13
Generation of n-dimensional correlated Gaussian variables

We have already explained in section 3.5 a method valid to generate a set of random variables $x = (x_1, x_2, \ldots, x_n)$ with a joint Gaussian distribution given by (1.80) with mean values $\mu = (\mu_1, \ldots, \mu_n)$ and correlation matrix $C_{ij} = \langle(\hat{\mathbf{x}}_i - \mu_i)(\hat{\mathbf{x}}_j - \mu_j)\rangle$ with $C = A^{-1}$. An equivalent way of looking at this problem is via the diagonalization of the quadratic form in the exponential of (1.80). This means to use the matrix relation $A = \Phi^\intercal D \Phi$, being $\Phi^\intercal$ the transpose matrix of Φ, the matrix of change of variables, and D a diagonal matrix with diagonal elements (the eigenvalues) $(\lambda_1, \lambda_2, \ldots, \lambda_n)$. As A is a symmetric matrix, this diagonalization is always possible and, as the quadratic form is supposed to be positive definite, the eigenvalues are strictly positive, $\lambda_i > 0, \forall i$. Furthermore, the determinant of the matrix of the change of variables is $|\Phi| = 1$ and $|A| = \prod_{i=1}^n \lambda_i$. The change of variables $x = \mu + \Phi z$ or $x_i = \mu_i + \sum_{j=1}^n \Phi_{ij} z_j$, in coordinates, changes the quadratic form to $\exp\left[-\frac{1}{2} z^\intercal D z\right] = \exp\left[-\frac{1}{2}\sum_{i=1}^n \lambda_i z_i^2\right]$. As the Jacobian of this change is $|\Phi| = 1$, it leads to a pdf for the z variables:

$$f_{\hat{\mathbf{z}}_1,\ldots,\hat{\mathbf{z}}_n}(z_1, z_2, ..., z_n) = \sqrt{\frac{\prod_{i=1}^n \lambda_i}{(2\pi)^n}} \exp\left[-\frac{1}{2}\sum_{i=1}^n \lambda_i z_i^2\right] \tag{13.1}$$

$$= \prod_{i=1}^n \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{z_i^2}{2\sigma_i^2}\right], \tag{13.2}$$

indicating that the z_i's are independent Gaussian variables of zero mean and variance $\sigma_i^2 \equiv 1/\lambda_i$. Once the set of z_i's has been generated by our favorite Gaussian number generator we just need to change back variables to the x_i's. In general, this algorithm is slow (it requires to solve a full matrix diagonalization problem, finding the eigenvalues and the eigenvectors) but it can be an alternative to the one explained in section 3.5. It turns out that there are some cases of interest where the change of variables adopts a particularly simple form and the whole algorithm can be speeded up with the help of the fast Fourier transform. We now have a look at two of these cases.

The free model

Let us here consider the so-called "1-d free model" in which the Gaussian joint pdf

has the particular expression:

$$f_{\hat{\mathbf{x}}_1,\ldots,\hat{\mathbf{x}}_n}(x_1,\ldots,x_n) = C\exp[-\mathcal{L}_0(x_1,\ldots,x_n)], \tag{13.3}$$

where the quadratic form given by the function $\mathcal{L}_0$ is[1)]:

$$\mathcal{L}_0(x_1,\ldots,x_n) = \frac{1}{2}\sum_{i=1}^{n}\left[(x_{i+1}-x_i)^2 + ax_i^2\right], \quad a>0, \tag{13.4}$$

and C is the normalization constant. Here we consider what are called "periodic boundary conditions" that have been discussed previously, i.e. whenever x_{n+1} appears in any formula, it should be replaced by x_1. If we write the quadratic form as $\mathcal{L}_0 = \frac{1}{2}x^\intercal Ax$, matrix A is:

$$A = \begin{pmatrix} 2+a & -1 & 0 & 0 & \cdots & 0 & -1 \\ -1 & 2+a & -1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 2+a & -1 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & a+2 & -1 \\ -1 & 0 & 0 & 0 & \cdots & -1 & a+2 \end{pmatrix} \tag{13.5}$$

with diagonal elements $A_{ii} = a+2$, off-diagonal $A_{i-1,i} = A_{i,i-1} = -1$ and the corner elements $A_{1,n} = A_{n,1} = -1$.

We make now a change of variables based on the discrete Fourier transform of the variables[2)]. Let us define, then, a set of complex variables $\hat{x}_0,\ldots,\hat{x}_{n-1}$ as:

$$\hat{x}_k = \sum_{j=0}^{n-1} e^{\frac{2\pi \mathbf{i}}{n}jk}x_{j+1}, \qquad k=0,1,\ldots,n-1. \tag{13.6}$$

Here $\mathbf{i}$ stands for the imaginary unit $\mathbf{i}=\sqrt{-1}$. The inverse transformation is:

$$x_{j+1} = \frac{1}{n}\sum_{k=0}^{n-1} e^{-\frac{2\pi \mathbf{i}}{n}jk}\hat{x}_k, \qquad j=0,\ldots,n-1. \tag{13.7}$$

Note that the definition (13.6) can be used for whatever value of $k\in\mathbb{Z}$ but as $\hat{x}_{n+k}=\hat{x}_k$, only values of $\hat{x}_k$ between $k=0$ and $k=n-1$ are, in general, independent of each other. In the particular case that the x_j's are real variables, it follows the additional condition $\hat{x}_{-k} = \hat{x}_{n-k} = \hat{x}_k^*$. This reduces the number of independent values of $\hat{x}_k$ even further. The analysis of the independent set of $\hat{x}_k$ is slightly different for n even or n odd. In what follows, we consider only the case that n is an even number and leave the reader to redo the details for the case of n odd. For instance, if $n=8$, the following relations apply: $\hat{x}_7 = \hat{x}_1^*$, $\hat{x}_6 = \hat{x}_2^*$, $\hat{x}_5 = \hat{x}_3^*$ as well as $\hat{x}_0 =$

1) In some contexts, this function is called a "free Lagrangian".

2) The reader not familiar with the Fourier transform might have a look at appendix 18. For convenience, the definitions used in this section are slightly different from those of the appendix, where the numbers x_i run from $i=0$ to $i=N-1$, whereas in the notation used here they run from $i=1$ to $i=N$.

$\hat{x}_0^*$, $\hat{x}_4 = \hat{x}_4^*$. These relations imply that if we write $\hat{x}_k = r_k + is_k$, then $s_0 = s_4 = 0$ and the only independent variables are $r_0, r_1, s_1, r_2, s_2, r_3, s_3, r_4$. For compactness, we will use the notation $(z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8) \equiv (r_0, r_4, r_1, s_1, r_2, s_2, r_3, s_3)$ and a similar generalization for arbitrary n even.[3)]

The nice feature of the change of variables (13.6-13.7) is that it diagonalizes the quadratic form:

$$\sum_{i=1}^{n} \left[(x_{i+1} - x_i)^2 + a x_i^2 \right] = \frac{1}{n} \sum_{k=0}^{n-1} \omega_k^2 |\hat{x}_k|^2, \qquad \omega_k^2 = a + 4 \sin^2 \left(\frac{\pi k}{n} \right). \quad (13.8)$$

To prove this identity it is useful the relation

$$\frac{1}{n} \sum_{i=1}^{n} \mathrm{e}^{\frac{2\pi \mathrm{i}}{n} ik} = \begin{cases} 1, & k = 0, \\ 0, & k \neq 0. \end{cases} \quad (13.9)$$

However, we must not look at $\mathcal{L}_0$ written in terms of $\hat{x}_k$ and jump to the conclusion that the full set of the $\hat{x}_k$'s are independent Gaussian variables, because we just proved that there are relations between them and some of these variables depend on the others. We need to rewrite the quadratic form $\mathcal{L}_0$ using only the set of independent variables $z_k, k = 1, \ldots, n$. Including only these variables and using $\omega_{n-k} = \omega_k$, the quadratic form can be written as[4)]:

$$\begin{aligned} \frac{1}{n} \sum_{k=0}^{n-1} \omega_k^2 |\hat{x}_k|^2 &= \frac{1}{n} \left(\omega_0^2 |\hat{x}_0|^2 + \omega_{\frac{n}{2}}^2 r_{\frac{n}{2}}^2 + 2 \sum_{k=1}^{\frac{n}{2}-1} \omega_k^2 |\hat{x}_k|^2 \right) \\ &= \sum_{k=1}^{n} \Lambda_k z_k^2, \end{aligned} \quad (13.10)$$

with $\Lambda_1 = \omega_0^2/n, \Lambda_2 = \omega_{\frac{n}{2}}^2/n, \Lambda_3 = \Lambda_4 = 2\omega_1^2/n, \Lambda_5 = \Lambda_6 = 2\omega_2^2$, etc. As $f_{\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_n}(z_1, \ldots, z_n) \propto e^{-\frac{1}{2} \sum \Lambda_k z_k^2}$, this shows that the z_k's are independent Gaussian variables of zero mean and variance $\sigma_k^2 = 1/\Lambda_k$. Once these variables have been generated, we use the discrete Fourier transform (13.7) to obtain the original variables $(x_1, \ldots, x_n)$.

To sum up, to generate n Gaussian variables $(x_1, \ldots, x_n)$ whose pdf is (13.3) we generate a set of independent Gaussian variables $(z_1, \ldots, z_n)$ of zero mean and variance $\sigma_k^2 = 1/\Lambda_k$. From these variables, we construct the Fourier variables $(\hat{x}_1, \ldots, \hat{x}_n)$ fulfilling the symmetry relations $\hat{x}_{-k} = \hat{x}_{n-k} = \hat{x}_k^*$. Finally, using an inverse discrete Fourier transform, we obtain the desired set $(x_1, \ldots, x_n)$.

In practice, most fast Fourier transform routines organize their variables such that, for the discrete Fourier transform of a set of real variables x_i, only the independent set of values z_k are kept in memory, precisely in the same order that have been defined

3) For $n = 9$, an odd number, the relations between the Fourier variables is: $\hat{x}_8 = \hat{x}_1^*$, $\hat{x}_7 = \hat{x}_2^*$, $\hat{x}_6 = \hat{x}_3^*$, $\hat{x}_5 = \hat{x}_4^*$ and $\hat{x}_0 = \hat{x}_0^*$. The set of independent Fourier variables is $(z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9) = (r_0, r_1, s_1, r_2, s_2, r_3, s_3, r_4, s_4)$.

4) The reader can check this formula using $n = 8$, for example.

here. For example, if $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ is the original vector of $n = 8$ real variables, the Fourier coefficients $\hat{x}_k = r_k + is_k$, are kept in a vector organized precisely as $(r_0, r_4, r_1, s_1, r_2, s_2, r_3, s_3)$[5]. The next program listing implements this algorithm .

```
program freeGaussian
implicit double precision (a-h,o-z)
parameter (n=256)
dimension x(n)
open(66,file='freeGaussian.dat',status='unknown')
a=1.0d0
pin=3.14159265358979d0/n
   x(1)=ran_g()*dsqrt(n/a)
   x(2)=ran_g()*dsqrt(n/(a+4.0d0))
   do i=2,n/2
     sigma=dsqrt(n/(2.0d0*(a+4.0d0*(dsin(pin*(i-1)))**2)))
     x(2*i-1)=sigma*ran_g()
     x(2*i)=  sigma*ran_g()
   enddo
   call realfft1d(x,n,-1)
do i=1,n
  write(66,*) x(i)
enddo
end program freeGaussian
```

Note that in our notation `x(2)` contains really $r_{n/2}$ or z_2. We have used the values $\Lambda_1 = \omega_0^2/n = a/n$, $\Lambda_2 = \omega_{\frac{n}{2}}^2/n = (a+4)/n$. We include the call to `realfft1d(x,n,-1)`, a generic name for a routine which provides the inverse discrete Fourier transform in the case that the variables are real numbers[6].

Translational invariance

The second case[7] in which the use of discrete Fourier transform can be useful is when the correlation function of the random variables $(x_1, \ldots, x_n)$ depends only on the absolute value of the difference between the indexes, i.e. when $\langle (x_i - \mu_i)(x_j - \mu_j)\rangle = C_{ij} = C_{|i-j|}$, $i, j = 1, n$. As we will see, the method we are going to explain yields a correlation function that satisfies, beyond the general condition $C_\ell = C_{-\ell}$, the so-called periodic boundary conditions, namely: $C_\ell = C_{n+\ell} = C_{n-\ell}$. These boundary conditions appear naturally if we consider that the $(x_1, \ldots, x_n)$ variables are placed in a ring such that x_1 and x_n are neighbors of each other, see figure 13.1. In this setup, and considering $n = 8$ for the sake of clarity, it is clear that the correlation between the x_1 and x_3 variables is the same whether we consider that the distance separating them is $\ell = 3 - 1 = 2$ if computed counter-clockwise from x_1 to x_3, or $\ell = 8 + 1 - 3 = 6$, computed clockwise, hence it is natural to assume that

5) Or, in general, organized as $(r_0, r_{n/2}, r_1, s_1, r_2, s_2, \ldots, r_{n/2-1}, s_{n/2-1})$

6) Remember to read the exact definition of the discrete Fourier transform used by the numerical fast Fourier transform package in order to correct by the right factors of n if needed, see appendix 18.

7) Actually, this second case includes the first one as a particular example.

$C_2 = C_{8-2} = C_6$. If it is not reasonable to assume that our set of random variables $(x_1, \ldots, x_n)$ fulfill the periodic boundary conditions, there is a simple trick we can use. We first double the number of variables and consider $(x_1, \ldots, x_{2n})$, which then we generate assuming periodic boundary conditions. Finally, simply discard all variables $(x_{n+1}, \ldots, x_{2n})$, leaving us with the required set $(x_1, \ldots, x_n)$.[8)]

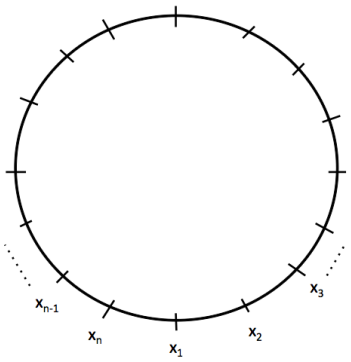

Figure 13.1 Graphical representation of the periodic boundary conditions for the set of variables $(x_1, \ldots, x_{2n})$.

Without loss of generality, we assume $\mu_i = 0$ and hence $C_{|i-j|} = \langle x_i x_j \rangle$. When $\mu_i \neq 0$ all we need to do is to add μ_i to the generated value of the random variable x_i. Let $\hat{x}_k$ be the discrete Fourier transform of x_i as defined in (13.6). As x_j are real numbers, it is $\hat{x}_k^* = \hat{x}_{-k}$, from where

$$\begin{aligned} \langle |\hat{x}_k|^2 \rangle &= \langle \hat{x}_k \hat{x}_{-k} \rangle = \sum_{i,j=0}^{n-1} e^{\frac{2\pi \mathrm{i}}{n}(i-j)k} \langle x_{i+1} x_{j+1} \rangle \\ &= \sum_{i,j=0}^{n-1} e^{\frac{2\pi \mathrm{i}}{n}(i-j)k} C_{|i-j|}. \end{aligned} \tag{13.11}$$

If we now make the change of variables $\ell = i - j$ and use the periodic boundary conditions $C_\ell = C_{n+\ell}$ we arrive at:

$$\frac{1}{n}\langle |\hat{x}_k|^2 \rangle = \sum_{\ell=0}^{n-1} e^{\frac{2\pi \mathrm{i}}{n}\ell k} C_\ell \equiv S_k. \tag{13.12}$$

We have defined S_k, the discrete Fourier transform of the correlation function C_ℓ. This function is called, in some contexts, the "structure factor" or the "power spectrum" in others. It is obvious from its definition that it is a real, positive defined

8) The variables $(x_{n+1}, \ldots, x_{2n})$ also satisfy the required correlations, but they are not independent of the set $(x_1, \ldots, x_n)$.

function. Moreover, it satisfies $S_{-k} = S_k$and $S_{k+n} = S_k$. It is possible to invert the discrete Fourier transform to find C_ℓ from S_k:

$$C_\ell = \frac{1}{n}\sum_{k=0}^{n-1} e^{-\frac{2\pi \mathrm{i}}{n}\ell k} S_k, \qquad \ell = 0, \ldots, n-1. \tag{13.13}$$

We now introduce a set of independent Gaussian variables $(z_1, \ldots, z_n)$ of zero mean and variance 1, $\langle z_i z_j \rangle = \delta_{ij}$. It is possible to show that if $\hat{z}_k$ is the discrete Fourier transform, defined as in (13.6), the corresponding structure factor turns out to be $\langle |\hat{z}_k|^2 \rangle = n$. Therefore, if we define $\hat{x}_k = S_k^{1/2}\hat{z}_k$, it is clear that $\frac{1}{n}\langle |\hat{x}_k|^2 \rangle = S(k)$, the desired relation (13.12) in Fourier space. All that remains now is to use the inverse discrete Fourier transform (13.7) to obtain $(x_1, \ldots, x_n)$.

In summary, to generate n values $(x_1, \ldots, x_n)$ of Gaussian random variables with a given correlation function C_ℓ, follow the next steps:
(0) Compute the discrete Fourier transform S_k of the correlation function C_ℓ (implemented using periodic boundary conditions).
(1) Generate a set of independent Gaussian variables $(z_1, \ldots, z_n)$ of zero mean and variance 1.
(2) Compute its discrete Fourier transform $\hat{z}_k = \mathcal{F}_D(z)$ and obtain $\hat{x}_k = S_k^{1/2}\hat{z}_k$.
(3) Compute the inverse discrete Fourier transform to obtain $x = \mathcal{F}_D^{-1}(\hat{x})$.
(4) Add the average value μ_i to x_i, if needed.

This method is known in the literature as the Fourier filtering method[64]. A final word of warning is necessary here. Many times one introduces an *ad hoc* correlation function C_ℓ. For instance, a power-law type, $C_\ell \sim |\ell|^{-\gamma}$. To use the previous algorithm, we need precise values for C_ℓ for $\ell = 0, \ldots, n-1$ and the fulfillment of the periodic boundary conditions $C_\ell = C_{n-\ell}$. Therefore, it is important to be more precise about the exact meaning of "$C_\ell \sim |\ell|^{-\gamma}$". For example, we could define (we assume n even):

$$C_\ell = \begin{cases} C_0, & \ell = 0, \\ |\ell|^{-\gamma}, & -n/2 \leq \ell \leq n/2, \ell \neq 0, \end{cases} \tag{13.14}$$

with a given value for C_0, supplemented with the periodic boundary conditions $C_{n-\ell} = C_{-\ell} = C_\ell$, whenever needed. One has to be careful, though, with the value for C_0. If one computes, in general numerically using (13.12), the structure factor S_k corresponding to this definition, one will notice that it can become negative for some range values of k, depending on the value of C_0. This means that (13.14) is such that the quadratic form $\sum_{i,j} x_i A_{i,j} x_j$ with $(A^{-1})_{i,j} = C_{|i-j|}$, is not positive definite and, therefore, **there does not exist a set of Gaussian random variables whose correlation function is (13.14) with this particular choice for** C_0 .

This example shows that the first thing we have to do when confronted with a "reasonable" correlation function C_ℓ is to check whether the associated quadratic form $\sum_{i,j} x_i A_{i,j} x_j$ is positive definite. This is done by checking that all elements of the discrete Fourier transform S_k are positive. Would that not be the case, one can either discard the given C_ℓ or "fix it". The simplest way of fixing it is by finding the minimum (negative) value $S_{\min} = \min_k S_k$, and then replacing $S_k \to S_k - S_{\min}$. This,

of course, ensures that $S_k \geq 0, \forall k$ and, at the level of the correlation function, all this procedure does is to replace C_0 by a new value while letting unchanged all other values of $C_\ell, \ell \neq 0$. We call this the "minimal subtraction procedure".

One could use instead the following expression for the correlation function:

$$C_\ell = (1+\ell^2)^{-\gamma/2}, \qquad -n/2 \leq \ell \leq n/2 \tag{13.15}$$

supplemented, again, with the periodic boundary conditions $C_{n-\ell} = C_{-\ell} = C_\ell$, whenever needed. In this case the resulting S_k is always positive and we do not need to worry about any modifications.[9] We now present a program listing that implements this algorithm for the generation of Gaussian random numbers with a power-law correlation function of the form (13.15).

```
program C1d
implicit double precision (a-h,o-z)
parameter(N=64)
double complex s(0:N-1), x(0:N-1)
cc(i)=1.0d0/(1.0d0+i**2)**3.0d0
s(0)=dcmplx(cc(0),0.0d0)
do i=1,N/2
  s(i)=dcmplx(cc(i),0.0d0)
  s(N-i)=s(i)
enddo
call fft1d(s,N,1)
ss=0.0d0
do i=0,N-1
  ss=min(ss,real(s(i)))
enddo
if (ss < 0.0d0) stop 'Non-positive quadratic form'
s=dsqrt(real(s))
!Actual generation begins here
```

```
do i=0,N-1
  x(i)=ran_g()
enddo
call fft1d(x,N,1)
x=x*s
call fft1d(x,N,-1)
```

```
end program C1d
```

Note that the first lines up to the commented line are needed (i) to check that the correlation value is acceptable as it yields a positive definite quadratic form and (ii) to generate the discrete Fourier transform S_k and its square root $S_k^{1/2}$ from the given correlation function C_i defined as `cc(i)`. These lines need to be ran only once. The actual generation of the x_i's starts by the assignation of Gaussian values to the components `z(i)`. There are some symmetries that can be used to simplify this program using that S_k is a actually a real number. Note that the desired random numbers

9) This is true if $\gamma > 0$. Negative values of γ and small values of n could again yield negative S_k and require again the use of the minimal subtraction procedure.

$x_1, \ldots, x_n$ are stored in a complex vector `x(0:N-1)` but, in fact, the imaginary part of any component `x(i)` is equal to zero[10].

There are other ways in which a power-law-type $C_\ell \sim |\ell|^{-\gamma}$ can be accomplished. A possibility is to take as a starting point directly S_k instead of C_ℓ using some (reasonable) criterion. For instance, take (13.15) not as the real correlation function, but as an starting point and define the structure factor suggested by the manipulation

$$S_k = \sum_{\ell=-\frac{n}{2}+1}^{\frac{n}{2}} e^{\frac{2\pi \mathrm{i}}{n}\ell k} C_\ell \approx \int_{-\infty}^{\infty} d\ell\, e^{\mathrm{i}q\ell} C_\ell = \int_{-\infty}^{\infty} d\ell\, \frac{e^{\mathrm{i}q\ell}}{(1+\ell^2)^{\gamma/2}} \tag{13.16}$$

$$= \frac{2\pi^{1/2}\left(\frac{|q|}{2}\right)^{\frac{\gamma-1}{2}} K_{\frac{\gamma-1}{2}}(|q|)}{\Gamma\left(\frac{\gamma}{2}\right)}, \tag{13.17}$$

with $q \equiv \frac{2\pi}{n}k$ and $K_\nu(z)$ is the modified Bessel function of the second kind. We then **define**

$$S_k = \frac{2\pi^{1/2}\left(\frac{\pi|k|}{n}\right)^{\frac{\gamma-1}{2}} K_{\frac{\gamma-1}{2}}\left(\frac{2\pi|k|}{n}\right)}{\Gamma\left(\frac{\gamma}{2}\right)}, \tag{13.18}$$

which satisfies $S_k \geq 0, \forall k$. Now, we can skip step (0) and take this expression for S_k as the starting point for the algorithm. The resulting correlation function will not be given *exactly* by (13.15) but will still have the same asymptotic behavior $C_\ell \sim \ell^{-\gamma}$. This was, essentially, the procedure used in [65].

10) Typically the imaginary part of `x(i)` turns out to be a very small number, of the order of 10^{-12} or less, due to round-off errors of the fast Fourier transform routines.

Exercises

1) Prove equation (13.8).
2) Prove equation (13.12).
3) Run the program to generate n Gaussian variables distributed according to the 1-d free Lagrangian, compute the correlation function of the resulting numbers and compare with the exact result

$$C_\ell = \frac{\sinh((n-\ell)x) + \sinh(\ell x)}{4\sinh(x)\sinh^2\left(\frac{nx}{2}\right)^2},$$

being $x = \text{argcosh}(1 + a/2)$.
4) Take the correlation function C_ℓ of the previous exercise as the starting point and use the method based on the translational invariance property to generate values of Gaussian random variables distributed according to the 1-d free Lagrangian.
5) Compare the efficiency of the general method explained in section 3.5 and the one based on Fourier transforms to generate n Gaussian variables distributed according to the 1-d free Lagrangian.
6) Generalization to d dimensions: Consider a set of $n = L^d$ variables labeled as $x_{i_1,i_2,\ldots,i_d}$ with $i_k = 1,\ldots,L$, such that the exponent of the quadratic form is

$$\mathcal{L}_0(x_1,\ldots,x_n) = \frac{1}{2}\sum_{i_1=1}^{L}\cdots\sum_{i_d=1}^{L}\left[\sum_{\mu=1}^{d}(x_{\vec{i}_\mu} - x_{\vec{i}})^2 + a x_{\vec{i}}^2\right], \quad a > 0.$$

where we have introduced the notation $\vec{i} = (i_1,\ldots,i_d)$ and $\vec{i}_\mu = (i_1,\ldots,i_\mu + 1,\ldots,i_d)$. Write down a program to generate Gaussian variables distributed according to the free Lagrangian in d dimensions.
7) Prove that if $(x_1,\ldots,x_n)$ is a set of independent Gaussian variables of mean zero and variance one, then its discrete Fourier transform $\hat{x}_k$ satisfies $\langle|\hat{x}_k|^2\rangle = n$ for $k = 0,\ldots,n-1$.
8) Show that for the correlation function (13.14), the minimum value of S_k occurs at $k = n/2$ and that in order to keep the quadratic form $\sum_{i,j} x_i A_{i,j} x_j$ positive definite, C_0 must satisfy:

$$C_0 \geq (-1)^{n/2}\left(\frac{2}{n}\right)^\gamma - 2\sum_{\ell=1}^{n/2}\frac{(-1)^\ell}{\ell^\gamma}.$$

For large n this tends to $2(1-2^{1-\gamma})\zeta(\gamma)$, being $\zeta(\gamma)$ the Riemann zeta function.
9) Use (13.18) to generate Gaussian random numbers as explained in the text, and check that the resulting correlation function tends asymptotically to a power-law with exponent $-\gamma$.