

# Computational Field Theory and Pattern Formation

Raúl Toral

Departament de Física, Universitat de les Illes Balears, 07071-Palma de  
Mallorca, Spain

## 1. Monte Carlo Methods

### 1.1. Introduction

In many occasions we need to compute the value of a high-dimensional integral of the form:

$$\int_{-\infty}^{\infty} dx_1 \dots \int_{-\infty}^{\infty} dx_N f_{\mathbf{x}}(x_1, \dots, x_N) G(x_1, \dots, x_N) \equiv \int dX f_{\mathbf{x}}(X) G(X) \quad (1.1)$$

here  $N$  is a *large* number and we have denoted by  $X$  the set  $(x_1, \dots, x_N)$  of real variables. The function  $f_{\mathbf{x}}(X)$  is non-negative and normalized, i.e.:

$$\begin{aligned} (i) \quad & f_{\mathbf{x}}(X) \geq 0 \\ (ii) \quad & \int f_{\mathbf{x}}(X) dX \equiv \int_{-\infty}^{\infty} dx_1 \dots \int_{-\infty}^{\infty} dx_N f_{\mathbf{x}}(x_1, \dots, x_N) = 1 \end{aligned} \quad (1.2)$$

In next chapters we will give specific examples of when one needs to perform this type of calculation. If the above integral can not be computed by analytical methods (which happens more often than desired), one has to resource to numerical methods. One must realize, however, that a simple extension of the standard methods used in small- $N$  numerical integration (Simpson rules, Gauss integration, etc. (Faires et al. 1993)) to compute numerically the value of the integral will fail for very high-dimensional integrals. For instance, let us take  $N = 100$  (and this is a rather small number for typical applications) and suppose we want to use Simpson methods. We need to generate a grid in the  $X$  space and sum up all the values of  $G(X)$  on the points of this grid. If we choose, say, a grid of 10 points per every coordinate, we will have  $10^{100}$  terms to add in the Simpson rule. If every addition takes  $10^{-12}$  s (and today's computers take much longer than that) the time needed for the calculation exceeds any astronomical unit of time (including the age of the universe). The situation does not improve

by diminishing the number of points in the integration grid. Even 2 points per integration variable is beyond the capability of any present (or future) computer.

The solution to this problem is given by application of the Monte Carlo techniques (Kalos et al. 1986; Binder 1992, Heermann 1986). For our particular example, they consist in interpreting  $f_{\hat{\mathbf{x}}}(X)$  as a probability density function of some random variable  $\hat{\mathbf{x}}$  and, consequently, the integral is nothing but the average value of the function  $G(X)$ ,

$$\langle G \rangle = \int dX f_{\hat{\mathbf{x}}}(X) G(X) \quad (1.3)$$

The key idea behind Monte Carlo integration is to approximate the previous average by a sample average:

$$\langle G \rangle \approx \mu_M[G] = \frac{1}{M} \sum_{k=1}^M G(X^{(k)}) \quad (1.4)$$

where  $X^{(k)}$ ,  $k = 1, \dots, M$  are values of the  $N$ -dimensional variable  $X$  distributed according to the probability density function  $f_{\hat{\mathbf{x}}}(X)$ . This is called the technique of **importance sampling**. The basic principle of sampling is the same one that allows politicians to have an accurate knowledge of the election results before election day by making random polls amongst a representative sample of the electors.

A more precise statement on the relation between the average  $\langle G \rangle$  and the sample average  $\mu_M[G]$  is given by the *central limit theorem* (Feller 1971). This theorem states that, if the values of  $X^{(k)}$  are statistically independent, then the sample average  $\mu_M$  tends in the limit  $M \rightarrow \infty$  to a Gaussian distribution of mean  $\langle G \rangle$  and variance  $\sigma^2[\mu_M]$  given by:

$$\sigma^2[\mu_M] = \frac{1}{M} \sigma^2[G] = \frac{1}{M} [\langle G^2 \rangle - \langle G \rangle^2] \quad (1.5)$$

It is customary to express the relation between  $\mu_M[G]$  and  $\langle G \rangle$  as:

$$\langle G \rangle = \mu_M[G] \pm \sigma[\mu_M] \quad (1.6)$$

which is to be interpreted in the usual statistical sense for the Gaussian distribution, i.e. that there is a 95.45% probability that  $\langle G \rangle$  lies in the interval  $(\mu_M[G] - 2\sigma[\mu_M], \mu_M[G] + 2\sigma[\mu_M])$ , 99.73% in the interval  $(\mu_M[G] - 3\sigma[\mu_M], \mu_M[G] + 3\sigma[\mu_M])$ , etc. In practice, the unknown variance  $\sigma^2[G]$  can be replaced by the sample variance:

$$\sigma^2[G] \approx \frac{1}{M} \sum_{k=1}^M G(X^{(k)})^2 - \left( \frac{1}{M} \sum_{k=1}^M G(X^{(k)}) \right)^2 \quad (1.7)$$

According to relation (1.5) the error in a Monte Carlo integration decreases as the inverse square root of the number of samplings  $M$ . In the case that the values  $X^{(k)}$  are not independent, two major modifications are needed:

(a) The variance of the sample average is given by:

$$\sigma^2[\mu_M] = \frac{\sigma^2[G]}{M}(2\tau_G + 1) \quad (1.8)$$

where the *autocorrelation time*  $\tau_G$  is given (in the limit of large  $M$ ) by:

$$\tau_G = \sum_{k=1}^{\infty} \rho_G(k) \quad (1.9)$$

here the *normalized autocorrelation function*  $\rho_G(k)$  is defined as:

$$\rho_G(k) = \frac{\langle G(X^{(k')})G(X^{(k+k')}) \rangle - \langle G \rangle^2}{\sigma^2[G]} \quad (1.10)$$

The situation for non zero autocorrelation time can arise, for instance, when the value  $X^{(k)}$  depends on the value  $X^{(k-1)}$ . In our poll example, this can occur if the  $k$ -th person to be questioned lives in the neighbourhood of the  $(k-1)$ -th person, such that their social classes are likely to be similar. Intuitively,  $\tau_G$  measures the number of values of the sample  $X^{(k)}, X^{(k+1)}, \dots, X^{(k+\tau_G)}$  that we have to discard in order to consider that  $X^{(k)}$  and  $X^{(k+\tau_G)}$  are independent of each other. A problem of Monte Carlo methods is that, in many cases of interest,  $\tau_G$  becomes very large and grows with the number of variables  $N$ .

(b) The relation  $\langle G \rangle = \mu_M[G] \pm \sigma[\mu_M]$  has to be interpreted now according to Chebichev's theorem (Feller 1971):

$$P(|\mu_M[G] - \langle G \rangle| > k\sigma[\mu_M]) > 1 - \frac{1}{k^2} \quad (1.11)$$

i.e. the probability that  $\langle G \rangle$  lies in the interval  $(\mu_M[G] - 2\sigma[\mu_M], \mu_M[G] + 2\sigma[\mu_M])$  is *at least* 75%, that it lies in  $(\mu_M[G] - 3\sigma[\mu_M], \mu_M[G] + 3\sigma[\mu_M])$ , *at least* 88.89%, etc.

The above Monte Carlo importance sampling procedure is one of the most powerful methods available to compute high-dimensional integrals. In order to completely specify the method, though, we need a procedure to generate the values of the samples  $X^{(k)}$  that appear in the above formulae. These should be values of the  $N$ -dimensional variable  $(x_1, \dots, x_N)$  distributed according to the probability density function  $f_{\mathbf{x}}(X)$ . There are several methods devised to generate the required values  $X^{(k)}$ . Before we can explain the very powerful methods available we need to study in detail some simpler cases.

## 1.2. Uniform sampling

Let us start with a simple example. Consider the one-dimensional ( $N=1$ ) integral:

$$I = \int_0^1 dx \cos(x) \quad (1.12)$$

this is a particular case of Eq.(1.1) in which the function  $f_{\tilde{x}}(x)$  is equal to 1 in the interval  $(0, 1)$  and 0 otherwise. The name **uniform sampling** comes from the fact that the variable  $x$  is equally likely to take any value in the interval  $(0, 1)$  (or, generally speaking, in any finite interval  $(a, b)$ ). Although one could, in principle, devise some physical process that would produce values of a variable  $x$  uniformly distributed in the interval  $(0, 1)$ , it is preferable, for many reasons, to use simple algorithms that can be programmed on a computer and that produce "random-type" (or *pseudo-random*) numbers uniformly distributed in the interval  $(0, 1)$ . We will not discuss in these lectures the interesting topic of pseudo-random number generation. An excellent exposition can be found in references (Knuth 1981) and (James 1990). Let us simply assume that there exists a fortran function, which we will call *ran\_u()*, which returns an independent random number uniformly distributed in the interval  $(0, 1)$ . By using such a function, we can implement very easily the algorithm of uniform sampling:

```

m=1000
r=0.0
s=0.0
do 99 k=1,m
x=ran_u()
gk=g(x)
r=r+gk
s=s+gk*gk
99 continue
r=r/m
s=s/m-r*r
s=sqrt(s/m)
write(6,*) r,'+/-',s
end
function g(x)
g=cos(x)
return
end

```

The reader should find out the name given to the function *ran\_u()* is his own computer and run the above program checking that the error in the output value scales as the inverse square root of the number of samples  $M$ .

### 1.3. Importance sampling for $N=1$

In the  $N=1$  case, when the variable  $X = x$  has only one component, a simple and important theorem allows to generate  $x^{(k)}$  very easily. Recall that the probability distribution function  $F_{\hat{x}}(x)$  is defined by integration of the probability density function  $f_{\hat{x}}(x)$  as:

$$F_{\hat{x}}(x) = \int_{-\infty}^x f_{\hat{x}}(y) dy \quad (1.13)$$

**Theorem 1** *If  $f_{\hat{x}}(x)$  is a one-variable probability distribution function, then the variable  $u = F_{\hat{x}}(x)$  is uniformly distributed in the interval  $(0, 1)$ .*

As a consequence, we have the

**Corollary 1** *If  $u$  is uniformly distributed in the interval  $(0, 1)$ , then  $x = F_{\hat{x}}^{-1}(u)$  is distributed according to  $f_{\hat{x}}(x)$ .*

(The proof of this theorem is left as an exercise for the reader). This theorem reduces the problem of generating random numbers according to a given distribution  $f_{\hat{x}}(x)$  to the inversion of the corresponding probability distribution function. If, for instance, we want to use the method of importance sampling to compute the integral

$$I = \int_0^{\infty} dx e^{-x} x^2 \quad (1.14)$$

we can take  $f_{\hat{x}}(x) = e^{-x}$  if  $x \geq 0$ , and  $G(x) = x^2$ . This choice for  $f_{\hat{x}}(x)$  respects positivity and normalization. To generate values of  $x^{(k)}$  according to  $f_{\hat{x}}(x)$  we need to invert  $F_{\hat{x}}(x)$ , which in this case can be done analytically:

$$F_{\hat{x}}(x) = \int_0^x dy e^{-y} = 1 - e^{-x} = u \rightarrow x = F_{\hat{x}}^{-1}(u) = -\log(1 - u) \quad (1.15)$$

Since  $1 - u$  is also uniformly distributed in the interval  $(0, 1)$  we can write simply:

$$x = -\log(u) \quad (1.16)$$

which is equivalent to  $x = -\log(1 - u)$  from the statistical point of view. A program to compute integral (1.14) can be:

```

m=1000
r=0.0
s=0.0
do 99 k=1,m
x=ran_f()
gk=g(x)
r=r+gk
s=s+gk*gk
99 continue

```

```

r=r/m
s=s/m-r*r
s=sqrt(s/m)
write(6,*) r,'+/-',s
end
function g(x)
g=x*x
return
end
function ran_f()
ran_f=-log(ran_u())
return
end

```

A technical point is that, in some cases, the function  $F_{\tilde{x}}^{-1}(u)$  is not expressible in terms of elementary functions and some kind of approximative methods (such as numerical interpolation) might be needed to compute it. It is important to realize that the decomposition of a given integral in  $f_{\tilde{x}}(x)$  and  $G(x)$  to put it in the form of Eq.(1.3) might not be unique. In the previous example, Eq.(1.14), we could have taken as well  $f_{\tilde{x}}(x) = xe^{-x}$  and  $G(x) = x$ . This choice, however, makes the probability distribution function  $F_{\tilde{x}}(x)$  difficult to invert.

An important case in which the function  $F_{\tilde{x}}^{-1}(u)$  is not easily calculable is that of Gaussian random numbers (of mean zero and variance 1) for which the probability density function is:

$$f_{\tilde{x}}(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad (1.17)$$

(random numbers  $y$  of mean  $\mu$  and variance  $\sigma^2$  are easily obtained by the linear transformation  $y = \sigma x + \mu$ ). The inverse probability distribution function is

$$x = \sqrt{2} \operatorname{erf}^{-1}(2u - 1) \quad (1.18)$$

where  $\operatorname{erf}^{-1}(z)$  is the inverse error function (Abramowitz et al. 1972; Valls et al. 1986). The inverse error function does not usually belong to the set of predefined functions in a programming language, although some libraries (for example, the NAG library) do include it in their list of functions. An alternative to the generation of Gaussian distributed random numbers is the algorithm of Box-Muller-Wiener (Box et al. 1958; Ahrens et al. 1972; 1988) which is based on a change of variables to polar coordinates. Namely,

$$\begin{aligned} x_1 &= \sqrt{\rho} \cos(\theta) \\ x_2 &= \sqrt{\rho} \sin(\theta) \end{aligned} \quad (1.19)$$

If  $x_1$  and  $x_2$  are independent Gaussian variables, it is easy to prove that  $\rho$  and  $\theta$  are independent variables.  $\rho$  follows an exponential distribution and  $\theta$  a uniform

distribution, so that their inverse distribution functions can be expressed in terms of elementary functions. Namely,

$$f_\rho(\rho) = \frac{1}{2}e^{-\rho/2} \rightarrow F_\rho(\rho) = 1 - e^{-\rho/2}, \quad \rho \geq 0$$

$$f_\theta(\theta) = \frac{1}{2\pi} \rightarrow F_\theta(\theta) = \frac{\theta}{2\pi}, \quad 0 \leq \theta \leq 2\pi \quad (1.20)$$

The resulting algorithm is:

$$x_1 = \sqrt{-2 \log(u)} \cos(2\pi v)$$

$$x_2 = \sqrt{-2 \log(u)} \sin(2\pi v) \quad (1.21)$$

Where  $u$  and  $v$  are independent random variables uniformly distributed in the interval  $(0, 1)$ . The main advantage of this Box-Muller-Wiener algorithm is that it is exact, yielding two independent Gaussian variables,  $x_1, x_2$  from two independent uniform variables,  $u, v$ . Its main disadvantage, though, is that it is extremely slow since it involves the calculation of trigonometric, logarithm and square root functions. In most of the applications a linear interpolation approximation to the inverse error function does produce sufficiently good quality Gaussian random numbers at a considerable gain in speed (Toral et al. 1993). A possible implementation of the Box-Muller-Wiener algorithm is the following:

```
function ran_g()
data is /-1/
is=-is
if (is.eq.1) then
a=sqrt(-2.0*log(ran_u()))
b=6.2831853072*ran_u()
ran_g=a*cos(b)
x2=a*sin(b)
return
endif
ran_g=x2
return
end
```

(a practical note: it might be necessary to tell the compiler that the two functions `ran_u()` that appear in the previous program produce different values and need to be computed separately, sometimes compilers are too clever).

Another important case that deserves being considered explicitly, in despite of its simplicity, concerns the generation of events with a given probability. Imagine we want to simulate tossing a biased coin, with a  $p = 0.6$  probability for heads (and 0.4 for tails). We need to generate a variable that takes some value (1, for example) 60% of the times and another value (0), 40% of the times. This can be done by comparing a random number  $u$  uniformly distributed in the interval  $(0, 1)$  with the given probability  $p$ . If  $u \leq p$  then we take  $x = 1$ , otherwise we take  $x = 0$ . This is achieved by the program:

```

if (ran_u().lt.p) then
    x=1
else
    x=0
endif

```

For  $N$ -dimensional variables  $(x_1, \dots, x_N)$ , the situation is much more complex. The equivalent of Theorem 1 states that in order to generate values of the  $N$ -dimensional variable  $(x_1, \dots, x_N)$  we need (Rubinstein 1981): (i) generate  $N$ -independent random numbers  $(u_1, \dots, u_N)$  uniformly distributed in the interval  $(0, 1)$  (this is the easy part) and (ii) solve the following set of equations:

$$\begin{aligned}
 F_{x_1}(x_1) &= u_1 \\
 F_{x_2}(x_2|x_1) &= u_2 \\
 &\dots\dots \\
 F_{x_N}(x_N|x_1, \dots, x_{N-1}) &= u_N
 \end{aligned}
 \tag{1.22}$$

Where, for example,  $F_{x_2}(x_2|x_1)$  is the conditional probability distribution function of the variable  $x_2$  given that  $x_1$  has taken a particular value, and so on. The calculation of the conditional probability distribution functions is generally at least as complicated as the calculation of the original integral we wanted to compute numerically and the above procedure is of little practical use. In order to develop alternative methods suitable for the generation of  $N$ -dimensional variables we need first to introduce the so-called rejection methods for 1 variable.

#### 1.4. Rejection method for N=1

In those cases that the inverse probability distribution function is difficult to compute, the rejection method offers a very convenient alternative. Also, it is the basis of the  $N$ -dimensional methods which we will develop later. The method is based on the fact that the probability density function  $f_{\hat{x}}(x)$  is proportional to the probability that the variable  $x$  takes a particular value. If, for example,  $f_{\hat{x}}(x_1) = 2f_{\hat{x}}(x_2)$ , we can affirm that the value  $x_1$  is twice as probable as the value  $x_2$ . The rejection method (in its simplest version) proposes the values  $x_1$  and  $x_2$  with the same probability and then accepts the proposed value  $x$  with a probability  $h(x)$  proportional to  $f_{\hat{x}}(x)$ , such that, in our example,  $x_1$  will be accepted twice as many times as  $x_2$ . Consequently,  $x_1$  will appear twice as many times as  $x_2$ , which is equivalent to saying that  $x_1$  is twice as probable as  $x_2$  as desired. We will illustrate the method by an example. Let us consider the probability density function:

$$f_{\hat{x}}(x) = 6x(1-x), \quad x \in (0, 1) \tag{1.23}$$

Which has the shape indicated in Fig.1.

We propose a value of  $x$  uniformly in the interval  $(0, 1)$ . The proposed value has to be accepted with a probability  $h(x)$  proportional to  $f_{\hat{x}}(x)$ ,  $h(x) = \alpha f_{\hat{x}}(x)$ .



**Fig. 1.** Probability density function  $f_{\hat{x}}(x) = 6x(1 - x)$  (solid line) and acceptance probability  $h(x) = 4x(1 - x)$  (dashed line)

The constant  $\alpha$  is arbitrary but the resulting function  $h(x)$  has to be interpreted as a probability, which means that we have to keep the bounds  $0 \leq h(x) \leq 1$ . This yields  $\alpha \leq \frac{1}{\max_x f_{\hat{x}}(x)}$ . Obviously, the rejection method is more efficient when the acceptance probability is high, which means that one has to take the maximum possible value for  $\alpha$ . In our example, the maximum for  $f_{\hat{x}}(x)$  is at  $x = 0.5$ ,  $f_{\hat{x}}(0.5) = 1.5$ , so that we take  $\alpha = 2/3$  and, consequently,  $h(x) = 4x(1 - x)$ , see Fig. 1. The acceptance process is done as explained in the previous section by comparing the probability  $h(x)$  with a random number uniformly distributed in  $(0, 1)$ . The whole process is repeated until the proposed value is accepted. The final algorithm can be coded as:

```

function ran_f()
h(x)=4.0*x*(1.0-x)
1  x=ran_u()
   if (ran_u().gt.h(x)) goto 1
   ran_f=x
   return
end

```

We will now consider a more general version of the rejection method in which a value is proposed according to some probability distribution function  $g_{\hat{x}}(x)$  and then accepted according to some probability  $h(x)$ . The probability distribution function of the joint process "propose the value  $x$  and accept it" is  $g_{\hat{x}}(x)h(x)$ . According to the Bayes theorem, the probability density function of the variable  $x$  given that it has been accepted is (Grimmett et al. 1982):

$$f_{\hat{x}}(x) = \frac{g_{\hat{x}}(x)h(x)}{\int_{-\infty}^{\infty} dz g_{\hat{x}}(z)h(z)} \quad (1.24)$$

One interesting thing to notice is that the previous expression is properly normalized. This is very important in those cases where the normalization constant for  $f_{\hat{x}}(x)$  is not known explicitly. Looking at Eq.(1.24) one can see that there is some freedom in choosing the functions  $g_{\hat{x}}(x)$  and  $h(x)$ . The only requirement is that its product  $g_{\hat{x}}(x)h(x)$  be proportional to the function  $f_{\hat{x}}(x)$  we want to generate. An important constraint is that the proposal  $g_{\hat{x}}(x)$  must be normalized and that the acceptance function  $h(x)$  must be bound in the interval  $(0, 1)$ . For instance, in the example of Eq.(1.23), we could have considered  $g_{\hat{x}}(x) = 2x$ . The function  $h(x)$  should then be proportional to  $3(1-x)$ . Since  $0 \leq h(x) \leq 1$  one takes simply  $h(x) = 1-x$ . The generation of  $g_{\hat{x}}(x) = 2x$  is straightforward since the inverse probability distribution function is  $G_{\hat{x}}^{-1}(u) = \sqrt{u}$ . The proposed value is accepted with probability  $1-x$ , i.e. rejected with probability  $x$ . This can be coded as:

```

function ran_f()
1  x=sqrt(ran_u())
   if(ran_u().lt.x) goto 1
   ran_f=x
   return
end

```

An efficient method is one that requires, on average, a small average number of proposals before acceptance. This average number is given by the inverse of the overall acceptance probability,  $p_a$  which is nothing but the denominator of expression (1.24):

$$p_a \equiv p(\text{accept}) = \int_{-\infty}^{\infty} dz g_{\hat{x}}(z) h(z) \quad (1.25)$$

It is easy to show that of the two methods developed for the function (1.23) the first algorithm is more efficient than the second.

Let us consider yet another example: to generate numbers distributed according to

$$f_{\hat{x}}(x) = C \exp \left[ -\frac{x^2}{2} - x^4 \right] \quad (1.26)$$

the obvious choice is (notice that the precise value of  $C$  is not needed):

$$\begin{aligned} g_{\hat{x}}(x) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \\ h(x) &= e^{-x^4} \end{aligned} \quad (1.27)$$

Here  $g_{\hat{x}}(x)$  is nothing but a Gaussian distribution, and  $h(x)$  is the optimal choice, given the choice for  $g_{\hat{x}}(x)$ , since  $\max_x h(x) = 1$ . The overall acceptance probability is, according to Eq.(1.25):

$$p_a = \int_{-\infty}^{\infty} dz \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} e^{-z^4} \approx 0.62028 \quad (1.28)$$

And the average number of trials needed to accept a value is  $1/0.62028 \approx 1.6$ .

### 1.5. Rejection with repetition

There is a modification of the rejection method consisting in the following: if the proposed value is not accepted then, instead of proposing a new one, the previous value is repeated. This method obviously produces correlated values (some of them will be equal to the previous ones) but, however, can be very efficient for vector computers in which the usual structure of the rejection method prevents vectorization. We still have to prove that the values generated with this method do follow the required distribution  $f_{\hat{\mathbf{x}}}(x)$ . According to Bayes theorem, the probability density function  $f_{\hat{\mathbf{x}}_n}(x)$  generated at the  $n$ -proposal step will have two different contributions corresponding to the acceptance or rejection, respectively, of the proposed value:

$$f_{\hat{\mathbf{x}}_n}(x) = f_{\hat{\mathbf{x}}_n}(x|accept)p(accept) + f_{\hat{\mathbf{x}}_n}(x|reject)p(reject) \quad (1.29)$$

The first term of the right hand side can be written as the probability of acceptance given  $x$  (which is the function  $h(x)$ ) times the probability density function of proposing  $x$  (which is  $g_{\hat{\mathbf{x}}}(x)$ ):

$$f_{\hat{\mathbf{x}}_n}(x) = h(x)g_{\hat{\mathbf{x}}}(x) + f_{\hat{\mathbf{x}}_n}(x|reject)(1 - p_a) \quad (1.30)$$

If  $x$  has been rejected, the previous value is repeated which means that the probability density function at the  $n$ -step in the case of rejection is the same that the one at the  $(n - 1)$ -step:

$$f_{\hat{\mathbf{x}}_n}(x) = h(x)g_{\hat{\mathbf{x}}}(x) + f_{\hat{\mathbf{x}}_{n-1}}(x) \left[ 1 - \int_{-\infty}^{\infty} dz h(z)g_{\hat{\mathbf{x}}}(z) \right] \quad (1.31)$$

where we have substituted  $p_a$  as given by Eq.(1.25). The solution of this linear recurrence relation is:

$$f_{\hat{\mathbf{x}}_n}(x) = (1 - p_a)^n \left[ f_{\hat{\mathbf{x}}_0}(x) - \frac{h(x)g_{\hat{\mathbf{x}}}(x)}{\int_{-\infty}^{\infty} dz h(z)g_{\hat{\mathbf{x}}}(z)} \right] + \frac{h(x)g_{\hat{\mathbf{x}}}(x)}{\int_{-\infty}^{\infty} dz h(z)g_{\hat{\mathbf{x}}}(z)} \quad (1.32)$$

We can write this result in terms of the desired distribution  $f_{\hat{\mathbf{x}}}(x)$ , Eq. (1.24):

$$f_{\hat{\mathbf{x}}_n}(x) = (1 - p_a)^n [f_{\hat{\mathbf{x}}_0}(x) - f_{\hat{\mathbf{x}}}(x)] + f_{\hat{\mathbf{x}}}(x) \quad (1.33)$$

Given that  $0 < p_a \leq 1$ , one concludes that the solution tends to the desired distribution in the limit  $n \rightarrow \infty$  independently of the initial distribution  $f_{\hat{\mathbf{x}}_0}$ :

$$\lim_{n \rightarrow \infty} f_{\hat{\mathbf{x}}_n}(x) = f_{\hat{\mathbf{x}}}(x) \quad (1.34)$$

If the initial proposed numbers  $x_0$  are distributed according to  $f_{\hat{\mathbf{x}}}(x)$ , i.e. if  $f_{\hat{\mathbf{x}}_0}(x) = f_{\hat{\mathbf{x}}}(x)$ , then we would have  $f_{\hat{\mathbf{x}}_n}(x) = f_{\hat{\mathbf{x}}}(x)$ ,  $\forall n$  and the numbers we obtain are distributed according to the desired distribution from the beginning. However, if the initial numbers are not distributed according to  $f_{\hat{\mathbf{x}}}(x)$  (as it is usually the case) we can still affirm that, thanks to the factor  $(1 - p)^n$ , the initial condition will be lost after a number sufficiently large of proposals. This process of rejecting the, say,  $M_0$  initial values produced because they are not distributed

yet according to the required distribution is called *thermalization*. The initial time needed for thermalization can be computed from the non-linear correlation function (Binder et al, 1988):

$$\rho_x^{NL}(k) = \frac{\langle x^{(k)} \rangle - \langle x \rangle}{\langle x^{(0)} \rangle - \langle x \rangle} \quad (1.35)$$

The time  $M_0$  can be measured as a characteristic time for the decay of the above function towards the stationary value 0. For the method of rejection with repetition the non-linear correlation function coincides with the auto-correlation function (see later):

$$\rho_x^{NL}(k) = (1 - p_a)^k \quad (1.36)$$

The thermalization time  $M_0$  can be defined as the time it takes the above function to reach a small value  $\epsilon$ .

$$M_0 = \frac{\ln \epsilon}{\ln(1 - p_a)} \quad (1.37)$$

If, for instance,  $\epsilon = 10^{-8}$ ,  $p_a = 0.5$ , we have  $M_0 \approx 27$ .

We can illustrate now the enhancement of the errors due to the correlations. In this method of rejection with repetition, one can compute the correlation function of the variable  $x$  as (see Eq.(1.10)):

$$\rho_x(k) = \lim_{k' \rightarrow \infty} \frac{\langle x^{(k)} x^{(k+k')} \rangle - \langle x \rangle^2}{\langle x^2 \rangle - \langle x \rangle^2} = (1 - p_a)^k \quad (1.38)$$

From Eq.(1.9) we deduce that the autocorrelation time is:

$$\tau_x = \sum_{k=1}^{\infty} \rho_x(k) = \frac{1 - p}{p} \quad (1.39)$$

Remember that, according to Eq.(1.8) the statistical error gets enhanced by the factor  $(2\tau_x + 1)^{1/2}$ .

We finally mention that it is possible to interpret this algorithm of rejection with repetition as a Markov succession (Grimmett et al. 1982). Indeed, Eq.(1.31) can be cast in the form:

$$f_{\hat{\mathbf{x}}_n}(x) = \int_{-\infty}^{\infty} f(x|y) f_{\hat{\mathbf{x}}_{n-1}}(y) dy \quad (1.40)$$

where the transition probability is:

$$f(x|y) = h(x) g_{\hat{\mathbf{x}}}(x) + \left[ 1 - \int_{-\infty}^{\infty} dz h(z) g_{\hat{\mathbf{x}}}(z) \right] \delta(x - y) \quad (1.41)$$

### 1.6. The algorithm of Metropolis et al.

Although it is very powerful, the rejection method can fail if the distribution is very peaked around an (unknown) value  $x_0$  (see Fig.2). If our proposal does not take into account that only a very small part of the variable space (that around  $x_0$ ) is important, we are bound to have a extremely high rejection probability yielding the rejection method completely useless. The solution comes by making a proposal that does not differ much of the present value of the random variable we are sampling, i.e. by having the transition probabilities of the Markov succession depend explicitly on the present value, say  $y$ , of the variable. This implies that the functions  $h(x)$  and  $g_{\tilde{x}}(x)$  depend on  $y$  and become two-variable functions,  $g_{\tilde{x}}(x|y)$ ,  $h(x|y)$ . The transition probability Eq.(1.41) then becomes also a two variable function:

$$f(x|y) = h(x|y)g_{\tilde{x}}(x|y) + \left[ 1 - \int_{-\infty}^{\infty} h(z|y)g_{\tilde{x}}(z|y) dz \right] \delta(x - y) \quad (1.42)$$

**Fig. 2.** Function  $f_{\tilde{x}}(x)$  for which the rejection method can fail

The equation giving the evolution of the probability density function at the  $n$ -step is still Eq. (1.40), but now the recursion equation for  $f_{\tilde{x}_n}(x)$  does not tell us in a straight way which is the stationary ( $n \rightarrow \infty$ ) distribution. We would like to have  $f_{\tilde{x}}(x)$  as stationary solution of the recursion equation, i.e:

$$f_{\tilde{x}}(x) = \int_{-\infty}^{\infty} f(x|y)f_{\tilde{x}}(y) dy \quad (1.43)$$

By using the normalization condition  $\int_{-\infty}^{\infty} f(y|x) dy = 1$  and the definition (1.42) for the transition probability  $f(x|y)$ , one gets easily:

$$\int dy [g_{\tilde{x}}(x|y)h(x|y)f_{\tilde{x}}(y) - g_{\tilde{x}}(y|x)h(y|x)f_{\tilde{x}}(x)] = 0 \quad (1.44)$$

A sufficient condition to fulfill this relation is given by the *detailed balance condition*:

$$g_{\hat{x}}(x|y)h(x|y)f_{\hat{x}}(y) = g_{\hat{x}}(y|x)h(y|x)f_{\hat{x}}(x) \quad (1.45)$$

to be satisfied by the proposal  $g_{\hat{x}}(x|y)$  and the acceptance probability  $h(x|y)$ . Let us remark that if the detailed balance condition is satisfied and if  $f_{\hat{x}_0}(x) = f_{\hat{x}}(x)$  then  $f_{\hat{x}_n}(x) = f_{\hat{x}}(x)$ ,  $\forall n$ . Also, if the transition probabilities satisfy the condition of *ergodicity*, which, roughly speaking, means that any value for the random variable can be achieved starting from any other value after a sufficient number of steps, then we can assure that  $f_{\hat{x}}(x)$  is the only stationary solution of the Markov succession and that  $\lim_{n \rightarrow \infty} f_{\hat{x}_n}(x) = f_{\hat{x}}(x)$ , independently of the value of  $f_{\hat{x}_0}(x)$  (see Kalos et al, 1986).

It is important to stress that any functions  $g_{\hat{x}}(x|y)$  and  $h(x|y)$  that satisfy ergodicity and the detailed balance condition are suitable to use in this rejection method. The most widely used solution is the one given by the **the Metropolis algorithm** (Metropolis et al. 1953) in which  $g_{\hat{x}}(x|y)$  is a given function usually of the form:

$$g_{\hat{x}}(x|y) = \frac{1}{2\Delta}, \quad \text{if } |x - y| \leq \Delta \quad (1.46)$$

i.e.  $x$  is sample uniformly from the interval  $(y - \Delta, y + \Delta)$ . Once the value of  $x$  has been proposed one looks for acceptance probabilities  $h(x|y)$  verifying the detailed balance condition:

$$\frac{h(x|y)}{h(y|x)} = \frac{g_{\hat{x}}(y|x)f_{\hat{x}}(x)}{g_{\hat{x}}(x|y)f_{\hat{x}}(y)} \equiv q(x|y) \quad (1.47)$$

When searching for solutions of these equations one has to remember that  $h(x|y)$  is a probability and must also satisfy the condition  $0 \leq h(x|y) \leq 1$ . A possible solution is the *Metropolis solution*:

$$h(x|y) = \min(1, q(x|y)) \quad (1.48)$$

Another widely used solution is the *Glauber solution* (Glauber 1963):

$$h(x|y) = \frac{q(x|y)}{1 + q(x|y)} \quad (1.49)$$

A family of solutions is obtained by:

$$h(x|y) = \sqrt{q(x|y)} \omega(q(x|y)) \quad (1.50)$$

where  $\omega(z)$  is any function satisfying

$$\omega(z) = \omega(z^{-1}) \quad (1.51)$$

In particular, the Metropolis solution is recovered taking:

$$\omega(z) = \min(z, z^{-1}) \quad (1.52)$$

and the Glauber solution taking

$$\omega(z) = \frac{1}{z^{1/2} + z^{-1/2}} \quad (1.53)$$

Before we use the Metropolis algorithm to study some models of field theory, we want to finish this section by showing the use of the Metropolis algorithm with a simple example in a one variable case. We will consider the Gaussian distribution:

$$f_{\mathbf{x}}(x) = Ae^{-x^2/2} \quad (1.54)$$

the normalization constant  $A$  is irrelevant for the Metropolis method. The proposal will be a value belonging to the interval  $(y - \Delta, y + \Delta)$ . Notice that this choice (Eq.(1.46)) satisfies the symmetry relation  $g_{\mathbf{x}}(x|y) = g_{\mathbf{x}}(y|x)$ . This proposal will be accepted with a probability given, for example, by the Metropolis solution Eq.(1.48):

$$h(x|y) = \min(1, q(x|y)) = \min(1, e^{(y^2-x^2)/2}) \quad (1.55)$$

i.e if  $|x| \leq |y|$  accept with probability 1, otherwise accept with probability  $e^{(y^2-x^2)/2}$ . The Metropolis algorithm can be coded as:

```
function ran_g(y,delta)
x=y+delta*(2*ran_u()-1)
if(abs(x).gt.abs(y)) then
  if (exp(0.5*(y-x)*(y+x)).lt.ran_u()) then
    ran_g=y
    return
  endif
endif
y=x
ran_g=y
return
end
```

Intuitively, the effect of this algorithm is to accept with a larger probability those proposals which tend towards the origin, where the probability has a maximum.

An important warning concerns the fact that we must keep the old value of the variable if the new one has been rejected. We can not keep proposing until a value is accepted (at variance with what happened in the ordinary rejection method in which the two procedures were correct). If one does not keep the old value, a reasoning similar to the one used in the rejection method, section 1.4, leads to the following transition probabilities, see Eq.(1.24):

$$f(x|y) = \frac{h(x|y)g(x|y)}{\int_{-\infty}^{\infty} dz h(z|y)g(z|y)} \quad (1.56)$$

and:

$$f(y|x) = \frac{h(y|x)g(y|x)}{\int_{-\infty}^{\infty} dz h(z|x)g(z|x)} \quad (1.57)$$

Now it is very difficult to satisfy the detailed balance conditions, Eq.(1.45), namely:

$$\frac{h(x|y)g(x|y)}{\int_{-\infty}^{\infty} dz h(z|y)g(z|y)} f_{\hat{x}}(y) = \frac{h(y|x)g(y|x)}{\int_{-\infty}^{\infty} dz h(z|x)g(z|x)} f_{\hat{x}}(x) \quad (1.58)$$

since the integrals are generally difficult to compute (in particular, the Metropolis solution does not satisfy this condition). The reader can test the following modification of the Metropolis algorithm for the Gaussian distribution:

```

function ran_g(y,delta)
1  x=y+delta*(2*ran_u()-1)
  if(abs(x).gt.abs(y)) then
    if (exp(0.5*(y-x)*(y+x)).lt.ran_u()) goto 1
  endif
  y=x
  ran_g=y
  return
end

```

and check that it does not produce Gaussian random numbers (it is worse for large  $\Delta$ ).

For efficiency of the Metropolis algorithm  $\Delta$  must be chosen such that the autocorrelation time of the algorithm,  $\tau$ , takes its minimum value. It is easy to understand why there must be a minimum of  $\tau$  as a function of  $\Delta$ . If  $\Delta$  is very small, the new proposal will be very close to the old one (hence highly correlated),  $\tau$  will be large and  $h(x|y)$  will be very close to 1 such that the acceptance will be also close to 1. On the other hand, if  $\Delta$  is large, the acceptance probability will be small and the proposed value will be often rejected, such that the new value will be equal to the old one (again highly correlated). A rule of thumb tells us that  $\Delta$  must be chosen such that the acceptance probability is neither too high, neither too small, i.e. of the order of 50%.

### 1.7. Rejection method in the N-dimensional case

We really do not need to develop any new concepts. The nice thing about the Metropolis algorithm is that is easily generalizable to the  $N$ -dimensional case. Simply replace  $x$  and  $y$  in the above expressions by the  $N$ -dimensional variable vectors  $(x_1, \dots, x_N)$  and  $(y_1, \dots, y_N)$ , respectively, and all of the above formulae and concepts will still apply. In order to keep the acceptance probability within reasonable limits, the proposal can not be too different from the old value. This is achieved by changing only one of the  $N$  variables at one time. The variable to be updated can be chosen randomly amongst the set of the  $N$  variables or sequentially. If chosen randomly the function  $g_{\hat{x}}(x|y)$  is explicitly given by:

$$g_{\hat{x}}(x|y) = \frac{1}{N} \sum_{i=1}^N \frac{1}{2\Delta} \prod_{j \neq i} \delta(x_j - y_j) \quad (1.59)$$



Since it verifies the symmetry condition  $g_{\hat{\mathbf{x}}}(x|y) = g_{\hat{\mathbf{x}}}(y|x)$ , the function  $q(x|y)$  is simply  $q(x|y) = f_{\hat{\mathbf{x}}}(x)/f_{\hat{\mathbf{x}}}(y)$ . The acceptance probability can be chosen as the Metropolis solution  $h(x|y) = \min(1, q(x|y))$  or other solution.

### 1.8. Heat Bath

In the Metropolis algorithm the acceptance probability  $h(x|y)$  was determined (with some freedom) once the proposal  $g_{\hat{\mathbf{x}}}(x|y)$  had been specified. In the so called **heat bath** algorithm, which is useful for  $N$ -dimensional variables, the proposed value for the variable is always accepted, i.e. one chooses

$$h(x|y) = 1 \quad (1.60)$$

The detailed balance condition becomes

$$g_{\hat{\mathbf{x}}}(x|y)f_{\hat{\mathbf{x}}}(y) = g_{\hat{\mathbf{x}}}(y|x)f_{\hat{\mathbf{x}}}(x) \quad (1.61)$$

In the 1-variable case, a trivial solution (remember that  $g_{\hat{\mathbf{x}}}(x|y)$  must be a probability density function) is:  $g_{\hat{\mathbf{x}}}(x|y) = f_{\hat{\mathbf{x}}}(x)$  which is independent of the old value  $y$ . In the  $N$ -variable case, a solution is found by changing only one of the  $N$  variables at a time:

$$g_{\hat{\mathbf{x}}}(x|y) = \frac{1}{N} \sum_{i=1}^N g_{\hat{\mathbf{x}}}(x_i|y) \prod_{j \neq i} \delta(x_j - y_j) \quad (1.62)$$

where each of the functions  $g_{\hat{\mathbf{x}}}(x_i|y)$  satisfies detailed balance:

$$\begin{aligned} g_{\hat{\mathbf{x}}}(x_i|y)f_{\hat{\mathbf{x}}}(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_N) = \\ g_{\hat{\mathbf{x}}}(y_i|x)f_{\hat{\mathbf{x}}}(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_N) \end{aligned} \quad (1.63)$$

A solution of this functional equation is obtained by taking  $g_{\hat{\mathbf{x}}}(x_i|y) = g_{\hat{\mathbf{x}}}(x_i)$ , independent of  $y$ , as the following conditional probability:

$$g_{\hat{\mathbf{x}}}(x_i) = f_{\hat{\mathbf{x}}}(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N) \quad (1.64)$$

It is trivial to verify detailed balance, Eq.(1.63), if one remembers the definition of the conditional probability:

$$f_{\hat{\mathbf{x}}}(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N) = \frac{f_{\hat{\mathbf{x}}}(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_N)}{f_{\hat{\mathbf{x}}}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N)} \quad (1.65)$$

Intuitively, what the heat bath method does when updating variable  $x_i$ , is to select the value of  $x_i$  according to the conditional probability given that the rest of the variables is fixed. These fixed variables act as a *heat bath* for variable  $x_i$ . In many cases, it turns out that the conditional probability  $g_{\hat{\mathbf{x}}}(x_i) = f_{\hat{\mathbf{x}}}(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N)$  takes a simple form and can be sampled by any of the 1-variable methods explained in this chapter.

## 2. $\Phi^4$ Model

### 2.1. Introduction and basic definitions

We will illustrate the use of Monte Carlo techniques in field theory with the scalar  $\Phi^4$  model. This model has been used in many different contexts, in such different topics as quantum field theory (Zinn Justin 1989) or in the study of structural phase transitions (Cowley 1980; Bruce 1980; Bruce et al. 1980).

Let us consider a  $d$ -dimensional (hyper-cubic) regular lattice  $\Lambda$ , consisting of  $N = L^d$  points. Every point  $i = 1, \dots, N$  of this lattice has  $d$  coordinates:  $i = (j_1, \dots, j_d)$ . Periodic boundary conditions are assumed on this lattice. On every site of the lattice there is a scalar variable  $\Phi_i$ . The set of all variables is  $[\Phi] \equiv (\Phi_1, \dots, \Phi_N)$ . We also introduce a Hamiltonian function  $\mathcal{H}$  given by:

$$\beta\mathcal{H}([\Phi]) = \sum_{i=1}^N a_0^d \left[ \frac{-b}{2}\Phi_i^2 + \frac{u}{4}\Phi_i^4 + \frac{K}{2} \sum_{\mu=1}^d \left( \frac{\Phi_{i_\mu} - \Phi_i}{a_0} \right)^2 \right] \quad (2.1)$$

The sum over  $\mu$  runs over  $d$ -nearest neighbours of site  $i$ , i.e. if the coordinates of  $i$  are  $(j_1, \dots, j_d)$  then the coordinates of  $i_\mu$  are  $(j_1, \dots, j_\mu + 1, \dots, j_d)$ . The continuum limit of the system can be obtained by letting the lattice spacing  $a_0$  tend to 0 and the system size  $L$  to  $\infty$ . In this limit, the sums are replaced by integrals and the sum over  $\mu$  tends to a gradient. The continuum Hamiltonian is then:

$$\beta\mathcal{H}([\Phi]) = \int d\mathbf{r} \left[ \frac{-b}{2}\Phi(\mathbf{r})^2 + \frac{u}{4}\Phi(\mathbf{r})^4 + \frac{K}{2}|\nabla\Phi(\mathbf{r})|^2 \right] \quad (2.2)$$

This expression for the Hamiltonian is the preferred one for analytical treatment, such as series expansions or renormalization-group treatments in momentum space (Amit 1984). The lattice version is the preferred one for numerical studies, besides the fact that it yields a regularization of the Hamiltonian (Parisi 1988). We will consider from now on only the lattice version of the  $\Phi^4$  model. We can set the lattice spacing  $a_0 = 1$ , since it can be rescaled in the parameters  $b$ ,  $u$  and  $K$ .

The first two terms of the sum appearing in the Hamiltonian (2.1) are local terms (depending only on the field at location  $i$ ) and can be thought of as local potential terms  $V(\Phi_i)$ :

$$V(\Phi_i) = \frac{-b}{2}\Phi_i^2 + \frac{u}{4}\Phi_i^4 \quad (2.3)$$

For stability of this local potential, the parameter  $u$  must be greater than 0. The local potential changes qualitative when the parameter  $b$  changes sign (see Fig.3). If  $b < 0$  the local potential has only one minimum at  $\Phi_i = 0$ . On the other hand, when  $b > 0$  there are two minima of equal depth located at  $\Phi_i = \pm\sqrt{b/u}$ .

The third term in (2.1), the one multiplied by  $K$ , is called the interaction term. When  $K > 0$ , the contribution of this term is always non negative and the ground state of the Hamiltonian  $\mathcal{H}$  is reached when the contribution of this term is zero, i.e. when all the fields take the same values  $\Phi_i = \Phi_0, \forall i$ . This is

the situation considered by the simple **mean field** theory (Landau et al. 1980). Since all the fields take the same value, the mean field Hamiltonian reduces to:

$$N^{-1}\beta\mathcal{H}^{MF}(\Phi_0) = \frac{-b}{2}\Phi_0^2 + \frac{u}{4}\Phi_0^4 \quad (2.4)$$

**Fig. 3.** Local Potential  $V(\Phi)$ , Eq.(2.3), in the cases (i)  $b > 0$  and (ii)  $b < 0$

The value of the mean field value  $\Phi_0$  depends on the constants  $b, u$  and is given by:

$$\Phi_0 = \begin{cases} 0 & \text{if } b < 0 \\ \sqrt{\frac{b}{u}} & \text{if } b > 0 \end{cases} \quad (2.5)$$

(see Fig.4). To go beyond mean field theory one needs to consider fluctuations: the fields  $\Phi_i$  do not take a constant value, but fluctuate from one lattice site to another. In this case, we need a statistical description which, according to the principles of Statistical Mechanics (Pathria 1972), is given by the canonical distribution. The probability density function  $f([\Phi])$  governing the statistical properties of the fields  $[\Phi]$  at inverse temperature  $\beta$  is:

$$f([\Phi]) = \frac{e^{-\beta\mathcal{H}([\Phi])}}{\mathcal{Z}(b, u, K)} \quad (2.6)$$

The denominator of this expression is called the **partition function** and it is the multiple integral of the numerator for all the field values:

$$\mathcal{Z}(b, u, K) = \int_{-\infty}^{\infty} d\Phi_1 \dots \int_{-\infty}^{\infty} d\Phi_N e^{-\beta\mathcal{H}([\Phi])} \equiv \int d\Phi e^{-\beta\mathcal{H}([\Phi])} \quad (2.7)$$

The magnitudes of interest are computed as averages of field functions,  $G([\Phi])$ , with the probability density function  $f([\Phi])$ :

$$\langle G([\Phi]) \rangle = \mathcal{Z}^{-1} \int d\Phi G([\Phi]) e^{-\beta\mathcal{H}([\Phi])} \quad (2.8)$$

Example of quantities of interest are the *magnetization*,  $m$  defined as:

$$m = \left\langle \frac{1}{N} \sum_{i=1}^N \Phi_i \right\rangle \quad (2.9)$$

the energy,  $e$ :

$$e = \left\langle \frac{\mathcal{H}[\Phi]}{N} \right\rangle \quad (2.10)$$

the magnetic susceptibility,  $\kappa_T$ :

$$\kappa_T = N \left[ \left\langle \left( \frac{1}{N} \sum_{i=1}^N \Phi_i \right)^2 \right\rangle - \langle m \rangle^2 \right] \quad (2.11)$$

the specific heat,  $C_V$ :

$$C_V = N \left[ \left\langle \left( \frac{\mathcal{H}[\Phi]}{N} \right)^2 \right\rangle - \langle e \rangle^2 \right] \quad (2.12)$$

and many others.

In general, these quantities differ considerably from the mean field values. In Figure 4 we can see the difference between the magnetization computed from a numerical calculation in two dimensions and the mean field result (which is independent of dimension).

It is known that the  $\Phi^4$  model belongs to the universality class of the Ising model (Amit 1984). This means that both models share the same critical exponents and scaling functions. Here, however, we want to point out a different relation between the two models, namely, that the Ising model can be obtained as a suitable limit of the  $\Phi^4$  model. The limit is obtained as follows: expanding the square in the "gradient" term and using the periodic boundary conditions, one can rewrite the Hamiltonian in the form:

$$\beta\mathcal{H}([\Phi]) = \sum_{i=1}^N \left[ \left( dK - \frac{b}{2} \right) \Phi_i^2 + \frac{u}{4} \Phi_i^4 \right] - K \sum_{\langle i,j \rangle} \Phi_i \Phi_j \quad (2.13)$$

where the sum over  $\langle i,j \rangle$  means sum over all the possible pairs of  $i$ - $j$  nearest neighbours in the lattice. Introducing a rescaled field variable

$$S_i = \frac{\Phi_i}{\sqrt{\frac{b-2dK}{u}}} \quad (2.14)$$

one can write:

$$\beta\mathcal{H}([S]) = \alpha \sum_{i=1}^N \left[ -\frac{1}{2} S_i^2 + \frac{1}{4} S_i^4 \right] - \beta J \sum_{\langle i,j \rangle} S_i S_j \quad (2.15)$$

**Fig. 4.** Mean field magnetization, Eq.(2.5) (solid line) and the result of a numerical calculation using Monte Carlo techniques in the two-dimensional system (dashed line). In this plot we use the value  $\chi = 1$  (see Eq.(2.21) for a definition of the parameters).

with

$$\alpha = \frac{(-b + 2dK)^2}{u}, \quad \beta J = \frac{K(b - 2dK)}{u} \quad (2.16)$$

The Gibbs factor is

$$e^{-\beta\mathcal{H}([S])} = \prod_{i=1}^N \left\{ \exp \left[ -\alpha \left( -\frac{1}{2}S_i^2 + \frac{1}{4}S_i^4 \right) \right] \exp \left[ \beta J \sum_{\langle i,j \rangle} S_i S_j \right] \right\} \quad (2.17)$$

The local factor of this distribution is always double peaked centered around the values  $\pm 1$  (see Fig.5). Now we can think of the parameter  $\alpha$  as controlling the width of the local field distribution around these two maxima. A large value for  $\alpha$  implies that the field is bound to take essentially the values  $\pm 1$ , and  $S_i$  becomes what is called a *spin variable*. In the limit  $\alpha \rightarrow \infty$  with  $J$  kept constant we recover the Ising model. This limit is obtained in the original  $\Phi^4$  model by taking

$$\begin{aligned} b &\rightarrow \infty \\ u &\rightarrow \infty \\ \frac{b}{u} &\rightarrow \text{constant} \end{aligned} \quad \beta J = K \frac{b}{u} \quad (2.18)$$

In this limit, one obtains for the Gibbs factor:

$$e^{-\beta\mathcal{H}([S])} = \prod_{i=1}^N [\delta(S_i^2 - 1)] \exp \left[ \beta J \sum_{\langle i,j \rangle} S_i S_j \right] \quad (2.19)$$

This is equivalent to writing:

$$\mathcal{H}([S]) = -J \sum_{\langle i,j \rangle} S_i S_j \quad S_i = \pm 1 \quad (2.20)$$

Which is nothing but the Hamiltonian of the Ising model. For smaller values of  $\alpha$ , the field can fluctuate around the two maxima, hence the name of *soft spins model* that sometimes is given to the  $\Phi^4$  model in this situation (Cowley 1980; Bruce 1980; Bruce et al. 1980).

**Fig. 5.** Function  $f(x) = \exp[-\alpha(-\frac{1}{2}S_i^2 + \frac{1}{4}S_i^4)]$  appearing in Eq.(2.17), in the cases  $\alpha = 1$  (solid line),  $\alpha = 10$  (dashed line),  $\alpha = 100$  (dotted-dashed line). When  $\alpha \rightarrow \infty$ , the function  $f_{\mathbf{x}}(x)$  tends to the sum of two delta functions located at  $x = \pm 1$ .

## 2.2. Monte Carlo methods

Before we proceed further and show how one can obtain numerically the quantities of interest in the  $\Phi^4$  model, we notice that one of the three parameters of the model,  $u, b, K$ , is redundant since it can be absorbed in the field scale. Many ways of redefining parameters have been used in the literature (see (Torral et al. 1990) for a review). We will use here a simple reparametrization in which the field is rescaled by a factor  $K^{1/2}$  hence yielding a parameter independent interaction term (this rescaling is obviously valid only in the ferromagnetic case  $K > 0$ ). Specifically, we introduce a new field  $\phi$ , and two new parameters  $\theta$  and  $\chi$  by the definitions:

$$\begin{aligned} \phi &= K^{1/2} \Phi \\ \theta &= \frac{b}{K} \\ \chi &= \frac{u}{K^2} \end{aligned} \quad (2.21)$$

The Hamiltonian, in terms of these new variables, can be written as:

$$\begin{aligned}\beta\mathcal{H}([\phi]) &= \sum_{i=1}^N \left[ \frac{-\theta}{2} \phi_i^2 + \frac{\chi}{4} \phi_i^4 + \frac{1}{2} \sum_{\mu=1}^d (\phi_{i_\mu} - \phi_i)^2 \right] \\ &= \sum_{i=1}^N \left[ \frac{\tilde{\theta}}{2} \phi_i^2 + \frac{\chi}{4} \phi_i^4 - \sum_{\mu=1}^{2d} \phi_i \phi_{i_\mu} \right]\end{aligned}\quad (2.22)$$

(the sum over  $\mu$  runs over the  $2d$  nearest neighbours of site  $i$ ). Here we have introduced

$$\tilde{\theta} = 2d - \theta \quad (2.23)$$

This Hamiltonian can be explicitly separated into the local part and the interaction part,  $\beta\mathcal{H} = \beta\mathcal{H}_0 + \beta\mathcal{H}_I$  with

$$\beta\mathcal{H}_0 = \sum_{i=1}^N \left[ \frac{\tilde{\theta}}{2} \phi_i^2 + \frac{\chi}{4} \phi_i^4 \right] \quad (2.24)$$

$$\beta\mathcal{H}_I = - \sum_{\langle i,j \rangle} \phi_i \phi_j \quad (2.25)$$

Now it is about time we start applying what we learnt in the previous chapter about Monte Carlo techniques! To implement the Metropolis algorithm we need to take the following steps:

- (i) Select (randomly or sequentially) one of the  $N$  field variables,  $\phi_i$ .
- (ii) Propose a change to another close value,  $\phi'_i$ , chosen randomly in the interval  $\phi_i - \Delta, \phi_i + \Delta$ , with  $\Delta$  a suitable value selected such that the acceptance probability is around 0.5.
- (iii) Compute the change in energy,  $\Delta\mathcal{H}$  that this proposed change produces.
- (iv) Accept the proposed value  $\phi'_i$  with probability  $\min(1, e^{-\beta\Delta\mathcal{H}})$ .

In computing the change of energy one does not have to use the full expression (2.22), but rather notice that most of the terms disappear when subtracting the old and the new energy, such that the change in energy is simply:

$$\beta\Delta\mathcal{H} = \frac{\tilde{\theta}}{2}(\phi'^2_i - \phi_i^2) + \frac{\chi}{4}(\phi'^4_i - \phi_i^4) + (\phi'_i - \phi_i) \sum_{\mu=1}^{2d} \phi_{i_\mu} \quad (2.26)$$

This simple Metropolis algorithm has some convergence problems due to the fact that most of the trials belong to a small probability region. This is particularly true in the vicinities of the Ising limit in which the values around 0 have a very small probability (see again Fig.5). Several methods have been proposed as an alternative to the Metropolis algorithm and all of these methods fit into the proposal/acceptance scheme developed in the previous chapter: use a proposal probability density function  $g_{\hat{\mathbf{x}}}(x|y)$  (where  $x \equiv (\phi'_1, \dots, \phi'_N)$  and  $y \equiv (\phi_1, \dots, \phi_N)$  stand for the complete field configuration after and before the proposal, respectively) and an acceptance probability  $h(x|y)$ , satisfying detailed balance:

$$g_{\hat{\mathbf{x}}}(x|y)h(x|y) \frac{e^{-\beta\mathcal{H}(y)}}{\mathcal{Z}} = g_{\hat{\mathbf{x}}}(y|x)h(y|x) \frac{e^{-\beta\mathcal{H}(x)}}{\mathcal{Z}} \quad (2.27)$$

As usual, only one variable will be updated at a time, what makes the proposal  $g_{\tilde{\mathbf{x}}}(x|y)$  become a function  $g_{\tilde{\mathbf{x}}}(\phi'_i|y)$ . In the approach of reference (Milchev et al. 1986) the proposal  $g_{\tilde{\mathbf{x}}}(\phi'_i|y)$  is chosen independent of  $y$  and proportional to the Gibbs factor of the local term of the Hamiltonian:

$$g_{\tilde{\mathbf{x}}}(\phi'_i) = C \exp\left(-\frac{\tilde{\theta}}{2}\phi'^2_i - \frac{\chi}{4}\phi'^4_i\right) \quad (2.28)$$

this is a one-variable probability density function that can be sampled by any of the methods explained in chapter 1. The authors in reference (Milchev et al. 1986) chose a numerical inversion method. Once this proposal has been taken, in the detailed balance condition only the interaction part of the Hamiltonian appears explicitly,

$$h(x|y)e^{-\beta\mathcal{H}_I[y]} = h(y|x)e^{-\beta\mathcal{H}_I[x]} \quad (2.29)$$

For this equation we can use the Metropolis solution:

$$h(x|y) = \min(1, e^{-\beta\Delta\mathcal{H}_I}) \quad (2.30)$$

where the novel feature is that only the change of the interaction energy appears in the exponential. This adopts a very simple form:

$$\beta\Delta\mathcal{H}_I = (\phi'_i - \phi_i) \sum_{\mu=1}^{2d} \phi_{i\mu} \quad (2.31)$$

In general, this procedure can be used when the Hamiltonian can be splitted in a sum of local terms plus an interaction term. One can chose the new value of the variable, independently of the old value, according to the distribution dictated by the local term. This proposal is then accepted with the Metropolis probability using only the interaction term.

The heat-bath algorithm has also been used for this model (Torral et al. 1990). Let us remember than in the heat-bath algorithm, the acceptance probability is equal to 1 (i.e. the proposal is always accepted), one variable  $\phi'_i$  is changed at a time, and the proposal probability density function  $g_{\tilde{\mathbf{x}}}(\phi'_i)$  is obtained from a distribution in which all the other variables remain constant, i.e. we need to identify exactly where does the variable  $\phi'_i$  appears, all the other terms will be considered constants. From Eq.(2.22) it is very easy to find out the expression for  $g_{\tilde{\mathbf{x}}}(\phi'_i)$ :

$$g_{\tilde{\mathbf{x}}}(\phi'_i) = A \exp\left(-\frac{\tilde{\theta}}{2}\phi'^2_i - \frac{\chi}{4}\phi'^4_i + \phi'_i \sum_{\mu=1}^{2d} \phi_{i\mu}\right) \quad (2.32)$$

$A$  is some normalization factor depending on the values of the other fields (which are considered to be constant in the heat-bath method). To sample this one-variable probability density function it will not be useful to use a numerical inversion method, because this function depends of the sum of the  $2d$  neighbours of site  $i$ , which vary from trial to trial. To sample this distribution we can use instead a rejection technique. If we introduce  $c = \sum_{\mu=1}^{2d} \phi_{i\mu}$ , we can write the above function as:



$$g_{\mathbf{x}}(\phi'_i) = B \left[ \exp\left(-\frac{\chi}{4}\phi'^4_i\right) \right] \left[ \frac{1}{\tilde{\theta}^{-1/2}\sqrt{2\pi}} \exp\left(-\frac{(\phi'_i - c/\tilde{\theta})^2}{2\tilde{\theta}^{-1}}\right) \right] \quad (2.33)$$

$B$  is another normalization constant. This is the product of the function  $\exp(-\frac{\chi}{4}\phi'^4_i)$  and a Gaussian distribution of mean  $c/\tilde{\theta}$  and variance  $\tilde{\theta}^{-1}$  (for this we require  $\tilde{\theta} > 0$ ). This Gaussian distribution can be generated by the relation  $\phi'_i = c/\tilde{\theta} + \tilde{\theta}^{-1/2}\text{ran\_g}()$  where  $\text{ran\_g}()$  is a Gaussian distributed random variable of mean 0 and variance 1. Finally, the value  $\phi'_i$  is accepted with a probability  $\exp(-\frac{\chi}{4}\phi'^4_i)$ .

Another elegant way of implementing a Monte Carlo method for the  $\Phi^4$  model is that of Bruce (Bruce 1985) in his study of the *border model* (this is nothing but the  $\Phi^4$  model in the case  $\tilde{\theta} = 0$ ). In his approach, the proposal  $g_{\mathbf{x}}(x|y) \equiv g_{\mathbf{x}}(\phi'_i)$  is also independent of the old configuration  $y$  (in the same vein than heat-bath), but  $g_{\mathbf{x}}(\phi'_i)$  is chosen to be the sum of two Gaussians which best approximate the actual local distribution of the field  $\phi'_i$ :

$$g_{\mathbf{x}}(\phi'_i) = \frac{1}{2} \left[ \frac{1}{\sigma_1\sqrt{2\pi}} \exp\left(-\frac{(\phi'_i - \mu_1)^2}{2\sigma_1^2}\right) + \frac{1}{\sigma_2\sqrt{2\pi}} \exp\left(-\frac{(\phi'_i - \mu_2)^2}{2\sigma_2^2}\right) \right] \quad (2.34)$$

To sample this distribution one generates a Gaussian number,  $\text{ran\_g}()$  of mean 0 and variance 1 and a uniform number,  $\xi$ , in  $(0, 1)$ . If  $\xi < 0.5$  then one chooses the field from the first Gaussian distribution, i.e.  $\phi'_i = \mu_1 + \sigma_1\text{ran\_g}()$ , else, if  $\xi \geq 0.5$ , one chooses from the second,  $\phi'_i = \mu_2 + \sigma_2\text{ran\_g}()$ . An initial guess of the parameters  $\mu_1$ ,  $\mu_2$ ,  $\sigma_1$  and  $\sigma_2$  is later refined with the information coming from the computed local distribution. The acceptance probability must satisfy detailed balance:

$$g_{\mathbf{x}}(\phi'_i)h(x|y) \frac{e^{-\beta\mathcal{H}(y)}}{\mathcal{Z}} = g_{\mathbf{x}}(\phi_i)h(y|x) \frac{e^{-\beta\mathcal{H}(x)}}{\mathcal{Z}} \quad (2.35)$$

introducing the *effective* interaction Hamiltonian,  $\tilde{\mathcal{H}}([\phi])$ :

$$\beta\tilde{\mathcal{H}}([\phi]) = \beta\mathcal{H}([\phi]) + \sum_{i=1}^N \ln g_{\mathbf{x}}(\phi_i) \quad (2.36)$$

the detailed balance condition becomes:

$$h(x|y)e^{-\beta\tilde{\mathcal{H}}(y)} = h(y|x)e^{-\beta\tilde{\mathcal{H}}(x)} \quad (2.37)$$

from which it is clear that a possible solution for the acceptance probability is the Metropolis solution in terms of the function  $\tilde{\mathcal{H}}$ :

$$h(x|y) = \min\left(1, e^{-\beta\Delta\tilde{\mathcal{H}}}\right) \quad (2.38)$$

### 2.3. Histogram extrapolation

We have now developed some tools to study numerically the  $\Phi^4$  model. We shall next apply them to the computation of some magnitudes of interest. Let us focus first on the magnetization  $m$  (see Fig.4). This is a function of the parameters  $\theta$  and  $\chi$ . If we are interested, say, in the variation of  $m$  with  $\theta$  for a fixed value of  $\chi$ , we have to run our numerical algorithm for several values of  $\theta$ . For each run, we have to make sure that the system has reached equilibrium (thermalization) and then produce a sufficient number of independent configurations such that the errors are kept under reasonable limits. Repeating this procedure for a large number of  $\theta$  values in order to have smooth curves for the magnetization is a painful and slow procedure. There is a way of improving somehow this process by using information from simulations at a given value of the parameter  $\theta$  to obtain results for another value  $\theta'$ . The idea, which is extremely simple (and clever!), has been obtained and reobtained many times, but were Ferrenberg and Swendsen (see Ferrenberg et al. 1989 and references therein) who dramatically demonstrated its utility in the case of the Ising model. We now describe this *histrogram extrapolation technique* for the  $\Phi^4$  model.

The configurations obtained in a numerical simulation using the Monte Carlo method follow the Gibbs probability distribution which depend on the parameters  $\theta$  and  $\chi$ , that we now write out explicitly:

$$f([\Phi]; \theta, \chi) = \frac{e^{-\beta\mathcal{H}([\Phi]; \theta, \chi)}}{\mathcal{Z}(\theta, \chi)} \quad (2.39)$$

The type of averages one is interested on usually involve only the following functions:

$$\begin{aligned} m_0 &= \sum_{i=1}^N \phi_i \\ m_1 &= \sum_{i=1}^N \phi_i^2 \\ m_2 &= \sum_{i=1}^N \phi_i^4 \\ m_3 &= \sum_{i=1}^N \sum_{\mu=1}^d (\phi_{i_\mu} - \phi_i)^2 \end{aligned} \quad (2.40)$$

other functions  $m_i$  can be incorporated into what follows, but we restrict to these for simplicity. Since the Hamiltonian depends only on linear combinations of  $m_1$ ,  $m_2$  and  $m_3$ , the probability distribution function in terms of  $m_0$ ,  $m_1$ ,  $m_2$  and  $m_3$  is:

$$f(m_0, m_1, m_2, m_3; \theta, \chi) = \frac{N(m_0, m_1, m_2, m_3) \exp\left(\frac{\theta}{2}m_1 - \frac{\chi}{4}m_2 - \frac{1}{2}m_3\right)}{\mathcal{Z}(\theta, \chi)} \quad (2.41)$$

Here  $N(m_0, m_1, m_2, m_3)dm_0dm_1dm_2dm_3$  is the number of configurations with values of  $m_i$  in the interval  $(m_i, m_i + dm_i)$ ,  $i = 0, 1, 2, 3$ . These values of  $N(m_0, m_1, m_2, m_3) \equiv N([m_i])$  are recorded during the simulation forming a histogram (hence the name of histogram extrapolation). The partition function can be expressed in terms of  $N([m_i])$  as:

$$\mathcal{Z}(\theta, \chi) = \int \prod_i dm_i N([m_i]) \exp\left(\frac{\theta}{2}m_1 - \frac{\chi}{4}m_2 - \frac{1}{2}m_3\right) \quad (2.42)$$

The average value of any function  $G([m_i])$  can be computed as:

$$\langle G([m_i]) \rangle = \int \prod_i dm_i f([m_i]; \theta, \chi) G([m_i]) \quad (2.43)$$

The extrapolation scheme allows one to compute the probability density function  $f([m_i]; \theta', \chi')$ , for values  $\theta', \chi'$  of the model parameters, if  $f([m_i]; \theta, \chi)$  is known. For our particular model, the method is based on the exact relation:

$$f([m_i]; \theta', \chi') = \frac{f([m_i]; \theta, \chi) \exp\left(\frac{\theta' - \theta}{2}m_1 - \frac{\chi' - \chi}{4}m_2\right)}{\int \prod_i dm_i f([m_i]; \theta, \chi) \exp\left(\frac{\theta' - \theta}{2}m_1 - \frac{\chi' - \chi}{4}m_2\right)} \quad (2.44)$$

Let us, in order to discuss the implication of this relation, consider the simpler case  $\chi' = \chi$  for which the above formula reduces to:

$$f([m_i]; \theta', \chi) = \frac{f([m_i]; \theta, \chi) \exp\left(\frac{\theta' - \theta}{2}m_1\right)}{\int \prod_i dm_i f([m_i]; \theta, \chi) \exp\left(\frac{\theta' - \theta}{2}m_1\right)} \quad (2.45)$$

This identity is telling us that if we know the probability distribution function of the  $m_i$ 's at a given value of the parameter  $\theta$  then we know it **for any other value**  $\theta'$ . This is wonderful! Only one simulation at a selected value of  $\theta$  is required! Once we have computed the probability density function  $f([m_i]; \theta, \chi)$  we can obtain, by means of the previous formula, the probability density function  $f([m_i]; \theta', \chi)$  and from the knowledge of  $f([m_i]; \theta, \chi)$  we can compute any average we need using Eq.(2.43).

In practice, alas!, things are not so easy. Relation (2.45) is indeed exact, but what it is telling us is that the information contained in the tail of the distribution of  $f([m_i]; \theta, \chi)$  is used to compute the distribution  $f([m_i]; \theta', \chi)$ , see Fig. 6. The only problem is that the errors in the tail of the distribution usually are very large, so making infeasible to extrapolate to a value of  $\theta'$  very far from  $\theta$ .

Of course, the width of the distribution  $f([m_i]; \theta, \chi)$  will be very important to determine the extrapolation range  $\Delta\theta$ . If the distribution is very wide, one can extrapolate to farther away values than for a very narrow distribution. This is the reason why this histogram extrapolation technique is useful near the critical point where the distributions are wider.

**Fig. 6.** Probability distribution functions  $f([m_i]; \theta, \chi)$  at two different values of  $\theta$  using Eq.(2.43). According to Eq.(2.45) the region around the maximum of the function given by the dashed line is obtained from the tail of the function given by the solid line, where statistical errors are larger.

## 2.4. Finite size scaling

In Figures 7 to 10 we have plotted, for different system sizes and using the histogram extrapolation scheme, results for several quantities of interest: magnetization, energy, specific heat and susceptibility. Each of these figures has been produced by just two simulations whose location is marked with points in the figures. The rest of the curves have been obtained using the extrapolation scheme detailed above.

These figures show what is called *finite size effects* (Barber 1983). Let us compare Fig. 7 for the magnetization with Fig.4 which depicts the mean field behavior and the expected true behavior. The theory predicts that there should be a value of  $\theta = \theta_c$  below which the spontaneous magnetization is strictly zero. In fact, the magnetization behaviour near  $\theta_c$  can be characterized by a critical exponent,  $\beta$  such that:

$$m(\theta) = \begin{cases} 0 & \text{if } \theta < \theta_c \\ a\epsilon^\beta & \text{if } \theta > \theta_c \end{cases} \quad (2.46)$$

Here  $\epsilon = 1 - \frac{\theta}{\theta_c}$ . This is a non-analytical behaviour which, strictly speaking, can only appear in a system of infinite size. For a finite system, the number of integrations appearing in the definition of the partition function and the calculation of averages is always finite, thus leading necessarily to an analytical result. We can understand intuitively why finite size effects will be more important near a second order phase transition. In this situation the correlation length which measures the linear range over which fields at different sites of the lattice are correlated, diverges (in an infinite system) with a power-law singularity:

**Fig. 7.** Magnetization  $m$  for the two-dimensional  $\phi^4$  model for  $\chi = 1$  and different values of the system side  $L = 8, 16, 24, 32, 40$  (from bottom to top lines). For each value of  $L$  a continuous line has been drawn by running two simulations at the points marked with a dot and extrapolating to other values of  $\theta$  using Eq.(2.45).

**Fig. 8.** Same than figure 7 for the energy  $\epsilon$  and  $\chi = 0.1$ .

$$\xi(\theta) \sim |\epsilon|^{-\nu} \quad (2.47)$$

For a finite system, the fields can not be correlated longer than the system side and we must have  $\xi \sim L$ . The theory of finite size tells us exactly how (and why!) the averages of interest behave (Cardy 1988). The basic idea is that now the magnetization, say, becomes a *homogeneous* function of  $\xi$  and the system side  $L$ ,  $m(\xi, L) = \xi^x \tilde{m}(\xi/L)$ . The unknown exponent  $x$  is obtained by demanding that in the infinite system, and close enough to the critical point, one recovers the known behaviour given by Eq.(2.46). This implies that the function  $\tilde{m}(z)$

takes a finite limit when  $z \rightarrow 0$  and then:

$$m(\theta) = \lim_{L \rightarrow \infty} m(\theta, L) = \xi^x \tilde{m}(0) \sim [\epsilon^{-\nu}]^x \sim \epsilon^{-x\nu} \quad (2.48)$$

compared to (2.46) one concludes  $\beta = -x\nu$  and then the prediction for the magnetization near the critical point for a finite system is:

$$m(\theta, L) = \xi^{-\beta/\nu} \tilde{m}(\xi/L) = L^{-\beta/\nu} \tilde{m}[(1 - \theta/\theta_c)L^{-1/\nu}] \quad (2.49)$$

The typical way of checking this scaling behavior is to plot  $m(\theta, L)L^{\beta/\nu}$  vs the rescaled variable  $(1 - \theta/\theta_c)L^{-1/\nu}$ . If, as it is usually the case, the critical value  $\theta_c$  and the critical exponents  $\nu, \beta$  are not known, this procedure implies a three-parameter fit which is very difficult to do in practice. One can use the equivalent scaling relations for the specific heat and the susceptibility:

$$\begin{aligned} C(\theta, L) &= L^{\alpha/\nu} \tilde{C}[(1 - \theta/\theta_c)L^{-1/\nu}] \\ \kappa(\theta, L) &= L^{\gamma/\nu} \tilde{\kappa}[(1 - \theta/\theta_c)L^{-1/\nu}] \end{aligned} \quad (2.50)$$

**Fig. 9.** Specific heat for the two-dimensional  $\phi^4$  model for  $\chi = 1$ . Same symbols and meanings than in figure 7.

From Figures 9 and 10 we see that both  $C(\theta, L)$  and  $\kappa(\theta, L)$  develop maxima at locations  $\theta_c^1(L)$  and  $\theta_c^2(L)$ , respectively. By using the above expressions for the specific heat and the susceptibility one can see that these values behave as:

$$\begin{aligned} \theta_c^1(L) &= \theta_c + a_1 L^{-1/\nu} \\ \theta_c^2(L) &= \theta_c + a_2 L^{-1/\nu} \end{aligned} \quad (2.51)$$

From where we deduce that a plot of  $\theta_c^1(L)$  and  $\theta_c^2(L)$  vs  $L^{-1/\nu}$  must yield a straight line whose interception at the origin is precisely  $\theta_c$ . This can be seen in Fig.11 for which the critical value  $\theta_c = 1.265 \pm 0.010$  for  $\chi = 1$  is deduced.

**Fig. 10.** Magnetic susceptibility for the two-dimensional  $\phi^4$  model for  $\chi = 0.1$ . Same symbols and meanings than in figure 7.

In this figure we have used the known value  $\nu = 1$  for the  $2 - d$   $\phi^4$  model. When the value of  $\nu$  is not known one could use directly the three parameter fit  $\theta_c^i(L) = \theta_c + a_i L^{-1/\nu}$  to obtain  $\theta_c$  although the quality of the fit usually is not good enough to allow also for a very accurate measurement of  $\nu$ . Once the value of  $\theta_c$  is known one can use the remaining finite size scaling relations to obtain the critical exponents.

### 3. Field Dynamics

#### 3.1. Introduction

Up to now we have focused only on the equilibrium properties of the scalar  $\Phi^4$  model. If one wants to write dynamical equations for the fields one finds the difficulty that the  $\Phi^4$  Hamiltonian, so successful for equilibrium properties, does not contain enough information (e.g. kinetic energy terms) to determine the dynamics of the model. Instead of adding more terms to the Hamiltonian, the usual approach to study dynamical properties is to consider that the field variables follow some phenomenological equations of motion. These equations of motion will be stochastic in nature to reflect our ignorance of the detailed effect of the microscopic variables in the field variables (Gunton et al. 1983b). In addition, they must respect the basic symmetries of the system and lead, in the limit of infinite time, to the equilibrium canonical distribution given by the Gibbs factor.

**Fig. 11.** Pseudo critical parameters  $\theta_c^1(L)$  (open dots) and  $\theta_c^2(L)$  (black dots) defined as the location of the maxima of the specific heat and susceptibility, respectively, plotted vs.  $L^{-1}$ . The extrapolation to the origin, according to Eq.(2.51) yields the critical value  $\theta_c = 1.265 \pm 0.010$ .

### 3.2. The Langevin equation

As discussed before, one writes down semi-phenomenological stochastic equations (Langevin equations) to model the time evolution of the field. For the simplest case of the  $\Phi^4$  model without any further symmetry (remember than the main symmetry of the model is the  $\Phi \rightarrow -\Phi$ ), one is guided by the fact that at sufficiently low temperatures the final configuration must be a minimum of the Hamiltonian. Then, the time evolution is defined by a purely relaxational model such that the speed of approaching the ground state is proportional to the gradient of the Hamiltonian. One adds a stochastic contribution (usually called the noise term) that ensures the correct thermal fluctuations in the stationary (equilibrium) state. The resulting Langevin equation is:

$$\frac{\partial \Phi(\mathbf{r}, t)}{\partial t} = -\Gamma \frac{\delta \mathcal{H}}{\delta \Phi(\mathbf{r}, t)} + \eta(\mathbf{r}, t) \quad (3.1)$$

Here  $\Gamma$  is a positive coefficient usually assumed to be constant, although in some cases one also studies the effect of the dependance of  $\Gamma$  on the field  $\Phi$  (Langer et al. 1975; Lacasta et al. 1992, 1993). As in the equilibrium case, one assumes a lattice discretization in which, instead of a real variable in every point of the space, only points on a  $d$ -dimensional lattice  $\Lambda$  are considered. In the lattice case, the functional derivative  $\delta \mathcal{H} / \delta \Phi(\mathbf{r}, t)$  becomes a partial derivative:

$$\frac{d\Phi_i(t)}{dt} = -\Gamma a_0^{-d} \frac{\partial \mathcal{H}}{\partial \Phi_i(t)} + \eta_i(t) \quad (3.2)$$

In the absence of the noise term, it is easy to show that the system evolves towards a minimum of the Hamiltonian. We have:



$$\frac{d\mathcal{H}}{dt} = \sum_{i=1}^N \frac{\partial \mathcal{H}}{\partial \Phi_i} \frac{d\Phi_i}{dt} = -\Gamma a_0^{-d} \sum_{i=1}^N \left( \frac{\partial \mathcal{H}}{\partial \Phi_i} \right)^2 \leq 0 \quad (3.3)$$

The statistical properties of the noise terms,  $\eta_i(t)$ , have to be determined in order to obtain the correct equilibrium distribution. They are supposed to be Gaussian distributed of mean zero and correlations given by:

$$\langle \eta_i(t) \eta_j(t') \rangle = D a_0^{-d} \delta_{i,j} \delta(t - t') \quad (3.4)$$

characteristic of the so called *white noise* (Gardiner 1985).  $D$  is a parameter, to be determined, representing the intensity of the noise and, consequently,  $D$  must increase with temperature. In the limit of the lattice spacing  $a_0$  going to 0, the Kronecker-delta becomes a Dirac-delta function:

$$\langle \eta(\mathbf{r}, t) \eta(\mathbf{r}', t') \rangle = D \delta(\mathbf{r} - \mathbf{r}') \delta(t - t') \quad (3.5)$$

From now on, we will take the lattice spacing as unit of length  $a_0 = 1$ . Since  $\eta_i$  is Gaussian, these correlation functions completely determine its statistical properties. We now relate the noise intensity  $D$  to temperature  $T$ . For later reference, we will be more general and consider the set of stochastic differential equations:

$$\frac{d\Phi_i(t)}{dt} = F_i(t) + \sum_{j=1}^N G_{ij} \eta_j(t) \quad (3.6)$$

With the same statistical properties for the noise term than before, Eq.(3.4).  $G_{ij}$  is a constant matrix. The complete statistical description of the fields  $[\Phi]$  is given by the joint probability density function  $P([\Phi]; t)$ , which satisfies the Fokker-Planck equation (Gardiner 1985). The description given by the Langevin equations is equivalent to the one given by the Fokker-Planck equation:

$$\frac{\partial P([\phi]; t)}{\partial t} = \sum_{i=1}^N \frac{\partial}{\partial \Phi_i} \left[ -F_i P + \frac{D}{2} \sum_{j=1}^N H_{ij} \frac{\partial P}{\partial \Phi_j} \right] \quad (3.7)$$

The matrix  $H$  is  $H = GG^\top$ . In the case we are concerned with, Eq.(3.2),  $H = \mathbf{I}$ , unity matrix, and  $F_i$  is related to the partial derivative of the Hamiltonian:

$$\frac{\partial P([\phi]; t)}{\partial t} = \sum_{i=1}^N \frac{\partial}{\partial \Phi_i} \left[ \Gamma \frac{\partial \mathcal{H}}{\partial \Phi_i} P + \frac{D}{2} \frac{\partial P}{\partial \Phi_i} \right] \quad (3.8)$$

It is easy to verify that the stationary solution of this equation is:

$$P_{st}([\Phi]) \propto \exp \left[ -\frac{2\Gamma}{D} \mathcal{H}([\Phi]) \right] \quad (3.9)$$

Since we want the statistical stationary properties of  $[\Phi]$  to be determined by the Gibbs distribution,  $\exp[-\beta \mathcal{H}([\Phi])]$  we are led to identify:

$$\frac{2\Gamma}{D} = \beta \implies D = 2k_B T \Gamma \quad (3.10)$$

$D$  is then proportional to temperature, as we had anticipated. By using the expression for the lattice Hamiltonian equation (2.1) we arrive at the following equation:

$$\frac{d\Phi_i(t)}{dt} = \Gamma k_B T (b\Phi_i - u\Phi_i^3 + K\nabla_L^2\Phi_i) + \eta_i(t) \quad (3.11)$$

$\nabla_L^2$  is the lattice laplacian operator defined by:

$$\nabla_L^2\Phi_i = \sum_{\mu=1}^{2d} (\Phi_{i_\mu} - \Phi_i) \quad (3.12)$$

here the sum over  $\mu$  runs over the  $2d$  nearest neighbours of site  $i$ . By redefining time  $t \rightarrow t\Gamma k_B T$  and by introducing the same field rescaling and parameters than in the equilibrium case, Eq. (2.21), we arrive at the somewhat simpler equation:

$$\frac{d\phi_i(t)}{dt} = \theta\phi_i - \chi\phi_i^3 + \nabla_L^2\phi_i + \xi_i(t) \quad (3.13)$$

where the new noise variables  $\xi_i$  satisfy:

$$\langle \eta_i(t)\eta_j(t') \rangle = 2\delta_{i,j}\delta(t-t') \quad (3.14)$$

The Langevin equation we have considered so far is the simplest dynamical model than can be written for a scalar field and in the famous taxonomy of Hohenberg and Halperin (Hohenberg et al. 1977) is simply called *model A*. Langevin-type equations can describe more complex systems, such as *model B* which is obtained formally by replacing the constant coefficient  $\Gamma$  in Eq.(3.1) defining model A by  $-F\nabla^2$ . The resulting equation of motion is:

$$\frac{\partial\Phi(\mathbf{r}, t)}{\partial t} = \nabla^2 \left( \Gamma \frac{\delta\mathcal{H}([\Phi])}{\delta\Phi(\mathbf{r}, t)} \right) + \eta(\mathbf{r}, t) \quad (3.15)$$

where  $\eta(\mathbf{r}, t)$  are Gaussian distributed random variables of mean zero and correlations:

$$\langle \eta(\mathbf{r}, t)\eta(\mathbf{r}', t') \rangle = -D\delta(t-t')\nabla^2\delta(\mathbf{r}-\mathbf{r}') \quad (3.16)$$

We let the physical interpretation of this model to the next chapter. Here we will mention that model B is suitable to describe the relaxational evolution of a *conserved* field, i.e. one for which the spatial integral of the field is time independent:

$$m \equiv \int d\mathbf{r}\Phi(\mathbf{r}, t) \longrightarrow \frac{dm}{dt} = 0 \quad (3.17)$$

The lattice discretization of model B equation leads to:

$$\frac{d\Phi_i(t)}{dt} = \nabla_L^2 \left( \Gamma \frac{\partial\mathcal{H}([\Phi])}{\partial\Phi(\mathbf{r}, t)} \right) + \eta_i(t) \quad (3.18)$$

Again, in order to ensure that the correct equilibrium distribution is obtained, one needs to tune the correct value for  $D$ . The Fokker-Planck equation is now:

$$\begin{aligned} \frac{\partial P([\phi]; t)}{\partial t} &= \sum_{i=1}^N \frac{\partial}{\partial \Phi_i} \left[ -\Gamma \nabla_L^2 \left( \frac{\partial \mathcal{H}}{\partial \Phi_i} \right) P + \frac{D}{2} \sum_{j=1}^N \frac{\partial}{\partial \Phi_j} (-\nabla_L^2 \delta_{i,j} P) \right] \\ &= - \sum_{i=1}^N \frac{\partial}{\partial \Phi_i} \nabla_L^2 \left[ \Gamma \frac{\partial \mathcal{H}}{\partial \Phi_i} P + \frac{D}{2} \frac{\partial P}{\partial \Phi_i} \right] \end{aligned} \quad (3.19)$$

The equilibrium distribution is given by  $\exp[-\frac{2\Gamma}{D}\mathcal{H}([\Phi])]$  and we are led to identify  $D = 2\Gamma k_B T$  as in model A.

### 3.3. Numerical Solution of Stochastic Differential Equations

The Langevin equations for models A and B that we considered in the previous section can not be solved exactly. A great deal of what we know about the behavior of the solution of these equations comes from numerical simulations. In this section we want to study how to handle numerically Langevin equations of the general form:

$$\frac{dx_i(t)}{dt} = F_i(t) + \sum_{j=1}^N G_{ij} \eta_j(t) \quad i = 1, \dots, N \quad (3.20)$$

$F_i$  are functions that depend on the set of variables  $x_1, \dots, x_N$ . If  $G_{i,j}$  also depend on  $x_1, \dots, x_N$ , one talks of *multiplicative* noise and the resulting stochastic differential equations will be considered in the Stratonovich sense (van Kampen, 1981). If, on the other hand,  $G_{i,j}$  are constant functions, one talks of *additive* noise. The noise variables  $\eta_i(t)$  are Gaussian distributed, independent random variables of mean zero and correlations given by:

$$\langle \eta_i(t) \eta_j(t') \rangle = D \delta_{i,j} \delta(t - t') \quad (3.21)$$

Of course, the appearance of the Dirac-delta functions tells us that we might have some problems of mathematical rigor in we insist in considering the noise variables  $\eta_i$  as simple functions. They should be considered as distributions, but most of the results we can get with more mathematical rigor are also obtained with the usual function formalism . . . and some care. In particular, since the noise variables are not differentiable, we can not use the standard Taylor expansions that help so much in developing algorithms for numerically solving ordinary (not stochastic) differential equations (Press et al. 1986). Instead, we must use integration expansions (Kloeden et al. 1992). We can express these ideas more clearly with a simple example.

Let us consider a one variable stochastic differential equation:

$$\frac{dx(t)}{dt} = F(t) + \eta(t) \quad (3.22)$$

where  $F(t)$  is a sufficiently smooth (differentiable) function and  $\eta(t)$  is a Gaussian distributed variable of mean zero and correlations:

$$\langle \eta(t)\eta(t') \rangle = D\delta(t - t') \quad (3.23)$$

In fact, this equation is so simple that we can solve it exactly. However, let us develop a numerical algorithm to solve it. The algorithm generates a recursion relation that will allow us to compute  $x(t+h)$  given  $x(t)$ ,  $h$  is the integration step. This is the same structure than in ordinary differential equations in which numerical methods do not give the solution for every value of time  $t$  but only at regular intervals  $t_n = t_0 + nh$ ,  $n = 0, 1, \dots$  separated by a integration step  $h$ . In order to find the recursion relation we integrate (3.22) between  $t_n$  and  $t_n + h$  to obtain:

$$\int_{t_n}^{t_n+h} dt \frac{dx(t)}{dt} = x(t_n+h) - x(t_n) = \int_{t_n}^{t_n+h} ds F(s) + \int_{t_n}^{t_n+h} ds \eta(s) \quad (3.24)$$

the first term of the right hand side of this equation can be approximated by

$$\int_{t_n}^{t_n+h} ds F(s) = hF(t_n) + o(h^2) \quad (3.25)$$

The second term is a Gaussian variable since it is the integral of a Gaussian variable. Hence, its statistical properties are completely determined by the mean value and the correlations. Let us define the random variable:

$$\omega_h(t) = \int_t^{t+h} ds \eta(s) \quad (3.26)$$

whose mean value is:

$$\langle \omega_h(t) \rangle = \int_t^{t+h} ds \langle \eta(s) \rangle = 0 \quad (3.27)$$

and whose correlations are given by:

$$\langle \omega_h(t)\omega_h(t') \rangle = \int_t^{t+h} ds \int_{t'}^{t'+h} du \langle \eta(s)\eta(u) \rangle = \int_t^{t+h} ds \int_{t'}^{t'+h} du D\delta(s-u) \quad (3.28)$$

This integral is an easy exercise on delta-function integration. The result is:

$$\langle \omega_h(t)\omega_h(t') \rangle = \begin{cases} D(h - |t - t'|) & |t - t'| \leq h \\ 0 & |t - t'| \geq h \end{cases} \quad (3.29)$$

The important thing to notice is that for the times  $t_n$  that appear in the recursion relation (3.24), one has  $\langle \omega_h(t_n)\omega_h(t'_n) \rangle = Dh\delta_{n,n'}$ , so that the variables  $\omega_h(t_n)$  are independent, Gaussian, random variables of zero mean and variance  $\langle \omega_h(t_n)^2 \rangle = Dh$ . Hence they can be written as

$$\omega_h(t_n) = \sqrt{Dh}u^{(n)} \quad (3.30)$$

where  $u^{(n)}$  are independent Gaussian variables of mean zero and variance unity which can be generated, for instance, by the Box-Muller-Wiener method. The final recursion relation reads:

$$\begin{aligned} x^{(0)} &= x_0 \\ x^{(n+1)} &= x^{(n)} + hF^{(n)} + \sqrt{Dh}u^{(n)} \end{aligned} \quad (3.31)$$

with the obvious notation  $x^{(n)} = x(t_0 + nh)$ ,  $F^{(n)} = F(x(t_0 + nh))$ , etc. This simple example has taught us that the stochastic contribution is of order  $h^{1/2}$  whereas the deterministic contribution is of order  $h$ . This singular expansion is the main reason for the failure of algorithms based on a naïve Taylor series on integer powers of  $h$ . Let us now consider another example:

$$\frac{dx(t)}{dt} = F(x) + G(x)\eta(t) \quad (3.32)$$

Integration of this equation leads to (we have taken  $t_0 = 0$  to simplify notation):

$$x(h) = x(0) + \int_0^h ds F(x(s)) + \int_0^h ds G(x(s))\eta(s) \quad (3.33)$$

To proceed, we Taylor expand the functions  $F(x)$  and  $G(x)$ :

$$\begin{aligned} F(x(s)) &= F(x(0)) + \left. \frac{dF}{dx} \right|_{x=x(0)} (x(s) - x(0)) + o(x(s) - x(0))^2 \\ G(x(s)) &= G(x(0)) + \left. \frac{dG}{dx} \right|_{x=x(0)} (x(s) - x(0)) + o(x(s) - x(0))^2 \end{aligned} \quad (3.34)$$

By substituting these expansions in (3.33) we notice is that at lowest order,  $h^{1/2}$ , we have

$$x(s) - x(0) = G(x(0)) \int_0^s d\eta(u) \quad (3.35)$$

and hence, to have an algorithm correct to  $o(h)$  it suffices to consider the following terms in the expansion:

$$\begin{aligned} x(h) &= x(0) + F(x(0))h + G(x(0))\omega_h(0) + \\ &G(x(0)) \left. \frac{dG}{dx} \right|_{x=x(0)} \int_0^h ds \int_0^s d\eta(s)\eta(u) + o(h^{3/2}) \end{aligned} \quad (3.36)$$

The double integral gives  $\omega_h(0)^2/2$ . Replacing  $\omega_h$  by  $\sqrt{Dh}u$ , see Eq.(3.30), we arrive at the following recursion relation, known as the Milshtein algorithm (Milshtein 1974):

$$\begin{aligned} x^{(0)} &= x_0 \\ x^{(n+1)} &= x^{(n)} + hF^{(n)} + \sqrt{Dh}u^{(n)} + \frac{1}{2}G^{(n)} \left. \frac{dG}{dx} \right|^{(n)} Dh[u^{(n)}]^2 \end{aligned} \quad (3.37)$$

The Milshtein method is an algorithm correct to order  $h$  and can then be considered as the stochastic equivalent of the Euler method to solve numerically ordinary differential equations. Sometimes in the literature (Greiner et al. 1988) the name Euler method for stochastic differential equations (in the Stratonovich sense) is given to the above expression in which the term  $[u^{(n)}]^2$  is replaced by its mean value  $\langle [u^{(n)}]^2 \rangle = 1$ :

$$x^{(n+1)} = x^{(n)} + h \left[ F^{(n)} + \frac{D}{2} G^{(n)} \frac{dG}{dx} \Big|^{(n)} \right] + \sqrt{Dh} u^{(n)} \quad (3.38)$$

This is an algorithm with a worst convergence that the Milshtein algorithm. However, if one is interested only on computing the moments  $\langle x(t)^k \rangle$  of the variable  $x(t)$  one can prove that this Euler algorithm has the same accuracy than the Milshtein algorithm. Given that both algorithms have approximately the same computational complexity, it does not seem justified to use the poorer Euler algorithm instead of the Milshtein algorithm.

It is possible but very tedious to develop higher order algorithms. The good news is that very rarely one really needs them. This is so because, in general, one has to solve numerically the Langevin equations and average the results for different realizations of the noise (and maybe initial conditions). This generates a source of statistical errors coming from the averages which are, in many occasions, greater than the systematic errors due to the order of convergence of the numerical method. So it is usually better to spend the computer time in reducing the statistical errors by increasing the number of samples in the average rather than using a more complicated, higher order, algorithm. However, we would like to mention briefly the stochastic Runge-Kutta type methods because, although they do not yield better results than the Milshtein method for the convergence of the stochastic terms, they do treat better the deterministic terms, increasing in many cases the numerical stability. The simplest stochastic Runge-Kutta type method is the Heun method which, for Eq.(3.32) reads:

$$\begin{aligned} k &= hF(x(0)) \\ l &= \sqrt{Dh} u^{(0)} G(x(0)) \\ y &= x(0) + k + l \\ x(h) &= x(0) + \frac{h}{2} [F(x(0)) + F(y)] + \frac{\sqrt{Dh} u^{(0)}}{2} [G(x(0)) + G(y)] \end{aligned} \quad (3.39)$$

One can prove by Taylor expanding the functions  $F(x)$  and  $G(x)$  that the Heun method reproduces, to order  $h$ , the stochastic part of the Milshtein method (Gard 1987). For the purely deterministic case,  $D = 0$ , the Heun algorithm reduces to the standard  $2^{nd}$  order Runge-Kutta method.

So far, we have only considered one-variable equations. It is very complicated to develop algorithms for the general  $N$ -variable case of Eq.(3.20). The interested reader can find in (Kloeden et al. 1992) an extensive account of these general methods. One can always use the following extension of the Euler method:

$$x_i^{(n+1)} = x_i^{(n)} + h \left[ F_i^{(n)} + \frac{D}{2} \sum_{j=1}^N \sum_{k=1}^N G_{j,k}^{(n)} \frac{\partial G_{i,k}}{\partial x_j} \Big|^{(n)} \right] + \sqrt{Dh} \sum_{j=1}^N G_{i,j}^{(n)} u_j^{(n)} \quad (3.40)$$

where the  $u_i^{(n)}$  are independent Gaussian variables of mean 0 and variance 1:

$$\langle u_i^{(n)} u_{i'}^{(n')} \rangle = \delta_{n,n'} \delta_{i,i'} \quad (3.41)$$

In the case of *diagonal* noise, (Kloeden et al. 1992), i.e. one in which the noise terms do not couple variable at different lattice locations:

$$G_{i,j}(x_1, \dots, x_N) = G_i(x_i) \delta_{i,j} \quad (3.42)$$

one can generalize the Milshtein algorithm to:

$$x_i^{(n+1)} = x_i^{(n)} + h F_i^{(n)} + \sqrt{Dh} u_i^{(n)} + \frac{1}{2} G_i^{(n)} \frac{dG_i}{dx_i} \Big|^{(n)} Dh [u_i^{(n)}]^2 \quad (3.43)$$

In next chapters we will show some applications of the numerical integration of Langevin field equations. In Chapter 4 the Langevin equation will be combined with Monte Carlo techniques to yield a very useful tool to study equilibrium properties: the *hybrid Monte Carlo algorithm*. In Chapter 5 we will be concerned with the dynamics of growth in two very different situations: phase separation and growth of random surfaces.

## 4. Hybrid Monte Carlo

### 4.1. Introduction

The numerical integration of Langevin equations described in the previous chapter has been used extensively as an alternative to Monte Carlo methods to sample the canonical distribution. The idea is to run the Langevin equation up to the stationary state (a process similar to thermalization for Monte Carlo) and then average the different field configurations produced over time. An advantage of this procedure as compared to Monte Carlo is that it is not necessary to compute the change in energy every time a single variable is updated. This is important when the computation of the Hamiltonian is extremely time consuming. Such situation arises in lattice gauge theories that include dynamical fermions (Zinn Justin 1989). The effective Hamiltonian is so complicated for those systems that it is prohibitive having to compute  $\mathcal{H}$   $N$  times per Monte Carlo step.

An advantage of Monte Carlo methods over the Langevin integration is that the only errors in Monte Carlo are of statistical origin due to the limited number of configurations in the sample (and also to the correlations amongst them) and are known to behave as the inverse square root of the number of configurations. On top of the statistical errors, the numerical integration of the Langevin equation produces systematic errors due to the finiteness of the time step used in the integration. These systematic errors are much more cumbersome, since

their numerical importance is not known a priori. A recent idea aims to combine the best of both worlds. In the so called *Hybrid Monte Carlo* algorithm (Duane et al. 1987) the numerical integration of the stochastic differential equation is used as the proposal to which an acceptance/rejection step is applied. Before we develop this beautiful idea, we need to develop some concepts related to the microcanonical ensemble.

## 4.2. Molecular Dynamics

Molecular dynamics (Allen et al. 1987) offers a direct numerical approach to the behavior of a system with many degrees of freedom. Let  $[X] = (x_1, x_2, \dots, x_N)$  be microscopic variables and  $[P] \equiv (p_1, p_2, \dots, p_N)$  their conjugate momenta. The Hamiltonian function is:

$$\mathcal{H}(X, P) = \mathcal{V}(X) + \mathcal{T}(P) = \mathcal{V}(X) + \frac{P^2}{2} = \mathcal{V}(X) + \sum_{i=1}^N \frac{p_i^2}{2} \quad (4.1)$$

The representative point of the system  $[x, p]$  evolves in phase space according to Hamilton equations

$$\begin{aligned} \frac{dx_i}{dt} &= p_i \\ \frac{dp_i}{dt} &= F_i = -\frac{\partial \mathcal{H}}{\partial x_i} \end{aligned} \quad (4.2)$$

and the movement is confined to the hypersurface of constant energy,  $\mathcal{H}(X, P) = E$ . This property can be used to perform averages on the microcanonical ensemble according to the ergodic principle that allows substitution of ensemble averages  $\langle G \rangle$  by time averages  $\bar{G}$ :

$$\bar{G} = \lim_{T \rightarrow \infty} \frac{1}{T - t_0} \int_{t_0}^T dt G(x(t), p(t)) \quad (4.3)$$

To perform this time average we can integrate numerically Hamilton equations. In this way we can obtain equilibrium properties as well as dynamical (transitory) properties. However, similarly to what happened when using the numerical integration of Langevin equations to sample the canonical distribution, two kinds of errors are present: (i) statistical errors due to the finite number of samplings and (ii) systematic errors due to the time discretization used in the numerical solution.

The simplest integration scheme is the Euler method in which the derivative of a function  $b(t)$  is approximated by:

$$\frac{db(t)}{dt} = \frac{b(t + \delta t) - b(t)}{\delta t} + O(\delta t) \quad (4.4)$$

Using this approximation, Hamilton equations become:



$$\left. \begin{aligned} \frac{x_i(t + \delta t) - x_i(t)}{\delta t} + O(\delta t) &= p_i(t) \\ \frac{p_i(t + \delta t) - p_i(t)}{\delta t} + O(\delta t) &= F_i(t) \end{aligned} \right\} \quad (4.5)$$

From where one gets the following recursive relations:

$$\left. \begin{aligned} x_i(t + \delta t) &= x_i(t) + \delta t p_i(t) + O(\delta t)^2 \\ p_i(t + \delta t) &= p_i(t) + \delta t F_i(t) + O(\delta t)^2 \end{aligned} \right\} \quad (4.6)$$

The local integration error is of order  $o(\delta t)^2$ . After  $n$  integration steps,  $n\delta t = t$  the integration error is  $\epsilon = O(n(\delta t)^2) = O(t\delta t) = O(\delta t)$ . This error is usually too large and demands choosing a very small time step for accuracy and numerical stability. Of course, there are many other higher-order integration methods. Amongst all of them the *leap frog* algorithm will be particularly suited to our needs. The algorithm is:

$$\begin{aligned} x_i(t + \delta t) &= x_i(t) + \delta t \left( p_i(t) + \frac{\delta t}{2} F_i(t) \right) \\ p_i(t + \delta t) &= p_i(t) + \frac{\delta t}{2} (F_i(t) + F_i(t + \delta t)), \quad i = 1, \dots, N \end{aligned} \quad (4.7)$$

and has a local error of order  $o(\delta t)^3$ . Although this is an improvement over the Euler method, still discretization errors show up, for instance, in the fact that the energy is not exactly conserved. These energy fluctuations (of numerical origin) imply that the microcanonical ensemble is not sampled exactly.

Other properties of the Hamiltonian dynamics are, however, preserved by this algorithm. In particular leap frog satisfies:

(i) Time reversal. If:

$$\begin{pmatrix} X(t) \\ P(t) \end{pmatrix} \xrightarrow{\delta t} \begin{pmatrix} X(t + \delta t) \\ P(t + \delta t) \end{pmatrix} \quad (4.8)$$

then reversal of the momenta will take us back to the original value for  $[X]$  and the reversed value for the momenta  $[P]$ :

$$\begin{pmatrix} X(t + \delta t) \\ -P(t + \delta t) \end{pmatrix} \xrightarrow{\delta t} \begin{pmatrix} X(t + 2\delta t) \\ P(t + 2\delta t) \end{pmatrix} = \begin{pmatrix} X(t) \\ -P(t) \end{pmatrix} \quad (4.9)$$

(ii) Area preserving. The Jacobian of the change of variables induced during the time evolution is equal to one:

$$J \left( \frac{X(t + \delta t), P(t + \delta t)}{X(t), P(t)} \right) = 1 \quad (4.10)$$

But, let us repeat it once again, it does not conserve energy:

$$\delta \mathcal{H} = \mathcal{H}(t + \delta t) - \mathcal{H}(t) = O[(\delta t)^m] \neq 0 \quad (4.11)$$

Summarizing, the numerical integration by the leap-frog method induces a mapping given by Eq.(4.7) that evolves the whole system from one point of phase

space to another point with some error in the conservation of energy or, in other words, leap-frog defines a global change of variables that almost conserves energy. The idea naturally appears to combine this mapping with the acceptance/rejection technique typical of the Monte Carlo method. This is the main idea behind the hybrid Monte Carlo method introduced by Duane et al. (1987) that we now develop in more detail.

### 4.3. Hybrid Monte Carlo

Let us consider a scalar field  $\Phi = (\Phi_1, \dots, \Phi_N)$  on the regular hypercubic lattice  $\Lambda$ . The statistical properties of the system are given by the Gibbs factor  $\exp[-\mathcal{H}([\Phi])]$  (we take  $\beta = 1$  to simplify notation), with a Hamiltonian  $\mathcal{H}([\Phi])$ . In order to define a Hamiltonian dynamics we introduce some **fictitious** momenta fields  $[Q] = (Q_1, \dots, Q_N)$  and a new Hamiltonian function

$$\hat{\mathcal{H}}([\Phi], [Q]) = \mathcal{H}([\Phi]) + \frac{Q^2}{2} = \mathcal{H}(\Phi) + \sum_{i=1}^N \frac{Q_i^2}{2} \quad (4.12)$$

Since

$$\exp[-\hat{\mathcal{H}}([\Phi], [Q])] = \exp[-\mathcal{H}([\Phi])] \exp\left[-\frac{Q^2}{2}\right] \quad (4.13)$$

from the statistical point of view, the momenta variables  $(Q_1, \dots, Q_N)$  are simply Gaussian distributed independent variables. By independent we imply not just independent of each other but also of the fields  $[\Phi]$ .

According to our plan, we want to make a proposal  $g_{\hat{\mathbf{x}}}(\Phi'|\Phi)$  by using the numerical solution of Hamilton equations. We specify some initial values for the momenta  $[Q]$  according to the Gaussian distribution  $\exp\left[-\frac{Q^2}{2}\right]$  and then integrate numerically, obtaining new values for the field:

$$\begin{pmatrix} \Phi \\ Q \end{pmatrix} \xrightarrow{\delta t} \begin{pmatrix} \Phi' \\ Q' \end{pmatrix} \quad (4.14)$$

This mapping can be done, for instance, by using the leap-frog algorithm, Eqs.(4.7):

$$\begin{aligned} \Phi'_i &= \Phi_i + \frac{\delta t^2}{2} F_i + \delta t Q_i \\ Q'_i &= Q_i + \frac{\delta t}{2} (F_i + F'_i) \end{aligned} \quad (4.15)$$

We use this mapping to construct the proposal probability  $g_{\hat{\mathbf{x}}}(\Phi'|\Phi)$ . More specifically, since the probability distribution function of the  $Q$  variables is  $\exp(-Q^2/2)$  we can write out the probability density function  $g_{\hat{\mathbf{x}}}(\Phi'|\Phi)$  as:

$$g_{\hat{\mathbf{x}}}(\Phi'|\Phi) d\Phi' = dQ e^{-\frac{Q^2}{2}} \quad (4.16)$$

being  $Q$  the value of the momenta necessary to go from  $\Phi$  to  $\Phi'$ . The key point now is to choose an acceptance probability such that detailed balance is satisfied with the hamiltonian  $\mathcal{H}([\Phi])$ :

$$e^{-\mathcal{H}([\Phi])} g_{\hat{\mathbf{x}}}(\Phi'|\Phi)h(\Phi'|\Phi) = e^{-\mathcal{H}([\Phi'])} g_{\hat{\mathbf{x}}}(\Phi|\Phi')h(\Phi|\Phi') \quad (4.17)$$

The proposal  $g_{\hat{\mathbf{x}}}(\Phi|\Phi')$  is given again by integration of Hamilton equations with a suitable value for the momenta,  $[Q'']$ :

$$\begin{pmatrix} \Phi' \\ Q'' \end{pmatrix} \xrightarrow{\delta t} \begin{pmatrix} \Phi \\ Q''' \end{pmatrix} \quad (4.18)$$

from where one gets for  $g_{\hat{\mathbf{x}}}(\Phi|\Phi')$ :

$$g_{\hat{\mathbf{x}}}(\Phi|\Phi')d\Phi = dQ'' e^{-\beta \frac{Q''^2}{2}} \quad (4.19)$$

being  $Q''$  the value of the momenta necessary to go from  $\Phi'$  to  $\Phi$ . The detailed balance condition becomes:

$$e^{-\frac{Q^2}{2}} e^{-\mathcal{H}(\Phi)} h(\Phi'|\Phi) d\Phi dQ = e^{-\frac{Q'^2}{2}} e^{-\mathcal{H}(\Phi')} h(\Phi|\Phi') d\Phi' dQ'' \quad (4.20)$$

Now the following properties apply:

- (i) If the numerical integration satisfies time reversal,  $Q'' = -Q'$  hence the Jacobian  $dQ''/dQ'$  is equal to 1.

$$e^{-\frac{Q^2}{2}} e^{-\mathcal{H}(\Phi)} h(\Phi'|\Phi) d\Phi dQ = e^{-\frac{Q'^2}{2}} e^{-\mathcal{H}(\Phi')} h(\Phi|\Phi') d\Phi' dQ' \quad (4.21)$$

- (ii) If it satisfies the area conserving property  $dQd\Phi = dQ'd\Phi'$

$$e^{-\frac{Q^2}{2}} e^{-\mathcal{H}(\Phi)} h(\Phi'|\Phi) = e^{-\frac{Q'^2}{2}} e^{-\mathcal{H}(\Phi')} h(\Phi|\Phi') \quad (4.22)$$

Which, using Eq.(4.13), can be written as:

$$e^{-\hat{\mathcal{H}}(\Phi, Q)} h(\Phi'|\Phi) = e^{-\hat{\mathcal{H}}(\Phi', Q')} h(\Phi|\Phi') \quad (4.23)$$

We already know a possible solution to this equation, the Metropolis solution:

$$h(\Phi'|\Phi) = \min(1, e^{-\delta\hat{\mathcal{H}}}) \quad (4.24)$$

where  $\delta\hat{\mathcal{H}} = \hat{\mathcal{H}}(t + \delta t) - \hat{\mathcal{H}}(t)$  is the change of total energy produced in the evolution by time step  $\delta t$ . Notice that, although we have made a global change by updating all the variables  $\Phi_i$  at once,  $\delta\hat{\mathcal{H}}$  is controllable and the acceptance probability can be made to stay within reasonable limits (close to 50%). It is not necessary to make the acceptance/rejection decision every time step  $\delta t$ . Instead, usually one integrates during  $n$  time steps before acceptance/rejection.

We give now an explicit implementation of the *Hybrid Monte Carlo method*:

- (i) Generate independent Gaussian momenta  $[Q]$ .
- (ii) Update the system  $[\Phi, Q]$  using leap-frog (or any other time reversible, area preserving method) during a time  $\Delta t = n\delta t$

$$\begin{pmatrix} \Phi(t) \\ Q(t) \end{pmatrix} \xrightarrow{\delta t} \begin{pmatrix} \Phi(t + \delta t) \\ Q(t + \delta t) \end{pmatrix} \xrightarrow{\delta t} \cdots \xrightarrow{\delta t} \begin{pmatrix} \Phi(t + \Delta t) \\ Q(t + \Delta t) \end{pmatrix}$$

(iii) Compute  $\Delta\mathcal{H} = \hat{\mathcal{H}}(t + \Delta t) - \hat{\mathcal{H}}(t)$

(iv) Accept configuration  $\Phi' = \Phi(t + \Delta t)$  with probability  $\min(1, e^{-\Delta\mathcal{H}})$

Hybrid Monte Carlo constitutes an important tool for the simulation of field theories and it has been applied successfully to  $SU(n)$  field theory (Sexton et al. 1992), Lennard-Jones systems (Mehlig et al. 1992a),  $\Phi^4$  scalar field (Mehlig et al. 1992b),  $X - Y$  model (Gupta 1992), polymer systems (Forrest et al. 1994; Irbeck 1994), etc. and it is today a routine tool of great applicability.

There is another important aspect of the Hybrid Monte Carlo algorithm, namely its relation with Langevin dynamics. Remember the Langevin equation defining model  $A$ . Eq.(3.1):

$$\frac{\partial\Phi_i}{\partial\tau} = -\frac{\delta\mathcal{H}}{\delta\Phi_i} + \eta_i(\tau) = F_i(\tau) + \eta_i(\tau) \quad (4.25)$$

where the noise terms satisfies:

$$\langle \eta_i(\tau)\eta_j(\tau') \rangle = 2\delta_{ij}\delta(\tau - \tau') \quad (4.26)$$

The numerical solution using the Euler algorithm proceeds via the following recursion relation:

$$\Phi_i(\tau + \delta\tau) = \Phi_i(\tau) + \delta\tau F_i(\tau) + \sqrt{2\delta\tau}u_i(\tau) \quad (4.27)$$

where  $u_i(\tau)$  are Gaussian variables of mean 0 and correlations  $\langle u_i(\tau)u_j(\tau) \rangle = \delta_{ij}$ . This is exactly the same evolution scheme of the leap-frog algorithm, Eq.(4.15), if we make the following relation between the time steps of the two methods:  $\delta\tau = \delta t^2/2$ . We conclude that Hybrid Monte Carlo uses the same updating scheme than Langevin (with a different unit of time), but the presence of the acceptance/rejection step makes the method independent of the time step used in the Langevin integration, so avoiding the systematic errors so difficult to deal with in the Langevin method.

#### 4.4. Generalized Hybrid Monte Carlo

It is clear that Hybrid Monte Carlo relies upon a numerical integration of the differential equations for a dynamical system in which the energy is conserved. We will now introduce a general class of dynamical equations that also conserve energy. This will give rise to a generalization of the hybrid Monte Carlo method (Ferreira et al. 1993; Toral et al. 1994).

The generalization goes as follow: to each field  $\Phi_i$  we assign a vector momenta variable  $\mathbf{Q}_i = (q_i^1, \dots, q_i^D)$  of arbitrary dimension  $D$ . We introduce the following equations of motion:

$$\begin{aligned} \frac{d\Phi_i}{dt} &= \sum_{s=1}^D \sum_{j=1}^N (\mathcal{A}^s)_{ij} Q_j^s \\ \frac{dq_i^s}{dt} &= \sum_{j=1}^N (\mathcal{A}^s)_{ji} F_j \end{aligned} \quad s = 1, \dots, D \quad (4.28)$$

or, written in more compact vector notation:

$$\begin{aligned} \frac{d\Phi}{dt} &= \sum_{s=1}^D \mathcal{A}^s Q^s \\ \frac{dQ^s}{dt} &= (\mathcal{A}^s)^\top F \quad s = 1, \dots, D \end{aligned} \quad (4.29)$$

where the  $\mathcal{A}^s$  are some linear operators which can be represented as a matrix, and  $F_j$  represents the force as computed from the Hamiltonian  $-\frac{\partial}{\partial \Phi_j} \mathcal{H}$ . We also introduce a total Hamiltonian  $\hat{\mathcal{H}}$ , including the kinetic energy terms:

$$\hat{\mathcal{H}}([\Phi, Q]) = \mathcal{H}(\Phi) + \sum_{i=1}^N \sum_{s=1}^D \frac{(q_i^s)^2}{2} \quad (4.30)$$

The reader can easily verify that the proposed dynamics in Eqs.(4.28) exactly conserves energy, i.e.,  $d\hat{\mathcal{H}}/dt = 0$ . The standard hybrid Monte Carlo can be obtained from the above dynamics setting  $D = 1$  and  $\mathcal{A}$  equal to the identity operator.

For the approximate integration of the previous equations of motion the leap-frog scheme can be used, introducing a discrete mapping  $[\Phi(t), Q(t)] \rightarrow [\Phi(t + \delta t), Q(t + \delta t)] = G^{\delta t}([\Phi(t), Q(t)])$ , dependent on the time step  $\delta t$  chosen. The leap-frog approximation reads:

$$\begin{aligned} \Phi' &= \Phi + \delta t \sum_{s=1}^D \mathcal{A}^s Q^s + \frac{(\delta t)^2}{2} \sum_{s=1}^D \mathcal{A}^s (\mathcal{A}^s)^\top F([\Phi]) \\ Q'^s &= Q^s + \frac{\delta t}{2} (\mathcal{A}^s)^\top (F([\Phi]) + F([\Phi'])) \end{aligned} \quad (4.31)$$

It can be shown that this leap-frog approximation for arbitrary matrices  $\mathcal{A}^s$  satisfies the properties of time reversibility and area preserving. However, again as a result of the time discretization used in the leap-frog scheme, the total energy is no longer conserved although its variation can be controlled by varying  $\delta t$ . We define yet another mapping obtained iterating  $n$  times the previous mapping, i.e.  $\mathcal{G} = (G^{\delta t})^n$ . To satisfy detailed balance the configuration obtained when one applies  $\mathcal{G}$  is accepted with probability  $\min[1, \exp(-\Delta\hat{\mathcal{H}})]$ , where  $\Delta\hat{\mathcal{H}} = \hat{\mathcal{H}}(\mathcal{G}([\Phi, Q])) - \hat{\mathcal{H}}([\Phi, Q])$ . As in the standard hybrid Monte Carlo, the momenta variables are refreshed after every acceptance/rejection step according to a Gaussian distribution of independent variables.

We have defined a general class of hybrid Monte Carlo-type methods characterized by a particular choice of matrices  $\mathcal{A}^s$ . One can choose the matrices  $\mathcal{A}^s$  that better suit a particular problem. This generalized hybrid Monte Carlo method turns out to be related to the method introduced in reference (Batrouni et al. 1985) using the numerical integration of a Langevin equation with a matrix time-step. This method is based upon the observation that the stationary solution of the Langevin equation (4.25) can be obtained approximately by the recursion relation:

$$\Phi_i(\tau + \delta\tau) = \Phi_i(\tau) + \sum_{j=1}^N \left[ -\delta\tau \epsilon_{ij} \frac{\delta\mathcal{H}}{\delta\Phi_j} + \sqrt{2\delta\tau} \sqrt{\epsilon_{ij}} u_j \right] \quad (4.32)$$

Where  $\epsilon_{ij}$  is an arbitrary matrix and  $u_j$  is a Gaussian variable of mean zero and correlations  $\langle u_i u_j \rangle = \delta_{ij}$ . Comparing with (4.31) we can see that it corresponds exactly to the one step leap-frog approximation of the generalized hybrid Monte Carlo method introduced above taking  $D = 1$  and if we identify:  $(\delta t)^2/2 = \delta\tau$  and  $\mathcal{A}\mathcal{A}^\top = \epsilon$ . The main difference between the two methods is the presence of an acceptance/rejection step in the generalized hybrid Monte Carlo. In this sense, we can say that the generalized hybrid Monte Carlo method introduced in this section makes exact (in the sense that averages are not biased by the choice of the time step) the numerical integration of the Langevin equation using a matrix time step introduced in reference (Batrouni et al. 1985).

#### 4.5. Fourier Acceleration

Near a second order phase transition simulations become very time consuming due to the large correlations which appear in the neighbourhood of the critical point. This enhancement of the time correlations, known as **critical slowing down**, greatly increases the statistical errors in a Monte Carlo simulation making it necessary to average over a large number of configurations in order to obtain values with an acceptable error. The physical origin of critical slowing down is related to the fact that, near the critical point, fluctuations of all wavelengths are present, each with a characteristic time scale. In a Langevin integration, for example, the integration step has to be small enough to treat correctly the short-wavelength components but, on the other hand, the integration has to proceed for a sufficient number of steps to have the long-wavelengths mode evolve significantly. In the **Fourier acceleration** method (Parisi 1983; Batrouni et al. 1985), an integration technique is developed such that each mode evolves with its own effective time step, large for long wavelengths and small for small wavelengths. Whether Fourier acceleration is successful in completely overcoming critical slowing down for a particular system depends on to which extent the evolution of the different Fourier modes can be considered indeed independent of each other.

Fourier acceleration has been traditionally framed in the language of Langevin equations. But we know already that numerical solutions of Langevin equations introduce systematic errors due to the finiteness of the time step used in the numerical integration. It is possible to develop exact Fourier acceleration methods in the context of the generalized hybrid Monte Carlo algorithm developed in the previous section. The basic ingredient is the following:

- Use matrices  $\mathcal{A}^s$  whose elements are diagonal in Fourier space.

The optimal choice for these matrices depends on the system we are considering. Let us now discuss the example of the Gaussian model, for which the

technique is most successful in reducing critical slowing down. Of course, the example is purely academic, since the Gaussian model can be solved analytically without having to resource to numerical methods.

The Gaussian model is defined by the following Hamiltonian for the scalar fields  $[\Phi_i]$  (Parisi, 1988):

$$\mathcal{H} = \sum_{i=1}^N \left[ \frac{\mu}{2} \Phi_i^2 + \frac{1}{2} |\nabla_L \Phi_i|^2 \right] \quad (4.33)$$

index  $i$  runs over the  $N = L^d$  sites of a  $d$ -dimensional regular lattice  $\Lambda$ , with periodic boundary conditions;  $\nabla_L$  is the usual lattice discretized version of the gradient operator;  $\mu$  is a parameter called the *field mass*. Although this model does not show a real phase transition, the critical slowing down occurs for the critical value"  $\mu = 0$ .

To implement the generalized hybrid Monte Carlo method we choose the number of momenta variables associated to a given field equal to 1 (i.e.  $D = 1$  in Eq.(4.28)). The total Hamiltonian  $\hat{\mathcal{H}}$ , Eq.(4.30), including the kinetic energy terms can be written in terms of the Fourier transform of fields and momenta:

$$\hat{\mathcal{H}} = \sum_{k=1}^N \left[ \frac{\omega_k^2}{2} |\hat{\Phi}_k|^2 + \frac{1}{2} |\hat{Q}_k|^2 \right] \quad (4.34)$$

the sum runs over the  $N$  points in the reciprocal lattice,  $k = (k_1, \dots, k_d)$ . Here  $\omega_k^2 = \mu + 4 \sum_{j=1}^d \sin^2(k_j/2)$  and  $\hat{\Phi}_k$  and  $\hat{Q}_k$  stand for the fields and momenta variables in Fourier space. Notice that the total Hamiltonian is a sum of terms each one depending on a single mode  $k$ , such that the modes evolve independently of each other.

According to the discussion above, we implement Fourier acceleration by choosing the matrix  $\mathcal{A}$ , generating the dynamics, diagonal in Fourier space. Let us denote by  $\hat{A}_k$  the diagonal elements of the matrix  $\mathcal{A}$  in Fourier space. After  $n$  leap-frog steps, the evolution equations (4.31) imply:

$$\begin{bmatrix} \omega_k \hat{\Phi}_k(n\delta t) \\ \hat{Q}_k(n\delta t) \end{bmatrix} = M_k^n \begin{bmatrix} \omega_k \hat{\Phi}_k(0) \\ \hat{Q}_k(0) \end{bmatrix} \quad k = 1, \dots, N \quad (4.35)$$

The dynamical properties of the algorithm are determined by matrices  $M_k^n$ , given by:

$$M_k^n = \begin{pmatrix} \cos(n\theta_k) & \sin(n\theta_k)/\cos(\theta_k/2) \\ -\cos(\theta_k/2)\sin(n\theta_k) & \cos(n\theta_k) \end{pmatrix} \quad (4.36)$$

where we have introduced  $\theta_k = \cos^{-1}(1 - c_k^2/2)$  and  $c_k = \hat{A}_k \omega_k \delta t$ . We see explicitly how in this model different modes evolve independently of each other. The fact that the evolution equations are linear greatly simplifies the analysis of this problem (this is similar to the standard hybrid Monte Carlo where  $\mathcal{A} = 1$ , Kennedy et al. 1991).

We can now answer the question of which is the optimal choice for matrix  $\mathcal{A}$ . If we choose the following values for the diagonal elements of  $\mathcal{A}$  in Fourier space:

$$\hat{A}_k = 1/\omega_k = \left[ \mu + 4 \sum_{j=1}^d \sin^2(k_j/2) \right]^{-1/2} \quad (4.37)$$

the evolution matrices  $M_k^n$  become independent of  $\mu$  and  $k$  such all the modes evolve with the same effective time step  $A_k \omega_k \delta t = \delta t$ . This choice for  $\hat{A}_k$  reduces completely critical slowing down because correlation times (which are related to the eigenvalues of  $M_k^n$ ) become independent of the mass  $\mu$  even when  $\mu$  tends to zero and the model becomes critical.

## 5. Applications in Domain Growth

### 5.1. Dynamics of first order phase transitions

#### 5.1.1. The Cahn-Hilliard-Cook Equation

Many binary (A-B) mixtures which are homogeneous at high temperature  $T_0$ , phase separate when quenched below a certain critical value ( $T_c$ ) of the temperature. In the final equilibrium configuration two phases coexist: the A and B phases, each one rich in the A and B material respectively (see Fig.12). We assume that the equilibrium properties of such mixture are given by the Gibbs distribution at the final temperature  $T_f$  and we are concerned here with the way the mixture reaches this thermal equilibrium, i.e. with the dynamical process of phase separation (Gunton et al. 1983; Binder 1990). In many occasions the relevant equations of motion are those of model B defined in a previous chapter, equations that we now justify from a more physical point of view.

The relevant field  $\Phi$  in this case is a scalar field representing the local concentration difference of the two components of the mixture  $\Phi(\mathbf{r}, t) = \rho_A(\mathbf{r}, t) - \rho_B(\mathbf{r}, t)$ . This field obviously follows a conservation law:

$$m \equiv \frac{1}{V} \int_V d\mathbf{r} \Phi(\mathbf{r}, t) \longrightarrow \frac{dm}{dt} = 0 \quad (5.1)$$

( $V$  is the system volume). This global conservation law can be expressed in local terms by means of a continuity equation:

$$\frac{\partial \Phi(\mathbf{r}, t)}{\partial t} + \nabla \cdot \mathbf{J}(\mathbf{r}, t) = 0 \quad (5.2)$$

The conservation current  $\mathbf{J}(\mathbf{r}, t)$  provides the driving force for field diffusion. Equilibrium is reached when the chemical potentials of the two substances are equal and uniform. Thus, we write the current  $\mathbf{J}(\mathbf{r}, t)$  as the sum of the gradient of the difference of chemical potentials of each component  $\mu = \mu_A - \mu_B$  (this is the well known Fick's law) plus a random, noise type term to take properly into account thermal fluctuations:



**Fig. 12.** Schematic phase diagram of a binary mixture  $A - B$ . The system is homogeneous at temperature  $T_0$ . After quenching at temperature  $T_f$  below  $T_c$  the system phase separates. At sufficiently late times, phase  $A$  and  $B$  coexist. The evolution proceeds differently if the quench is in region  $I$  or  $II$ , separated by the spinodal line (dashed line).

$$\mathbf{J}(\mathbf{r}, t) = -\Gamma \nabla \mu + \boldsymbol{\eta}(\mathbf{r}, t) \quad (5.3)$$

$\Gamma$  is a positive coefficient called *mobility*. The noise variables  $\boldsymbol{\eta} = (\eta_1, \dots, \eta_d)$  are Gaussian distributed, of mean zero and correlations:

$$\langle \eta_k(\mathbf{r}, t) \eta_{k'}(\mathbf{r}', t') \rangle = 2k_B T \delta_{k, k'} \delta(\mathbf{r} - \mathbf{r}') \delta(t - t') \quad (5.4)$$

The chemical potential is obtained as the derivative of  $\mathcal{H}$  with respect to the field:

$$\mu(\mathbf{r}, t) = \frac{\delta \mathcal{H}([\Phi])}{\delta \Phi(\mathbf{r}, t)} \quad (5.5)$$

Combining Eqs.(5.2) to (5.5), we reobtain model B equation, which in the context of phase separation is called the Cahn-Hilliard-Cook (CHC) equation (Cahn et al. 1958; Cook 1970):

$$\frac{\partial \Phi(\mathbf{r}, t)}{\partial t} = \nabla^2 \left( \Gamma \frac{\delta \mathcal{H}([\Phi])}{\delta \Phi(\mathbf{r}, t)} \right) + \xi(\mathbf{r}, t) \quad (5.6)$$

where  $\xi(\mathbf{r}, t) = \nabla \cdot \boldsymbol{\eta}(\mathbf{r}, t)$ . If we adopt expression (2.1) for the Hamiltonian we arrive at the following lattice form for the CHC equation:

$$\frac{d\Phi_i(t)}{dt} = \Gamma k_B T \nabla_L^2 (-b\Phi_i + u\Phi_i^3 - K \nabla_L^2 \Phi_i) + \xi_i(t) \quad (5.7)$$

If we perform the same field and time rescaling than in the section 3.2 for model A, we obtain the simpler equation:

$$\frac{d\phi_i(t)}{dt} = \nabla_L^2 (-\theta b \phi_i + \chi \phi_i^3 - \nabla_L^2 \phi_i) + \xi_i(t) \quad (5.8)$$

with correlations:

$$\langle \xi_i(t) \xi_j(t') \rangle = -2\delta(t-t') \nabla_L^2 \delta_{i,j} \quad (5.9)$$

The Cahn–Hilliard–Cook equation is a very complicated stochastic, nonlinear, partial differential equation that has so far defied an exact analytical treatment. Very successful systematic approximations valid for early times have been developed (Grant et al. 1985). For the late stages of the dynamics the most successful theory is that of Langer, Bar-on and Miller (Langer et al. 1975), based on some approximative mode decoupling. This theory, however, is very difficult to improve upon due to its non-perturbative nature. It seems at this stage that it is necessary to resource to numerical studies to analyze the long-time behaviour of the equation. One can integrate the CHC equation using any of the stochastic numerical methods developed in section 3.3. In Figure 13 we have plotted configurations resulting of a numerical integration using the simple Euler method. In this figure we observe that the evolution depends on the location of the quench point in phase diagram. For critical ( $m = 0$ ) or near critical concentrations, (region *I* of the phase diagram in Fig.12), the system is unstable against long wavelength, small amplitude fluctuations. This generates initially a highly interconnected pattern that coarsens with time. The system is said to evolve by *spinodal decomposition*. If, on the other hand, the system is in region *II*, between the so-called spinodal line and the coexistence curve, the system is unstable against the formation of nuclei of the minority phase. These nuclei evolve in time in such a way that large droplets grow at the expense of small droplets. This is the evolution for *nucleation and growth*. Although this simple picture is thought to be of general validity, it is not possible to sharply separate the domains in the phase diagram where the two processes dominate (Binder 1990; Chakrabarti 1992).

### 5.1.2 The Kawasaki model

A more microscopic approach to the dynamics of phase separation is the one given by the Kawasaki dynamical version of the Ising model (Kawasaki 1972). In each site of a regular lattice  $A$  we define two occupation variables,  $C_i^A$  and  $C_i^B$ , such that  $C_i^A$  takes the value 1 (0) if in site  $i$  there is (there is not) an  $A$ -atom.  $C_i^B$  is defined in a similar way. If we consider the situation in which every site of the lattice is occupied by either  $A$ -atoms or  $B$ -atoms, i.e. that there are no vacancies,  $C_i^A$  and  $C_i^B$  verify  $C_i^A + C_i^B = 1$ . We define a new variable  $S_i$  as:

$$S_i = C_i^A - C_i^B \quad (5.10)$$

from where it is easy to deduce:

$$C_i^A = \frac{1 + S_i}{2} \quad C_i^B = \frac{1 - S_i}{2} \quad (5.11)$$

**Fig. 13.** Time evolution of the Kawasaki model and the Cahn–Hilliard–Cook (CHC) model. Positive (negative) values for the fields or the spins are indicated by white (black) regions. Notice the similarity between the Kawasaki model (left column) and the CHC (center column) both at critical concentration ( $m = 0$ ). In these two cases, the system evolves by spinodal decomposition. In the right column, we plot a solution of the CHC model for an off-critical quench (in this case  $m = 0.2$ ). Notice that evolution proceeds via the formation of nuclei of the minority phase that coarsen and grow.

$S_i$  is an Ising type variable taking the value  $+1$  ( $-1$ ) if in site  $i$  there is a  $A$ -atom ( $B$ -atom). Let us denote by  $J_{i,j}(AA)$ ,  $J_{i,j}(BB)$ ,  $J_{i,j}(AB)$  the strength of the interaction between  $AA$ ,  $BB$  and  $AB$  atoms, respectively, at locations  $i$  and  $j$  in the lattice. The Hamiltonian is:

$$\mathcal{H} = \sum_{i,j} [J_{i,j}(AA)C_i^A C_j^A + J_{i,j}(BB)C_i^B C_j^B + J_{i,j}(AB)C_i^A C_j^B] \quad (5.12)$$

Writing the occupation variable  $C_i^A$  and  $C_i^B$  in terms of the spin variables  $S_i$  one gets, apart from a constant factor:

$$\mathcal{H} = - \sum_{i,j} J_{i,j} S_i S_j \quad (5.13)$$

Where  $J_{i,j}$  is some linear combination of the interaction strengths  $J_{i,j}(AA)$ ,  $J_{i,j}(BB)$  and  $J_{i,j}(AB)$ . We take now the approximation that the forces between atoms are of short range nature, such that the only relevant interactions are the ones between nearest neighbour sites on the lattice and we further assume that they are all equal to a constant value  $J$ . In this case, the Hamiltonian reduces to:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j \quad (5.14)$$

where the sum runs over the nearest neighbour pairs on the lattice. This is nothing but the Hamiltonian of the celebrated Ising model. In our binary mixture model, though, there is the restriction that  $N^{-1} \sum_i S_i = m$  must be a fixed constant. One can prove (Pathria 1972) that the grand canonical partition function of the binary mixture model is equal to the canonical partition function of the standard Ising model. Many equilibrium properties of the binary mixture, such as the phase diagram, can thus be obtained from the corresponding ones of the Ising model.

Our model for a binary mixture is indeed too simplified and it is absent of many realistic features present in real materials, such as defects in the lattice, no perfect symmetry between  $A$  and  $B$  compounds leading to asymmetric phase diagrams, vacancies, etc. One can not pretend, then, to reproduce all of the features present in these substances. In fact our hamiltonian is so simple that it is absent of any “natural” dynamics such as the one given by Hamilton equations. Kawasaki introduced an *stochastic* dynamics for the binary mixture model in the same spirit that Glauber dynamics for the Ising model (Glauber, 1963). The thermal fluctuations, mediated by the lattice phonons, induce random transitions in which atoms exchange positions in the lattice, see Fig.14 (in the ordinary Glauber dynamics, spins flip randomly their value, this is forbidden here due to the conservation law). This exchange process aims to mimic, in a very crude way, the diffusion process occurring in real materials.

In order to ensure that the system will asymptotically reach the correct equilibrium state given by the Gibbs distribution, Kawasaki assigns a probability to the microscopic exchange process. The equation governing the evolution of the system is a master equation for the probability density of configurations  $P([S_1, \dots, S_N]; t)$ :

$$\frac{d}{dt} P([S]; t) = -P([S]; t) \sum_{[S']} P([S] \rightarrow [S']) + \sum_{[S']} P([S']; t) P([S'] \rightarrow [S]) \quad (5.15)$$

**Fig. 14.** Kawasaki exchange model basic process

The transition rate for the process  $1 \rightarrow 2$  (see Fig. 14) is taken in Kawasaki formulation as:

$$P(1 \rightarrow 2) = \frac{1}{2\tau} \left[ 1 - \tanh \left( \frac{\Delta\mathcal{H}}{2k_B T} \right) \right] \quad (5.16)$$

Where  $\Delta\mathcal{H}$  is the change in energy required to go from state 1 to state 2.  $\tau$  simply fixes the unit of time. The previous expression is chosen by mathematical simplicity, but any function of the form  $P(1 \rightarrow 2) = \exp \left( -\frac{\Delta\mathcal{H}}{k_B T} \right) \omega \left( \frac{\Delta\mathcal{H}}{k_B T} \right)$  where  $\omega(z) = \omega(z^{-1})$  will satisfy detailed balance for the master equation and hence will ensure the correct equilibrium state (see discussion in section 1.6).

The Kawasaki exchange model and the Cahn–Hilliard–Cook model exhibit many common features. In fact, it is possible to obtain, using some approximations, the Cahn–Hilliard–Cook model by means of a coarse-graining of the Kawasaki exchange model (Langer 1971; Gunton et al. 1983b). In Fig.13 we show the evolution of the CHC and Kawasaki models for the case of a quench of critical concentration. Observe the great similarity between the evolution of the two models.

### 5.1.3 Dynamical Scaling

An important property of the late stages of the dynamics of phase separation process is that of *dynamical scaling* (Binder et al. 1974; Lebowitz et al. 1982). This scaling description is suggested by the fact that systems at different times look similar provided a space rescaling is performed. More precisely: if we take a snapshot of a configuration of the system at a given time  $t_1$  and make it larger by a factor  $\lambda$  it turns out that the resulting figure is *statistically indistinguishable* of a snapshot of the same system at a later time  $t_2 = \lambda^{1/a} t_1$ ,  $a$  being a dynamical scaling exponent (see Fig.15). This dynamical scaling description is based on the physical assumption that, at the late stages of the evolution, only one length, related to the typical domain size, is relevant. Mathematically, the scaling description implies that the pair correlation function  $G(\mathbf{r}, t)$ :

$$G(\mathbf{r}, t) = \langle \Phi(\mathbf{r}, t) \Phi(\mathbf{r} + \mathbf{r}', t) \rangle, \quad (5.17)$$

or, rather, its circular average:

$$G(r, t) = \frac{\int d\Omega G(\mathbf{r}, t)}{\int d\Omega} \quad (5.18)$$

( $\Omega$  denotes the angular variables), which, in principle, is a two variable function, depends only on a combination of them, namely:

$$G(r, t) = g(r/R(t)) \quad (5.19)$$

where the *scaling function*  $g(x)$  is a one variable function of the scaling variable  $x = r/R(t)$ .  $R(t)$  is a measure of the typical domain size and can be defined, for instance, as the first zero of the pair correlation function. The time behaviour of  $R(t)$  is described by a power-law:

$$R(t) \sim t^a \quad (5.20)$$

This expresses again the idea that the system at two different times  $t_1$ ,  $t_2$  will look similar if a space rescaling of magnitude  $\lambda$  for the system at time  $t_1$  is performed such that  $\lambda R(t_1) = R(t_2)$  or, according to the previous expression,  $\lambda = (t_2/t_1)^a$ . Another important feature of the dynamical scaling description is that of the universality of the scaling function  $g(x)$  and the scaling exponent  $a$ . The value  $a = 1/3$  seems to hold for the CHC and Kawasaki models independently of dimension and the quench location. The function  $g(x)$  does depend on dimension and quench location (Chakrabarti et al. 1991; 1993), although it seems that its variation is very small for quenches close to the critical concentration (Torral et al. 1989; Fratzl et al. 1989).

Dynamical scaling is usually checked in terms of the experimentally accessible structure function  $S(\mathbf{k}, t)$ , defined as the Fourier transform of the pair correlation function  $G(\mathbf{r}, t)$ . The scaling relation for  $S(k, t)$  is obtained from the one holding for  $G(r, t)$  as:

$$S(k, t) = R(t)^d \mathcal{F}(kR(t)) \quad (5.21)$$

$\mathcal{F}(x)$  is the scaling function,  $d$  is the system dimensionality. In Figure 16 we plot the time evolution of the angularly averaged structure factor  $S(k, t)$  from a numerical solution of the CHC equation in 3 dimensions in the case of a critical quench. We can see that the structure factor starts from a rather flat shape and, as time goes on, develops a maximum whose location  $k_m$  moves towards smaller values of  $k$  and whose intensity increases. The presence of this maximum signals the phase separation process and the location of the peak is related to the inverse typical linear length of the domains. In Figure 16 we also plot  $S(k, t)R(t)^{-d}$  versus  $kR(t)$  to check the scaling relation (5.21). One concludes that scaling is well satisfied at the times studied. Similar results hold for other dimensions and models, as well as for experiments (Binder 1990).

## 5.2. Growth of rough surfaces

Many surfaces in nature grow roughly. Examples include such diverse systems as biological tumors, mountain ranges, sand piles, a fluid flowing through a porous medium and many others (Lagally 1990; Thomas 1982). To study this widespread phenomenon, scientists have developed microscopic models and field models (Family et al. 1991).

The simplest of the microscopic models is one due to Eden (Eden 1958). The Eden model can be used to study the spread of epidemics. In this model one considers a regular lattice in which a site is initially marked as “infected”. At a given time a randomly selected infected site infects one of its nearest neighbours, again chosen at random. After some time, the infected area is rather compact with a rough surface, see Fig.17. From a careful analysis of this figure one can deduce that the surface has fractal properties and that the dependence of the roughness with time and with the linear dimensions of the Eden cluster can be described by a power-law with some characteristic exponents.

**Fig. 17.** A cluster grown by the Eden model rules (see text). The right figure is a detail of the cluster boundary to show that it is a rough surface.

Another kind of models aim to mimic the evolution of a surface that grows by external flux of particles (Meakin et al. 1986). In the simplest *solid on solid* model particles are added to a regular lattice. The incident particles add to the height  $h_i$  in lattice site  $i$ . The presence of a surface tension is simulated by rejecting the addition of particles that produce a difference in the height between two nearest neighbours in the lattice larger than some fixed amount (Kim et al. 1989).

One of the most successful field theory models to study surface growth is that of Kardar, Parisi and Zhang (KPZ) (Kardar et al. 1986). In the KPZ model

**Fig. 15.** Self similarity after rescaling of configurations for the two-dimensional CHC model in the case of a critical quench  $m = 0$ . A snapshot of the configuration at time  $t_1 = 50,000$  (upper row, left) is magnified by a factor  $256/119 \approx 2.15$ . The resulting figure is very similar (from the statistical point of view) to a snapshot of the same system at time  $t_2 = t_1 \times 2.15^3 \approx 500,000$ , showing that the dynamical exponent in Eq.(5.20) is close to  $1/3$ .

**Fig. 16.** Raw data (left figure) for the structure factor for the time evolution of the three-dimensional CHC in the case of a critical quench ( $m = 0$ ). In the right figure we have rescaled the data according to Eq.(5.21) to show that dynamical scaling holds reasonably well for this system at the times studied.



one defines a field  $h(\mathbf{r}, t)$  giving the surface height at location  $\mathbf{r}$  on a  $(d - 1)$ -dimensional substrate at time  $t$ . Several physical processes contribute to the time evolution of this field.

- (i) Tendency to grow normal to the surface. A simple geometrical argument shows that in this case the evolution is given by:

$$\frac{\partial h(\mathbf{r}, t)}{\partial t} = \lambda \sqrt{1 + (\nabla h)^2} \approx \frac{\lambda}{2} (\nabla h)^2$$

to the lowest order in the gradient of the surface field. Here  $\lambda$  is the surface mean growth velocity.

- (ii) Particle diffusion on the surface.

$$\nu \nabla^2 h$$

$\nu$  is a constant related to the surface tension.

- (iii) Random variations of the incident flux. These are modelled by a stochastic, noise type term:

$$\eta(\mathbf{r}, t)$$

Adding the contribution of the three terms we get the celebrated KPZ equation:

$$\frac{\partial h(\mathbf{r}, t)}{\partial t} = \nu \nabla^2 h(\mathbf{r}, t) + \frac{\lambda}{2} (\nabla h)^2 + \eta(\mathbf{r}, t) \quad (5.22)$$

The  $\eta(\mathbf{r}, t)$  noise is assumed to be Gaussian distributed of mean zero and correlations:

$$\langle \eta(\mathbf{r}, t) \eta(\mathbf{r}', t') \rangle = 2D \delta(\mathbf{r} - \mathbf{r}') \delta(t - t') \quad (5.23)$$

Some of the parameters of this equation,  $\lambda$ ,  $\nu$ ,  $D$  are redundant. One can reparametrize the field  $h \rightarrow (\nu/2D)^{1/2} h$  and the time  $t \rightarrow \nu t$  to obtain a simpler equation:

$$\frac{\partial h(\mathbf{r}, t)}{\partial t} = \nabla^2 h(\mathbf{r}, t) + \frac{\lambda}{2} (\nabla h)^2 + \eta(\mathbf{r}, t) \quad (5.24)$$

where

$$\langle \eta(\mathbf{r}, t) \eta(\mathbf{r}', t') \rangle = \delta(\mathbf{r} - \mathbf{r}') \delta(t - t') \quad (5.25)$$

Although many exact results are known of this stochastic non-linear partial differential equation, an explicit solution is still missing. Numerical studies have played an important role in characterizing the solutions of the KPZ equation (Moser et al. 1991). In the numerical studies in a one-dimensional substrate one typically uses the following discretization of the KPZ equation:

$$\frac{dh_j(t)}{dt} = \frac{h_{j+1} + h_{j-1} - 2h_j}{a_0^2} + \frac{\lambda}{2} \left( \frac{h_{j+1} - h_{j-1}}{2a_0} \right)^2 + \eta_j(t), \quad (5.26)$$

(periodic boundary conditions are considered). The lattice spacing is usually taken equal to 1,  $a_0 = 1$ . The numerical integration of the previous set of stochastic differential equations can proceed by any of the methods explained in Chapter 3, although the Heun method is particularly well suited in this case since it allows to use much larger values for the integration step  $\Delta t$  than the one allowed

**Fig. 18.** Plot of the height field  $h(x, t)$  from a numerical simulation of the KPZ equation in one-dimensional substrate for increasing times. We can see in this figure that the surface grows rough and that the roughness increases with time. The right figure shows a detail of the surface for the latest time.

in the Euler method (Forrest et al. 1993). If we write the previous equations in the general vector form:

$$\frac{\partial \mathbf{h}(t)}{\partial t} = \mathbf{F}(\mathbf{h}(t)) + \boldsymbol{\eta}(t), \quad (5.27)$$

The Heun method uses the following recurrence relation (see Eqs.(3.39)):

$$\begin{aligned} \mathbf{g}_1 &= \mathbf{F}(\mathbf{h}(t)) \\ \mathbf{g}_2 &= \mathbf{F}(\mathbf{h}(t) + \Delta t \mathbf{g}_1 + \sqrt{\Delta t} \mathbf{u}) \\ \mathbf{h}(t + \Delta t) &= \mathbf{h}(t) + \frac{\Delta t}{2} (\mathbf{g}_1 + \mathbf{g}_2) + \sqrt{\Delta t} \mathbf{u} \end{aligned} \quad (5.28)$$

Where  $\mathbf{u} = (u_1, \dots, u_N)$  are independent Gaussian variables of mean zero and variance unity.

In Figure 18 we can see the evolution of the field for a 1- $d$  substrate of length  $L$ . It is clear from this figure that the surface grows rough, similarly to the case of the microscopic Eden model. A quantitative measure of the surface roughness is given by the quantity  $w(t, L)$

$$w(t, L) = \sqrt{\langle \bar{h}^2 - \bar{h}^2 \rangle} \quad (5.29)$$

Where the bar denotes a spatial average  $\bar{h} = L^{-1} \sum_{i=1}^L h_i$  and the angular brackets denote an average over initial conditions and realizations of the noise.

It is important to characterize the time evolution of  $w(t, L)$ . The self-similar character of the surface implies that the late-time behaviour for a system of infinite size must be given by a power-law relation of the form:

$$w(t, \infty) \sim t^\beta \quad (5.30)$$

This relation has been well verified in computer simulations and the exact value for  $\beta$  can be obtained theoretically for 1-d systems ( $\beta = 1/3$ , see Family et al. 1991). The reader could think that the verification of this simple law should be an easy task. Not at all! In the rest of these notes, we just want to point out which are the main difficulties in trying to verify directly the, innocent looking, power-law given by Eq.(5.30).

The first and most obvious difficulty comes from the fact that solving numerically a set of stochastic differential equations is *always* difficult. One has to average over a large number of realizations to have small statistical errors and also make sure that results do not depend on the time step chosen for the numerical integration.

The second problem is that expression (5.30) is an **asymptotic** result, i.e. strictly valid only in the limit  $t \rightarrow \infty$ . However, earlier time regimes can also be described by a relation of the same functional form with an *effective* exponent  $\beta_{eff} \neq \beta$ . For example, at very early times, the noise term dominates (remember that it gives a contribution of order  $\sqrt{\Delta t}$ ) and the roughness grows as the square root of time,  $w(t, L) \sim t^{1/2}$ . For intermediate times, the linear term dominates. It is not difficult to solve this linear case ( $\lambda = 0$ , the so-called Edwards-Wilkinson model (Edwards et al. 1982)) and show that the roughness grows as  $w(t) \sim t^{1/4}$ . It is only at the very late stages that the effect of the non-linearity fully develops and one obtains the true asymptotic regime  $w(t) \sim t^{1/3}$ . But it is very difficult to be sure that one is in the asymptotic regime and that the exponent obtained from a, say, log-log plot of  $w(t)$  versus  $t$  is giving indeed the true asymptotic exponent instead of an asymptotic exponent corresponding to smaller times.

Another problem concerns finite size effects. The result (5.30) is valid only in the limit of an infinite system. If the system size  $L$  is finite (as it is bound to be in a numerical simulation), the surface roughness can saturate to an asymptotic value  $w(t \rightarrow \infty, L) \sim L^\zeta$  ( $\zeta = 3/2$  in 1-d) and one might never enter the time regime in which relation (5.30) holds. The dramatic effect of the finite size effects can be shown in detail for the linear case,  $\lambda = 0$ . In this case, the solution can be found as:

$$w_{(0)}^2(t, L) = \frac{1}{L} \sum_{k=1}^{L-1} \frac{1 - \exp(-8 \sin^2(\pi k/L)t)}{4 \sin^2(\pi k/L)}. \quad (5.31)$$

The strong  $L$  dependence of this linear solution can be seen in Fig. [19]. Looking at the figure, we can say that only for  $L \gtrsim 10^4$  there is a relative error  $w_{(0)} < 1\%$  for  $t \simeq 6000$ . The moral of the story is that if one wants to avoid finite size effects one has to take systems at least as large as the ones needed in the linear case, as it is sustained by a perturbative solution of the KPZ equation (Forrest et al. 1993).

**Fig. 19.** Dependence with  $L$  of the square of the surface roughness,  $w_{(0)}^2(t, L)$  for the linear solution of the KPZ equation, Eq.(5.31).

In fact, the finite size effects show up in a very determined way. It is now well established (Family et al. 1985; Jullien et al. 1985) that the roughness  $w(L, t)$ , for a finite substrate length  $L$ , follows an asymptotic scaling description in terms of both time  $t$  and  $L$ :

$$w(t, L) = t^\beta F(tL^{-\zeta/\beta}) \quad (5.32)$$

( $\zeta$  is the so-called *roughness exponent*). In principle, this relation can also be used to compute the exponent  $\beta$ . Again, the difficulty is that this relation is also valid for the linear case,  $\lambda = 0$  *with different values for the exponents*, namely  $w(t, L) = t^{1/4}F(tL^{-2})$  and one has to be careful enough to ensure that the non-linearity has fully developed. It is not strange, then, that the most powerful available computers have been used in this problem of computing precisely the exponents of the KPZ equation.

## Acknowledgments

Financial support from the DGICYT, project number PB-92-0046 is acknowledged.

## References

- Abramowitz, M. and Stegun, I.A. (1972): *Handbook of Mathematical Functions*, Dover Publications.
- Ahrens, J.H., Dieter, U. (1972): *Comm. of the ACM*, **15** 873; (1988): **31** 1331.
- Allen, M.P., Tildesley, D.J. (1987): *Computer Simulation of Liquids*, Clarendon Press.
- Amit, D.J. (1984): *Field Theory, the Renormalization Group and Critical Phenomena*, World Scientific.

- Barber, M. (1983): *Phase Transitions and Critical Phenomena*, vol. 8, edited by C. Domb and J. Lebowitz, Academic Press.
- Batrouni, G.G., Katz, G.R., Kronfeld, A.S., Lepage, G.P., Svetitsky, B., Wilson, K.G. (1985): *Phys. Rev.* **D 32** 2736.
- Binder, K., and Stauffer, D. (1974): *Phys. Rev. Lett.* **33**, 1006.
- Binder, K. (1990): *Materials Science and Technology, Vol. 5: Phase Transformation in Materials*, ed. P. Haasen, VCH (Weinheim, Germany), p-405.
- Binder, K. and Heermann, D.W. (1988): *Monte Carlo simulations in Statistical Physics*, Springer, Berlin, Heidelberg.
- Binder, K. (ed) (1992): *The Monte Carlo Method in Condensed Matter Physics*, Topics App. Phys. Vol 71 (Springer, Berlin, Heidelberg).
- Box, G.E.P. and Muller, M.E. (1958): *Ann. Math. Statist.* **29**, 610.
- Bruce, A.D. (1980): *Adv. Phys.* **29**, 111.
- Bruce, A.D. and Cowley, R. (1980): *Adv. Phys.* **29**, 219.
- Bruce, A.D. (1985): *J. Phys.* **A 18**, L873.
- Cahn, J.W. and Hilliard, J. E., (1958): *J. Chem. Phys.* **28**, 258.
- Cardy, J.L., editor (1988): *Finite Size Scaling*, North-Holland.
- Chakrabarti, A., (1992): *Phys. Rev. B* **45**, 9620.
- Chakrabarti, A., Toral, R. and Gunton, J.D., (1991): *Phys. Rev. A* **44**, 12133; (1993): *Phys. Rev. E* **47**, 3025.
- Cook, H. (1970): *Acta Metall.* **18** 297 (1970).
- Cowley, R. (1980): *Adv. Phys.* **29**, 1.
- Duane, S., Kennedy, A.D., Pendleton, B.J., and Roweth, D. (1987): *Phys. Lett.* **B 195** 216.
- Eden, M. (1958): *Symposium on Information Theory in Biology*, edited by H. P. Yockey (Pergamon Press, New York), p.359.
- Edwards, S.F. and Wilkinson, D.R. (1982): *Proc. R. Soc. Lon. A* **381** 17.
- Faires, J. and Burden, R. (1993): *Numerical Methods*, PWS Publishing Company, Boston.
- Family, F. and Vicsek, T. (1985): *J. Phys. A* **18**, L75.
- Recent reviews on kinetic roughening in surface growth models can be found in: F. Family and T. Vicsek, eds., *Dynamics of Fractal Surfaces*, (World Scientific, Singapore, 1991); J. Krug and H. Spohn, in *Solids Far From Equilibrium: Growth, Morphology and Defects*, edited by C. Godrèche, (Cambridge Univ. Press, Cambridge, England, 1991); L. Sander, *ibid*; T. Vicsek, *Fractal growth phenomena*, (World Scientific, Singapore, 1989).
- Feller, W. (1971): *An introduction to probability theory and its applications*, vols. 1,2. John Wiley & Sons.
- Ferrenberg, A.M. and Swendsen, R.H. (1989): *Comp. in Phys.* **Sept/Oct**, 101.
- Ferreira, A.L. and Toral, R. (1993): *Phys. Rev.* **E 47** R3848.
- Forrest, B.M. and Toral, R. (1993): *J. Stat. Phys.* **70**, 703.
- Forrest, B.M. and Toral, R. (1994): *J. Stat. Phys.* **77**, 473.
- Fratzl, P. and Lebowitz, J.L., (1989): *Acta Metall.* **37**, 3245; Fratzl, P. (1991): *J. Appl. Cryst.* **24**, 593.
- Gardiner, C.W. (1985): *Handbook of Stochastic Methods*, Springer-Verlag, Berlin.
- Gard, T.C. (1987): "Introduction to Stochastic Differential Equations", Marcel Dekker inc, vol 114 of "Monographs and Textbooks in Pure and Applied Mathematics".
- Glauber, R.J. (1963): *J. Math. Phys.* **4**, 294.
- Grimmett, G.R. and Stirzaker, D.R. (1982): *Probability and Random Processes*, Oxford U. Press.

- Grant, M., San Miguel, M., Viñals, J. and Gunton, J.D. (1985): Phys. Rev. **B 31**, 3027.
- Greiner, A., Strittmatter, W. and Honerkamp, J. (1988): J. Stat. Phys. **51**, 95.
- Gunton, J.D., San Miguel, M. and Sahni, P.S. (1983): in *Phase Transitions and Critical Phenomena*, Vol. 8, edited by C. Domb and J. L. Lebowitz (Academic, London).
- Gunton, J.D. and Droz, M. (1983): *Introduction of the Theory of Metastable and Unstable States*, Springer-Verlag.
- Gupta, S. (1992): Nuc. Phys. **B 370** 741.
- Heermann 1986: *Computer Simulation Methods*, Springer-Verlag.
- Hohenberg, P.C. and Halperin, B, (1977): Rev. Mod. Phys. **49**, 435.
- Irback, A. (1994): J. Chem.Phys. **101** 1661.
- James, F. (1990): Comp. Phis. Comm. **60**, 329.
- Jullien, R. and Botet, R. (1985): J. Phys. A **18**, 2279.
- Kawasaki, K. (1972): in *Phase Transitions and Critical Phenomena*, vol. 2, edited by C. Domb and M.S.Green, Academic Press.
- Kalos, M.H. and Whitlock, P.A. (1986): *Monte Carlo Methods*, John Wiley and Sons.
- Kardar, M., Parisi, G. and Zhang, Y.C. (1986): Phys. Rev. Lett. **56**, 889.
- Kennedy A.D. and Pendleton B. (1991), Nucl.Phys. B (Proc. Suppl). **20**, 118.
- Kim, J.M. and Kosterlitz, J.M. (1989): Phys. Rev. Lett. **62**, 2289.
- Kloeden, P.E. and Platen, E. (1992):*Numerical Solution of Stochastic Differential Equations*, Springer-Verlag.
- Knuth, D.E. (1981): "The Art of Computer Programming, vol.2 Seminumerical Algorithms", Addison-Wesley Pub. Co.
- Lacasta, A.M., Sancho, J.M., Hernández-Machado, A. and Toral, R. (1992): Phys. Rev. **B 45**, 5276; (1993): Phys. Rev. **B 48**, 6854.
- Langer, J.S. (1971): Ann.Phys. **65**, 53.
- Langer, J.S., Bar-on, B. and Miller, H.D. (1975): Phys. Rev. **A 11**, 1417.
- Landau, L.D. and Lifshitz, E.M. (1980):*Statistical Physics*, Pergamon Press.
- Lagally M. (ed.) (1990):*Kinetics of Ordering and Growth at Surfaces*, (Plenum, New York), and references therein.
- Lebowitz, J.L., Marro, J. and Kalos, M.H. (1982): Act. Metall. **30**, 290 and references therein.
- Meakin, P., Ramanlal, P., Sander, L.M., Ball, R.C. (1986): Phys. Rev. **A34**, 5091.
- Mehlig, B., Ferreira, A.L. and Heermann, D.W. (1992a): Phys. Lett. **B291** 151.
- Mehlig, B., Heermann, D.W. and Forrest, B.M. (1992b): Phys. Rev. **B 45** 679.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E. (1953): J. Chem. Phys. **21**, 917.
- Milshtein, G.N. (1974): Theory. Prob. Appl. **19**, 557; (1978): *ibid* **23**, 396.
- Milchev, A., Heermann, D. and Binder, K. (1986): J. Stat. Phys. **44**, 749.
- Moser, K., Kertész, J., and Wolf, D.E. (1991): Physica A **178**, 215.
- Pathria, R.K. (1972):*Statistical Mechanics*, Pergamon Press.
- Parisi, G. (1984): in *Progress in Gauge Field Theory* (Cargese lectures), ed. G't Hooft et al., Plenum Press (reproduced in Parisi, G. (1992), *Field theory, disorder and simulations*, World Scientific.
- Parisi, G. (1988):*Statistical Field Theory*, Addison Wesley.
- Press, W.H., Flannery, B.P., Teulolsky, S.A. and Vetterling, W. (1986): "Numerical Recipes". Cambridge University Press.
- Rubinstein, R.Y. (1981): *Simulation and the Monte Carlo method*, John Wiley & Sons.
- Sexton, J.C., Weingarten, D.W. (1992): Nuc. Phys. **B380** 665.
- Sokal, A.D. (1991): Nuclear Physics B (Proc. Suppl.) **20** 55.
- Thomas, T.R. (1982):*Rough Surfaces* (Longman, London).

- Toral, R., Chakrabarti, A. (1990): Phys. Rev. **B 42**, 2445.
- Toral, R., Chakrabarti, A. and Gunton, J.D. (1989): Phys. Rev. B **39**, 901; (1992): Phys. Rev. A **45**, R2147.
- Toral, R., Chakrabarti, A. (1993): Comp. Phys. Comm. **74**, 327.
- Toral, R. and Ferreira, A.L. (1994) *Proceeding fo the Conference Physics Computing, '94*, p. 265, R.Gruber, M. Tomasini, editors.
- Valls, O.T. and Mazenko, G.F. (1986): Phys. Rev. **B 34**, 7941.
- van Kampen (1981): *Stochastic Processes in Physics and Chemistry*, North-Holland.
- Zinn-Justin, J. (1989): *Quantum Field Theory and Critical phenomena*, Oxford. U. Press.

This book was processed by the author using the T<sub>E</sub>X macro package from Springer-Verlag.