr}[\log(A(x,x'))]} \quad ,$$

where

$$\begin{aligned} A(x,x') &= (\partial_x\, \partial_{x'} + r)\, \delta(x-x') \\ r = (cte), \quad \delta(x-x') &= \int_{-\infty}^{\infty} \frac{dp}{2\pi}\, e^{i\,p(x-x')} \\ \Rightarrow \quad A(x,x') &= \int_{-\infty}^{\infty} \frac{dp}{2\pi}\, e^{i\,p(x-x')}\, (p^2 + r) \\ \therefore \quad \mathrm{tr}[\log(A)] &= \int dx \int \frac{dp}{2\pi}\, \log(p^2 + r) \end{aligned}$$

Generalizing...

$$\int \mathcal{D}\phi\, e^{-\frac{1}{2}\int \left(\int \phi(x)\, A(x,x')\, \phi(x')\, dx + \int \rho(x)\, \phi(x)\, dx\right) dx'} = \underbrace{e^{-\frac{1}{2}\,\mathrm{tr}[\log(A)]}}_{=\sqrt{\frac{1}{\det(A)}}}\, e^{\frac{1}{2}\int \left(\int \rho(x)\, A^{-1}(x,x')\, \rho(x')\, dx\right) dx'}$$

where

$$A^{-1}(x,x') = \int_{-\infty}^{\infty} \frac{dp}{2\pi}\, \frac{e^{-i\,p(x-x')}}{p^2 + r}$$

Thus, the Gaussian integral, as presented above — (8) —, is a *functional* of $\mathbb{A}$.

Now, the action ($\mathcal{S}_0[\varphi]$) can be rewritten in terms of $\tilde{\varphi}$. For doing so, it is needed to Fourier transform everything, use the δ-functions in order to perform the sums over $\vec{k}'$ and remember that $\rho_\mu(-\vec{k}) = -\rho_\mu(\vec{k})$ and (reality condition) $\tilde{\varphi}(-\vec{k}) = \tilde{\varphi}^*(\vec{k})$. The final result will be

$$\mathcal{S}_0[\tilde{\varphi}] = \frac{N^d}{2} \sum_{\vec{k}} \left[\rho^2(\vec{k}) + \alpha\right] \left|\tilde{\varphi}(\vec{k})\right|^2 \quad . \tag{9}$$

Therefore, the normal modes are uncoupled, i.e., there are *no* products of $\tilde{\varphi}$ with different $\vec{k}$ dependence. The action has, actually, been anti-diagonalized: the momenta have been paired as $(\vec{k}, -\vec{k})$ — a product of the Fourier transform of the field by its complex conjugate.

Remark. *The action can be cast in the following form:*

$$\mathcal{S}_0[\varphi] = \frac{N^d}{2} \sum_{\vec{k},\vec{k}'} \tilde{\varphi}(\vec{k})\, \widetilde{\mathbb{K}}(\vec{k}, \vec{k}')\, \tilde{\varphi}(\vec{k}') \quad ,$$

where $\widetilde{\mathbb{K}}(\vec{k}, \vec{k}')$ *is the operator that "connects"* $\tilde{\varphi}(\vec{k})$ *to* $\tilde{\varphi}(\vec{k}')$*. This operator is anti-diagonal in the case of the* free *theory.*

With this knowledge, it can be shown that

$$\left\langle \tilde{\varphi}(\vec{k})\, \tilde{\varphi}(-\vec{k}) \right\rangle = \left\langle \left|\tilde{\varphi}(\vec{k})\right|^2 \right\rangle = \frac{1}{N^d \left[\rho^2(\vec{k}) + \alpha\right]} \quad .$$

Now, it is not difficult to see that,

$$\mathscr{I} = \int [d\tilde{\varphi}]\, e^{-\mathcal{S}_0[\tilde{\varphi}]} = \int [d\tilde{\varphi}]\, e^{-\frac{N^d}{2} \sum_{\vec{k}} (\rho^2(\vec{k}) + \alpha) \left|\tilde{\varphi}(\vec{k})\right|^2},$$

$$\therefore \quad \mathscr{I} = \prod_{\vec{k}} \sqrt{\frac{2\,\pi}{N^d \left[\rho^2(\vec{k}) + \alpha\right]}} \quad .$$

Thus, it follows that,

$$\left\langle \left|\tilde{\varphi}(\vec{k})\right|^{2n} \right\rangle = -2 \left[\frac{\partial}{\partial\{N^d(\rho^2(\vec{k}) + \alpha)\}}\right]^n \mathscr{I}$$

2.2 Correlation Functions

All the information of a given theory is contained within its correlation functions. Thus, it turns out to be essential a knowledge of these objects. In the following 2 sub-subsections this issue shall be addressed.

2.2.1 Factorization of the Correlation Functions

Correlation functions can be defined either in configuration or in momentum space

$$g(\vec{x}_1,\ldots,\vec{x}_n) = \langle\varphi(\vec{x}_1)\cdots\varphi(\vec{x}_n)\rangle \ ,$$
$$\tilde{g}(\vec{k}_1,\ldots,\vec{k}_n) = \left\langle\varphi(\vec{k}_1)\cdots\varphi(\vec{k}_n)\right\rangle \ .$$

Now, it is not difficult to see (symmetry arguments) that

$$\begin{aligned} & g(\vec{x}_1) = \langle\varphi(\vec{x}_1)\rangle \equiv 0 \ , \\ \Rightarrow \quad & g(\vec{x}_1,\ldots,\vec{x}_{2n+1}) = 0 \ , \\ \Rightarrow \quad & \tilde{g}(\vec{k}_1,\ldots,\vec{k}_{2n+1}) = 0 \ . \end{aligned}$$

Furthermore, this argument can be *refined*, in momentum space, to the following statement:

$$\tilde{g}(\vec{k}_1,\vec{k}_2) = \left\langle\tilde{\varphi}(\vec{k}_1)\,\tilde{\varphi}(\vec{k}_2)\right\rangle = \delta^d_{\vec{k}_1,-\vec{k}_2} \ ,$$

i.e., the incoming momentum, $(\tilde{\varphi}(\vec{k}_1))$, has to be *conserved* $\Rightarrow \tilde{\varphi}(-\vec{k}) = \tilde{\varphi}^*(\vec{k})$. Therefore, each $\tilde{\varphi}(\vec{k})$ term must be paired with a factor of $\tilde{\varphi}(-\vec{k})$ in order for the expectation value not to vanish. Thus, in momentum space,

$$\left\langle\left|\tilde{\varphi}(\vec{k}_1)\right|^2\,\left|\tilde{\varphi}(\vec{k}_2)\right|^2\right\rangle = \left\langle\left|\tilde{\varphi}(\vec{k}_1)\right|^2\right\rangle\cdot\left\langle\left|\tilde{\varphi}(\vec{k}_2)\right|^2\right\rangle \ ; \ \vec{k}_1\neq\vec{k}_2 \ .$$

This result says that, in the lattice, all correlation functions of an even number of fields can be written in terms of the 2-point correlation function and, with the aid of the results obtained in the end of the last section and the above one, it can be seen that,

$$\left\langle\left|\tilde{\varphi}(\vec{k})\right|^{2n}\right\rangle \equiv (2n-1)!!\,\left(\left|\tilde{\varphi}(\vec{k})\right|^2\right)^n \ , \ \ \tilde{\varphi}(\vec{k})\in\mathbb{R},$$
$$\left\langle\left|\tilde{\varphi}(\vec{k})\right|^{2n}\right\rangle \equiv (n!)\,\left(\left|\tilde{\varphi}(\vec{k})\right|^2\right)^n \ , \ \ \tilde{\varphi}(\vec{k})\in\mathbb{C}.$$

2.2.2 Structure of the Two-Point Function

In light of the results of the previous sub-subsection, the problem that has to be solved is given by,

$$g(\vec{x}_1,\vec{x}_2) \equiv \langle \varphi(\vec{x}_1)\,\varphi(\vec{x}_2)\rangle = \sum_{\vec{k}} e^{-i\frac{2\pi}{N}\vec{k}\cdot(\vec{x}_1-\vec{x}_2)} \left\langle \left|\tilde{\varphi}(\vec{k})\right|^2 \right\rangle \quad .$$

And, for the *free* theory,

$$g(\vec{x}_1,\vec{x}_2) = \sum_{\vec{k}} \frac{e^{-i\frac{2\pi}{N}\vec{k}\cdot(\vec{x}_1-\vec{x}_2)}}{N^d\left[\rho^2(\vec{k})+\alpha\right]} \quad .$$

From the theory of Fourier Transforms, it is known that the transform of an *even*, *real* function is, itself, an *even*, *real* function. Thus, $g(\vec{x}_1,\vec{x}_2) \in \mathbb{R}$ and is *even*. Furthermore, in a lattice with periodic boundary conditions, the symmetry implies that $g(\vec{x}_1,\vec{x}_2) = g(|\vec{x}_1-\vec{x}_2|)$.

2.3 Block Variables

Block variables were first introduced by Kadanoff in the study of statistical spin systems. Nowadays, their definition is central in the definition of a QFT because they determine what types of physical observables can be measured. They are motivated by the fact that, in order to perform a measurement, it is noted that it is impossible to achieve an instantaneous type of measurement. Something like this (instantaneous measurement) would imply that the measuring device would have to have a vanishing wavelength, thus it should consist of infinite quantuns of energy. Therefore, any type of measuring device will *always* measure the field(s) over some finite region, as tiny as it may be.

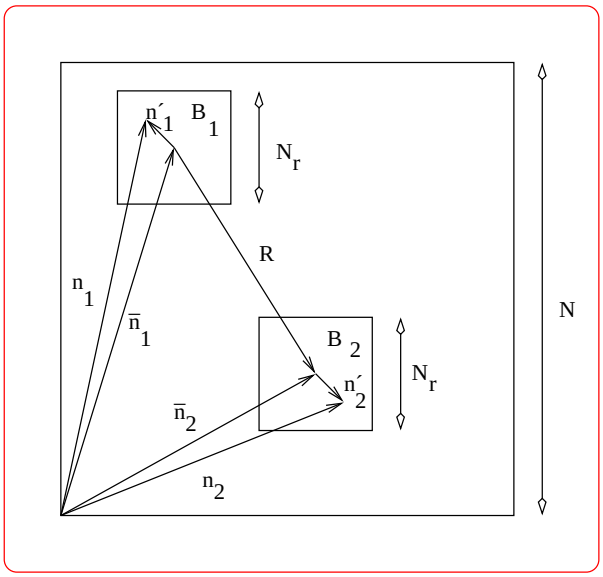

Figure 2: Cubic lattice with cubic blocks.

Definition 2.3. *Block Variables are (some sort of) averages over the fields in finite regions of spacetime (lattice). Those regions are called blocks.*

Usually, for the lattice formulation, a simple arithmetic average over the fields is good enough. However, there is no *a priori* reason for this to be a better choice than any other else made. Even non-linear choices might turn out to be reasonable.

Effectively, this will add a multiplicative factor to the Green's functions, called *form factor*.

Notation. *For the sake of clarity, in this section, note that instead of writing $\vec{n}$, $\vec{k}$, it shall be written $\mathbf{n}$ and $\mathbf{k}$ and a bar shall denote average over a block, $\bar{\mathbf{n}}$, $\bar{\mathbf{k}}$. Also, N_r is the number of sites inside one block and $\mathcal{B}$ shall denote a block itself.*

Thus, let us start figuring out how the block variables affects a given model.

$$\begin{aligned}\bar{\varphi}(\bar{\mathbf{n}}) &= \frac{1}{N_r^d}\sum_{\mathbf{n}\in\mathcal{B}}\varphi(\mathbf{n})\ ,\\ \bar{\mathbf{n}} &= \frac{1}{N_r^d}\sum_{\mathbf{n}\in\mathcal{B}}\mathbf{n}\ .\end{aligned}$$

Thus, the [dimensionless] block propagator is given by,

$$\begin{aligned}g_r(\bar{\mathbf{n}}_1,\bar{\mathbf{n}}_2) &= \langle\bar{\varphi}(\bar{\mathbf{n}}_1)\,\bar{\varphi}(\bar{\mathbf{n}}_2)\rangle\ ,\\ g_r(\mathbf{R}) &= \frac{1}{N^d}\sum_{\mathbf{k}}\frac{-i\frac{2\pi}{N}\mathbf{k}\cdot\mathbf{R}}{\rho^2(\mathbf{k})+\alpha}\left|f_r^d(\mathbf{k})\right|^2\ ,\\ f_r^d(\mathbf{k}) &= \frac{1}{N_r^d}\sum_{\mathbf{n}'\in\mathcal{B}}e^{-i\frac{2\pi}{N}\mathbf{k}\cdot\mathbf{n}'}\ .\end{aligned}$$

Fourier Transforming the above expression,

$$\tilde{g}_r(\mathbf{k}) = \frac{1}{N^d}\frac{\left|f_r^d(\mathbf{k})\right|^2}{\rho^2(\mathbf{k})+\alpha}\ .$$

Note that, as opposed to what happens in configuration space, there is no need to change coordinates in momentum space. Also, $\left|f_r^d(\mathbf{k})\right|\leqslant 1$, thus, in momentum space, the propagator is always less than or equal to the propagator of the fundamental field.

2.4 External Sources in the Quantum Theory

The model now will be given by a slightly modified free action, namely

$$\mathbb{S}_0^j[\varphi] = \frac{1}{2}\sum_l(\Delta_l\,\varphi)^2+\frac{\alpha}{2}\sum_s\varphi^2(s)-\sum_s j(s)\,\varphi(s)\ .$$

Much in the same way of what happened at the classical level, one can see that the symmetry $\varphi\mapsto-\varphi$ is lost in this action, thus is is expected that $v=\langle\varphi\rangle$ does not vanish anymore. Therefore, let us define a new variable — the translated field — by $\varphi'\equiv\varphi-v$. It is easy to see that, for this new variable, $\langle\varphi'\rangle\equiv 0$, by definition. Also, it should be noted that, once this substitution is carried out in the action, all the terms

that depend *only* on v will not contribute to the theory, because upon averaging of the observables those terms should drop out of the integral (once the integral will be over φ') and cancel out. Thus, it is not needed to worry about those terms.

Again, using the principle of least action (in its lattice version), it is possible to show that,

$$\Big[-\triangle + \alpha \Big]\, v(\vec{n}) + j(\vec{n}) = 0 \ ,$$

and, in momentum space

$$\tilde{v}(\vec{k}) = \frac{\tilde{j}(\vec{k})}{\rho^2(\vec{k}) + \alpha} \ .$$

It is also possible to show that,

$$\begin{aligned} v(\vec{n}) &= \sum_{\vec{n}'} K(\vec{n},\vec{n}')\, j(\vec{n}') \ , \\ K(\vec{n},\vec{n}') &= \langle \varphi(\vec{n})\, \varphi(\vec{n}') \rangle = g(\vec{n},\vec{n}') \ . \end{aligned}$$

2.5 Functional Generators in the Lattice

The basic functional that is desired to define[1] is a functional of the external source, j, denoted by $\mathscr{Z}[j]$.

Definition 2.4. *The* Partition Function *of the model is given by*

$$\mathscr{Z}[j] = \Big\langle e^{\sum_{\vec{n}} j(\vec{n})\, \varphi(\vec{n})} \Big\rangle = \frac{\int e^{-\mathcal{S}_j}\, [d\varphi]}{\int e^{-\mathcal{S}}\, [d\varphi]} \ , \tag{10}$$

where $\mathcal{S}_j[\varphi] = \mathcal{S}[\varphi] - \sum_s j(s)\, \varphi(s)$*, i.e., the source term is being separated from the rest of the action.*

Remark. *Is is not difficult to see that* $\mathscr{Z}[j]$ *is the ratio between both measures: one in the presence of* j *and the other one without it. Also note that a normalization for* $\mathscr{Z}[j]$ *is given implicitly, namely that* $\mathscr{Z}[j]|_{j=0} = 1$*.*

Notation. *It shall be used the letter* $\mathfrak{d}$ *to denote* functional *differentiation in the lattice, what is nothing more than a simple partial differentiation.*

[1] As usual, it is considered a hypercubic lattice with periodic boundary conditions.

Thus, in analogy with what is done in the continuum,

$$\frac{\mathfrak{d} j(\vec{n}_1)}{\mathfrak{d} j(\vec{n}_2)} = \delta^d_{\vec{n}_1,\vec{n}_2} \ ,$$

$$\Rightarrow \quad \frac{\mathfrak{d}^n \mathscr{Z}[j]}{\mathfrak{d} j_1 \cdots \mathfrak{d} j_n} = \frac{\int [\varphi_1 \cdots \varphi_n]\, e^{-\mathcal{S}_j}\, [d\varphi]}{\int e^{-\mathcal{S}}\, [d\varphi]} \ ,$$

$$\therefore \quad \left.\frac{\mathfrak{d}^n \mathscr{Z}[j]}{\mathfrak{d} j_1 \cdots \mathfrak{d} j_n}\right|_{j=0} = \langle \varphi_1 \cdots \varphi_n \rangle = g_{1,\ldots,n} \ .$$

Now, the correlation functions of this model are taken in the measure of $\mathcal{S}_j$, and are given by,

$$\frac{\mathfrak{d}^n}{\mathfrak{d} j_1 \cdots \mathfrak{d} j_n} \log\Big[\mathscr{Z}[j]\Big] = g^j_{1,\ldots,n} \ ,$$

what motivates the definition of

$$\mathscr{W}[j] = \log\Big[\mathscr{Z}[j]\Big] \ ,$$

$$\Rightarrow \quad \frac{\mathfrak{d}^n}{\mathfrak{d} j_1 \cdots \mathfrak{d} j_n} \mathscr{W}[j] = g^j_{1,\ldots,n} \ .$$

This new functional, $\mathscr{W}[j]$, generates a different set of correlation functions, known as *connected* correlation functions. Thus, for example, the second derivative of $\mathscr{W}[j]$ should give the *connected* propagator and so on and so forth. Completely calculating any of those functionals is *equivalent* to solving the theory completely. As an example of this, let us calculate the free field.

$$\mathscr{Z}[j] = e^{\mathscr{W}[j]} = \frac{\int e^{\sum_{\vec{n}} j(\vec{n})\,\varphi(\vec{n})}\, e^{-\mathcal{S}_0}\, [d\varphi]}{\int e^{-\mathcal{S}_0} [d\varphi]} \ ,$$

$$\therefore \quad \mathscr{Z}[j] = e^{\mathscr{W}[j]} = \mathscr{Z}[\tilde{j}] = e^{\mathscr{W}[\tilde{j}]} \ .$$

Now,

$$\mathscr{F}\left\{e^{\sum_{\vec{n}} j(\vec{n})\,\varphi(\vec{n})}\right\} = e^{N^d \sum_{\vec{k}} \tilde{j}(\vec{k})\tilde{\varphi}(-\vec{k})}\ ,\quad \tilde{\varphi}(-\vec{k}) = \tilde{\varphi}^*(\vec{k})\ ,$$

$$\therefore\quad \mathscr{Z}[\tilde{j}] = \prod_{\vec{k}} \sqrt{\frac{N^d\left[\rho^2(\vec{k})+\alpha\right]}{2\,\pi}}\cdot \underbrace{\int e^{N^d \sum_{\vec{k}} \tilde{j}(\vec{k})\tilde{\varphi}(-\vec{k})}\, e^{-S_0}\,[d\varphi]}_{=\beta}\ ,$$

$$\beta = \int e^{\frac{N^d}{2}\sum_{\vec{k}}[\tilde{\varphi}^*(\vec{k})(\rho^2(\vec{k})+\alpha)\tilde{\varphi}(\vec{k})+2\tilde{j}(-\vec{k})\tilde{\varphi}(\vec{k})]}\,[d\varphi]\ ,$$

$$\Rightarrow\quad \beta = \prod_{\vec{k}} \sqrt{\frac{2\,\pi}{N^d\left[\rho^2(\vec{k})+\alpha\right]}}\cdot e^{-\sum_{\vec{k}} \frac{\tilde{j}(-\vec{k})\tilde{j}(\vec{k})}{N^d[\rho^2(\vec{k})+\alpha]}}\ ,\ \tilde{j}(-\vec{k}) = \tilde{j}^*(\vec{k})\ .$$

$$\therefore\quad \mathscr{Z}[j] = e^{\mathscr{W}[\tilde{j}]} = e^{\frac{1}{N^d}\sum_{\vec{k}} \frac{|\tilde{j}(\vec{k})|^2}{\rho^2(\vec{k})+\alpha}}\ .$$

$$\therefore\quad \mathscr{W}[\tilde{j}] = \frac{1}{N^d}\sum_{\vec{k}} \frac{\left|\tilde{j}(\vec{k})\right|^2}{\rho^2(\vec{k})+\alpha}\ ,$$

thus, it is easy to see that,

$$\frac{\mathfrak{d}^n}{\mathfrak{d}\tilde{j}_1\cdots\mathfrak{d}\tilde{j}_n}\mathscr{W}[\tilde{j}] \equiv 0,\ \ \forall\ n \geqslant 3\ .$$

$$\therefore\quad \left.\frac{\mathfrak{d}\mathscr{W}[\tilde{j}]}{\mathfrak{d}\tilde{j}(\vec{k})}\right|_{\tilde{j}=0} = \left\langle\tilde{\varphi}(\vec{k})\right\rangle = \tilde{g}_1^{\text{conn}}\ ,$$

$$= \frac{2}{N^d}\left.\frac{\tilde{j}(\vec{k})}{\rho^2(\vec{k})+\alpha}\right|_{\tilde{j}=0}\ ,$$

$$\Rightarrow\quad \tilde{g}_1^{\text{conn}} \equiv 0:\ \text{No sources!}$$

Analogously,

$$\left.\frac{\mathfrak{d}^2\mathscr{W}[\tilde{j}]}{\mathfrak{d}\tilde{j}(\vec{k}_1)\,\mathfrak{d}\tilde{j}(\vec{k}_2)}\right|_{\tilde{j}=0} = \frac{1}{N^d}\,\frac{\delta_{-\vec{k}_1,\vec{k}_2}}{\rho^2(\vec{k})+\alpha}\ ,$$

where $\tilde{g}^{\text{conn}}$ stands for the *connected* Green's function.

Now, either motivated by Statistical Mechanics or by the fact that there is a well defined relation between $g_1^j \equiv \varphi_c[j] = \langle\varphi(\vec{n})\rangle_j$[m] and, thus, it *could* be possible[n] to write

[m] The reader should note that, the $\varphi_c[j] = \langle\varphi(\vec{n})\rangle_j$ defined here, plays the same role as $v[j]$ played in the last subsection. However, this "new" v_j is an average with respect to the measure *with* the source j! (As opposed to what happened before, in which case v was an average with respect to the *free* action.)

[n] Once again, there is a very tricky detail that *has* to be taken into account here. This one is given by a *functional* version of the *Inverse Function Theorem* that, basically, requires that $(\mathfrak{d}\varphi_c/\mathfrak{d}j) \neq 0$ so that it is possible to invert this functional.

j as a functional of φ_c instead. In this case, $\mathscr{W}$ would be written in terms of φ_c. Thus, one finds that,

$$\varphi_c[j] = \frac{\mathfrak{d}\mathscr{W}[j]}{\mathfrak{d}j} \ ,$$

and, writing the *functional differential*,

$$\mathrm{d}\mathscr{W} = \sum_{\vec{n}} \frac{\mathfrak{d}\mathscr{W}[j]}{\mathfrak{d}j(\vec{n})} \ \mathrm{d}j(\vec{n}) = \sum_{\vec{n}} \ \varphi_c(\vec{n}) \ \mathrm{d}j(\vec{n}) \ \ .$$

Now, a *new* functional can be defined via a functional version of a *Legendre* transform:

$$\Gamma[\varphi_c] = \sum_{\vec{n}} \ j(\vec{n}) \, \varphi_c(\vec{n}) - \mathscr{W}[j] \ \ .$$

The important point to note is,

$$\left\langle \sum_{\vec{n}} j(\vec{n})\varphi(\vec{n}) \right\rangle = \sum_{\vec{n}} \langle j(\vec{n})\varphi(\vec{n})\rangle = \sum_{\vec{n}} j(\vec{n}) \ \langle\varphi(\vec{n})\rangle = \sum_{\vec{n}} j(\vec{n}) \, \varphi_c(\vec{n}) \ ,$$

and, now, it is noted that,

$$\mathrm{d}\Gamma = \sum_{\vec{n}} j(\vec{n}) \ \mathrm{d}\varphi_c(\vec{n}) \ \ .$$

In this way, the conclusion is that Γ is only a functional of φ_c. Also,

$$\frac{\mathfrak{d}\Gamma[\varphi_c]}{\mathfrak{d}\varphi_c} = j \ \ .$$

Therefore, now, it can be written that,

$$\mathscr{Z}[j] = e^{\mathscr{W}[j]} = e^{\sum_{\vec{n}} j(\vec{n})\varphi_c(\vec{n})} \, e^{-\Gamma[\varphi_c]} = \frac{\int e^{\sum_{\vec{n}} j(\vec{n})\varphi(\vec{n})} \, e^{-\mathcal{S}_0} \, [d\varphi]}{\int e^{-\mathcal{S}_0} \, [d\varphi]} \ \ ,$$

$$\therefore \ e^{-\Gamma[\varphi_c]} \equiv \left\langle e^{\sum_{\vec{n}} j(\vec{n})[\varphi(\vec{n}) - \varphi_c(\vec{n})]} \right\rangle = \frac{\int e^{\sum_{\vec{n}} j(\vec{n})[\varphi(\vec{n}) - \varphi_c(\vec{n})]} \, e^{-\mathcal{S}_0} \, [d\varphi]}{\int e^{-\mathcal{S}_0} \, [d\varphi]} \ \ .$$

At this point, it is not difficult to notice that Γ is, somehow, related — in the form of a "classical action" — to the "classical field", φ_c.

Definition 2.5. *The functional $\Gamma[\varphi_c]$, as defined above, is called the* Effective Action*. And, the interpretation above, will turn out to be quite useful.*

Remark. *It should be noted that, the sources, j, that appear above should be treated as functionals: $j = j[\varphi_c]$.*

2.6 [Euclidean] Lattice Simulations

Let us now discuss how to apply the techniques discussed in sub-subsection 1.6.2 to the specific case of a QFT defined in a lattice of size N (hypercubic lattice) and dimension d.

This is completely equivalent to solving multiple integrals of graphs of functionals over a space of dimension N^d.

It is quite straightforward to see that

$$\langle \mathcal{O}[\varphi] \rangle = \frac{\int \mathcal{O}[\varphi]\, e^{-\mathcal{S}[\varphi]}\, [d\varphi]}{\int e^{-\mathcal{S}[\varphi]}\, [d\varphi]} \,,$$

$$\langle \mathcal{O}[\varphi] \rangle = \frac{1}{\|\Phi\|} \sum_{\varphi \in \Phi} \left\{ \frac{\left[\frac{\mathrm{Vol}(\mathcal{D})}{N^d}\right] \sum_{i=1}^{N^d} \mathcal{O}[\varphi^{[i]}]}{\left[\frac{\mathrm{Vol}(\mathcal{D})}{N^d}\right] \cdot N^d} \right\} ,$$

$$\therefore \quad \langle \mathcal{O}[\varphi] \rangle = \frac{1}{\|\Phi\|} \sum_{\varphi \in \Phi} \left\{ \frac{1}{N^d} \sum_{i=1}^{N^d} \mathcal{O}[\varphi^{[i]}] \right\} ,$$

where, $\|\Phi\|$ is the *cardinality* of the set Φ, i.e., the number of elements in the set. Now, once the set Φ is the set of all possible configurations of the field φ, its cardinality should be infinite, however, once what is being considered are lattice simulations, one should take into account a certain "number" of field configurations. This number turns out to be the norm of the set Φ. Thus, $[(1/\|\Phi\|) \sum_{\varphi \in \Phi}]$ indicates an average over the total number of field configurations.

In order to illustrate this, the free theory, $(\mathcal{S}_0)$, shall be considered.

Let us start with a configuration of the type $\varphi(\vec{n}) = C,\ \forall \vec{n} \in \mathbb{L}$, where $C \in \mathbb{R}$ is any constant value (for instance, the solution for the classical theory: $C = 0$). This is usually known as a *cold* or *oriented* (non-random), because from a statistical mechanics point of view, these are characteristic states of low temperature regimes. Then, the lattice is swept — in any given order — and the following analysis is done (in a per site basis): $\varphi \mapsto \varphi' = \varphi + r \cdot \delta$, where $r \in [-1, 1]$ and $\delta \in [-\delta, \delta]$ are two uniformly distributed (over the indicated range) random numbers. This means that the old field is being taken into a new field which has homogeneous probability in an interval of radius δ around the old value. Now, before moving on the following site, the change in the action should be calculated and, upon this, it should be decided whether or not this change in the field should be accepted. In order to evaluate the change in the action, it should be noted that most of the terms will not be affected, because only the particular site where this change is being done is affected, all the other one remain intact. Thus, it is not difficult to see that:

$$\begin{aligned}\delta\mathcal{S}_0 &= \delta\sum_{\rm nn}\varphi(s)\varphi({\rm nn}) + \left(d+\frac{\alpha}{2}\right)\,\delta\left(\varphi^2(s)\right)\,,\\ &= \delta\varphi(s)\,\left[\sum_{\rm nn}\varphi({\rm nn}) + \left(d+\frac{\alpha}{2}\right)\,\left(\varphi'(s)+\varphi(s)\right)\right]\,,\end{aligned}$$

where the symbol "nn" stands for (labels) "nearest neighbor" and $\varphi'(s)$ is the new [tentative] value for the site s and $\varphi(s)$ is the previous (old) value of the field in the site s. Thus, the following step is the Metropolis test (i.e., $e^{-\Delta\mathcal{S}_0} \geqslant \beta,\quad \beta\in[0,1]$ and $\Delta\mathcal{S}_0 = \mathcal{S}_0[\varphi'] - \mathcal{S}_0[\varphi]$, is the difference between the "old" and the "new" action). It is worth noting that the variation of the action is proportional to the variation of the field and, as such, it is possible to control the rate of accepted changes in the action by controlling $\delta\varphi$. In this way, if the Metropolis test is true, the new value for the field is substituted, otherwise the following site is tested.

Appendix

In this appendix I shall lay out the codes that I first used in my simulations. They were written in (standard) FORTRAN 77 and compiled with `g77`.

Code 1: fsqft.f

```
C--------------------------------------------------------------------------
C     Time-stamp: <01/11/09 15:13:16 danieldf>
C--------------------------------------------------------------------------
C     BeGiN
C--------------------------------------------------------------------------
      PROGRAM FSQFT
C--------------------------------------------------------------------------
      INTEGER dim, temp, maxit, cntr
      PARAMETER (dim=7, temp=1E3, maxit=1E2)
      DOUBLE PRECISION field(dim,dim,dim,dim), avrgobs(dim)
C--------------------------------------------------------------------------
      CALL CLASSOL (FIELD, DIM)
      CALL THERMAL (FIELD, DIM, TEMP)
C--------------------------------------------------------------------------
      DO cntr = 1, maxit
         CALL QFLUCT (FIELD, DIM)
         CALL GREENSS (FIELD, DIM, AVRGOBS)
      END DO
C--------------------------------------------------------------------------
      DO cntr = 1, dim
         WRITE(UNIT=*, FMT='(1I3,1X,D15.7)') cntr, avrgobs(cntr)
      END DO
C--------------------------------------------------------------------------
      END
C--------------------------------------------------------------------------
C     EnD
C--------------------------------------------------------------------------
```

Code 2: classol.f

```
C-----------------------------------------------------------------------
C     Time-stamp: <01/11/05 11:25:23 danieldf>
C-----------------------------------------------------------------------
C     BeGiN
C-----------------------------------------------------------------------
      SUBROUTINE CLASSOL (FIELD, DIM)
C-----------------------------------------------------------------------
      INTEGER dim, i, j, k, l
      DOUBLE PRECISION field(dim,dim,dim,dim), grnd
      EXTERNAL grnd
C-----------------------------------------------------------------------
      DO i = 1, dim
         DO j = 1, dim
            DO k = 1, dim
               DO l = 1, dim
                  field(i, j, k, l) = grnd()
               END DO
            END DO
         END DO
      END DO
C-----------------------------------------------------------------------
      RETURN
      END
C-----------------------------------------------------------------------
C     EnD
C-----------------------------------------------------------------------
```

Code 3: thermal.f

```
C-----------------------------------------------------------------------
c     Time-stamp: <01/10/25 11:06:11 danieldf>
C-----------------------------------------------------------------------
c     BeGiN
C-----------------------------------------------------------------------
      SUBROUTINE THERMAL (FIELD, DIM, TEMP)
C-----------------------------------------------------------------------
      INTEGER dim, temp, cntr
      DOUBLE PRECISION field(dim,dim,dim,dim)
C-----------------------------------------------------------------------
      DO cntr = 1, temp
         CALL QFLUCT (FIELD, DIM)
      END DO
C-----------------------------------------------------------------------
      RETURN
      END
C-----------------------------------------------------------------------
c     EnD
C-----------------------------------------------------------------------
```

Code 4: qfluct.f

```
C-----------------------------------------------------------------------
C     Time-stamp: <01/11/05 11:43:08 danieldf>
C     BeGiN
C-----------------------------------------------------------------------
      SUBROUTINE QFLUCT (FIELD, DIM)
C-----------------------------------------------------------------------
      INTEGER dim, i, j, k, l, aceitacao
      DOUBLE PRECISION grnd, mtrandn, field(dim,dim,dim,dim)
      DOUBLE PRECISION novo, velho, prob
      DOUBLE PRECISION raio, delta, varacao, nn, metrop
      DOUBLE PRECISION sup, halfsup, half
      PARAMETER ( half=0.5D0, sup=2.0D0 )
      PARAMETER ( halfsup=(half*sup) )
      EXTERNAL grnd, mtrandn
C-----------------------------------------------------------------------
      nn = 0.0D0
      varacao = 0.0D0
      aceitacao = 0
      DO i = 1, dim
         DO j = 1, dim
            DO k = 1, dim
               DO l = 1, dim
                  CALL NEARNGHBR (FIELD, I, J, K, L, DIM, NN)
                  prob = grnd()
                  raio = grnd()
                  delta = mtrandn(sup)
                  IF ( raio .LE. half ) raio = -raio
                  IF ( delta .LE. halfsup ) delta = -delta
                  velho = field(i,j,k,l)
                  novo = velho + ( raio*delta )
                  CALL DELTAS (NN, NOVO, VELHO, VARACAO)
                  metrop = DEXP( -varacao )
                  IF ( metrop .GE. prob ) THEN
                     field(i,j,k,l) = novo
                     aceitacao = aceitacao + 1
                  END IF
               END DO
            END DO
         END DO
      END DO
C-----------------------------------------------------------------------
      RETURN
      END
C-----------------------------------------------------------------------
C     EnD
C-----------------------------------------------------------------------
```

Code 5: nearnghbr.f

```
C-----------------------------------------------------------------------
c     Time-stamp: <01/11/02 11:54:32 danieldf>
c     BeGiN
C-----------------------------------------------------------------------
      SUBROUTINE NEARNGHBR (FIELD, I, J, K, L, DIM, NN)
C-----------------------------------------------------------------------
      INTEGER dim, i, j, k, l, in, ip, jn, jp, kn, kp, ln, lp
      DOUBLE PRECISION field(dim,dim,dim,dim), nn
C-----------------------------------------------------------------------
      ip = i + 1
      in = i - 1
      jp = j + 1
      jn = j - 1
      kp = k + 1
      kn = k - 1
      lp = l + 1
      ln = l - 1
      CALL PBC (in,i,ip,jn,j,jp,kn,k,kp,ln,l,lp,dim)
      nn = ( field(ip,j,k,l) + field(i,jp,k,l) +
     &     field(i,j,kp,l) + field(i,j,k,lp) +
     &     field(in,j,k,l) + field(i,jn,k,l) +
     &     field(i,j,kn,l) + field(i,j,k,ln) )
C-----------------------------------------------------------------------
      RETURN
      END
C-----------------------------------------------------------------------
c     EnD
C-----------------------------------------------------------------------
```

Code 6: pbc.f

```
1  C-----------------------------------------------------------------------
2  c     Time-stamp: <01/11/05 10:06:34 danieldf>
3  C-----------------------------------------------------------------------
4  c     BeGiN
5  C-----------------------------------------------------------------------
6        SUBROUTINE PBC (IN, I, IP, JN, J, JP, KN, K, KP, LN, L, LP, DIM)
7  C-----------------------------------------------------------------------
8        INTEGER in, i, ip, jn, j, jp, kn, k, kp, ln, l, lp, dim
9  C-----------------------------------------------------------------------
10       IF ( i .EQ. 1 ) in = dim
11       IF ( j .EQ. 1 ) jn = dim
12       IF ( k .EQ. 1 ) kn = dim
13       IF ( l .EQ. 1 ) ln = dim
14       IF ( i .EQ. dim ) ip = 1
15       IF ( j .EQ. dim ) jp = 1
16       IF ( k .EQ. dim ) kp = 1
17       IF ( l .EQ. dim ) lp = 1
18 C-----------------------------------------------------------------------
19       RETURN
20       END
21 C-----------------------------------------------------------------------
22 c     EnD
23 C-----------------------------------------------------------------------
```

Code 7: deltas.f

```
C-----------------------------------------------------------------------
c     Time-stamp: <01/11/05 15:40:29 danieldf>
c     BeGiN
C-----------------------------------------------------------------------
      SUBROUTINE DELTAS (NN, NOVO, VELHO, VARACAO)
C-----------------------------------------------------------------------
      DOUBLE PRECISION nn, novo, velho, varacao
      DOUBLE PRECISION half, alphazr, d, cnt, nearnghbr, deltacampo
      PARAMETER ( d=4.0D0, half=0.5D0, alphazr=10.0D0 )
C-----------------------------------------------------------------------
      nearnghbr = nn
      deltacampo = ( novo - velho )
      cnt = ( d + ( half*alphazr ) )*( novo + velho )
      varacao = deltacampo*( nearnghbr + cnt )
C-----------------------------------------------------------------------
      RETURN
      END
C-----------------------------------------------------------------------
c     EnD
C-----------------------------------------------------------------------
```

Code 8: greenss.f

```
c--------------------------------------------------------------------------
c     Time-stamp: <01/11/06 13:19:01 danieldf>
c--------------------------------------------------------------------------
c     BeGiN
c--------------------------------------------------------------------------
      SUBROUTINE GREENSS (FIELD, DIM, AVRGOBS)
c--------------------------------------------------------------------------
      INTEGER dim, i, j, k, l, cntr, norm
      INTEGER ip, in, jp, jn, kp, kn, lp, ln
      DOUBLE PRECISION field(dim,dim,dim,dim)
      DOUBLE PRECISION obs(dim,dim,dim,dim,dim)
      DOUBLE PRECISION avrgnn(dim,dim,dim,dim,dim), avrgobs(dim)
      DOUBLE PRECISION nnn, dnorm
      PARAMETER (nnn=(2.0D0*4.0D0))
c--------------------------------------------------------------------------
      norm = (dim*dim*dim*dim)
      dnorm = DBLE( norm )
c--------------------------------------------------------------------------
      CALL ZAVRG (AVRGNN, DIM)
      CALL ZOBS (OBS, DIM)
      CALL ZAVRGOBS (AVRGOBS, DIM)
c--------------------------------------------------------------------------
      DO cntr = 1, dim
         DO i = 1, dim
            DO j = 1, dim
               DO k = 1, dim
                  DO l = 1, dim
                     ip = i + cntr
                     in = i - cntr
                     jp = j + cntr
                     jn = j - cntr
                     kp = k + cntr
                     kn = k - cntr
                     lp = l + cntr
                     ln = l - cntr
                     CALL PBCAVRG (in,ip,jn,jp,kn,kp,ln,lp,dim)
                     avrgnn(cntr,i,j,k,l) = ( field(ip,j,k,l) +
     $                    field(i,jp,k,l) + field(i,j,kp,l) +
     $                    field(i,j,k,lp) + field(in,j,k,l) +
     $                    field(i,jn,k,l) + field(i,j,kn,l) +
     $                    field(i,j,k,ln) )/( nnn )
                     obs(cntr,i,j,k,l) = obs(cntr,i,j,k,l) + ( field(i,j
     $                    ,k,l)*avrgnn(cntr,i,j,k,l) )
                  END DO
               END DO
            END DO
         END DO
      END DO
c--------------------------------------------------------------------------
      DO cntr = 1, dim
         DO i = 1, dim
            DO j = 1, dim
               DO k = 1, dim
                  DO l = 1, dim
                     avrgobs(cntr) = avrgobs(cntr) + ( obs(cntr,i,j,k,l)
     $                    /dnorm )
                  END DO
               END DO
            END DO
         END DO
c         write(*,*) cntr, avrgobs(cntr)
      END DO
c--------------------------------------------------------------------------
      RETURN
      END
c--------------------------------------------------------------------------
c     EnD
c--------------------------------------------------------------------------
```

Code 9: pbcavrg.f

```
1  C-----------------------------------------------------------------------
2  C     Time-stamp: <01/11/05 14:33:17 danieldf>
3  C-----------------------------------------------------------------------
4  C     BeGiN
5  C-----------------------------------------------------------------------
6        SUBROUTINE PBCAVRG (IN, IP, JN, JP, KN, KP, LN, LP, DIM)
7  C-----------------------------------------------------------------------
8        INTEGER in, ip, jn, jp, kn, kp, ln, lp, dim, start, inicio
9        PARAMETER (inicio=0, start=1)
10 C-----------------------------------------------------------------------
11       IF ( in .EQ. inicio ) in = start
12       IF ( in .LT. inicio ) in = dim + in
13       IF ( jn .EQ. inicio ) jn = start
14       IF ( jn .LT. inicio ) jn = dim + jn
15       IF ( kn .EQ. inicio ) kn = start
16       IF ( kn .LT. inicio ) kn = dim + kn
17       IF ( ln .EQ. inicio ) ln = start
18       IF ( ln .LT. inicio ) ln = dim + ln
19 C-----------------------------------------------------------------------
20       IF ( ip .GT. dim ) ip = ip - dim
21       IF ( jp .GT. dim ) jp = jp - dim
22       IF ( kp .GT. dim ) kp = kp - dim
23       IF ( lp .GT. dim ) lp = lp - dim
24 C-----------------------------------------------------------------------
25       RETURN
26       END
27 C-----------------------------------------------------------------------
28 C     EnD
29 C-----------------------------------------------------------------------
```

References

[1] Jorge L. deLyra, *Teoria Quântica de Campos: Uma Abordagem Radical na Rede*, `http://latt.if.usp.br/bazar/Index_engl.html`