THE FORCED OSCILLATOR METHOD: EIGENVALUE ANALYSIS AND COMPUTING LINEAR RESPONSE FUNCTIONS

Tsuneyoshi NAKAYAMA, Kousuke YAKUBO

[a]*Department of Applied Physics, Hokkaido University, Sapporo 060-8628, Japan*

ELSEVIER

AMSTERDAM – LONDON – NEW YORK – OXFORD – PARIS – SHANNON – TOKYO

ELSEVIER

Physics Reports 349 (2001) 239–299

PHYSICS REPORTS

www.elsevier.com/locate/physrep

The forced oscillator method: eigenvalue analysis and computing linear response functions

Tsuneyoshi Nakayama, Kousuke Yakubo*

Department of Applied Physics, Hokkaido University, Sapporo 060-8628, Japan

Received September 2000; editor: A.A. Maradudin

Contents

*Corresponding author.

E-mail addresses: tnaka@eng.hokudai.ac.jp (T. Nakayama), yakubo@eng.hokudai.ac.jp (K. Yakubo).

0370-1573/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.

PII: S0370-1573(00)00115-0

Abstract

A review is given of the forced oscillator method (FOM), an algorithm particularly suitable to treat physical systems described by very large matrices. This scheme enables us to compute spectral densities, eigenvalues and their eigenvectors of both Hermitian and non-Hermitian matrices with high speed and accuracy, in particular when combined with the fast time-evolution method based on the Chebyshev polynomial expansion. In addition, linear response functions can be computed with high speed and accuracy in the context of the FOM. The emphasis will be on the presentation of the efficiency of the FOM for a broad range of applications with their computer source codes for the purpose of wide utility. © 2001 Elsevier Science B.V. All rights reserved.

PACS: 02.70. − c; 02.60.Dc; 63.50. + x; 31.15.Qg; 75.40.Mg

Keywords: Forced oscillator method; Large-scale matrices; Eigenvalue analysis; Linear response function

1. Introduction

In numerical analyses, the eigenvalue analysis of large matrices is often the most fundamental part. As sizes of Hamiltonian matrices become large, calculations by conventional methods become difficult since computing times as well as required memory space grows rapidly. So far, many algorithms suitable to treat *very large* matrices have been developed. Among these, numerical routines tridiagonalizing large matrices are widely employed, such as Lanczos or Householders method [1–5]. Once a given matrix has been reduced to tridiagonal form, it is easy to determine its eigenvalues by using the bisection scheme and the inverse iteration routine [6].

The forced oscillator method (FOM) [7,8] has offered a quite different scheme for computing spectral densities, eigenvalues and their eigenvectors of large-scale matrices, in addition to computing linear response functions such as the Kubo formula for AC conductivities [9]. The FOM is quite unique among algorithms for eigenvalue analysis, showing high performance when applied to *very large* matrices. The FOM utilizes a principle of Hamilton mechanics: a linear lattice dynamical system driven by a periodic external force of frequency Ω will respond with large amplitudes in those eigenmodes close to this frequency [7]. Namely, the eigenvalue analysis is reduced to the solution for the time development of the equations of motion. Particular advantages of the FOM lie in its simplicity, speed, and memory efficiency. Source codes for FOM-based programs can be easily vectorized for implementation on an array- or parallel-processing modern supercomputer. The FOM can treat numerically large-scale matrices of size $\sim 10^7 \times 10^7$ or more by using computers with 1 Gbyte memory space within a reasonable computing time.

The FOM was originally presented by Williams and Maris [7] in 1985, 15 years ago, for the purpose of calculating eigenfrequencies and their eigenmodes of a lattice dynamical problem described by real symmetric matrices [10]. It is straightforward to extend the FOM for analyses of both Hermitian and non-Hermitian matrices with complex elements [11]. Thus, we can deal with eigenvalue problems not only of lattice dynamics, but also for general types of matrices by mapping them onto those of lattice-dynamical equations of motion.

The most time-consuming part in the FOM is to solve lattice-dynamical equations of motion, for which the modified Euler method [12,13] (see Section 3.1) had been adopted [7]. This article demonstrates that the global propagator method or the fast time-evolution method, which we call the FEM hereafter, remarkably enhances the efficiency of the FOM. The time evolution of large-scale dynamical systems is calculated on the basis of the Chebyshev polynomial expansion of the formal operator solution of the Schrödinger equation [14–18] or the general type of Sturm–Liouville differential equations [19]. The FEM enables us to calculate or simulate the state of a dynamical system at arbitrary time t with extraordinarily high speed and accuracy. Though the FEM is not appropriate to pursue the state at each time step Δt, this disadvantage turns the advantage for the use of the FOM not requiring the intermediate time-developed state. The computing time of the FOM incorporating the FEM is *greatly* reduced (by about 10 times) compared to the case using the modified Euler method.

In this review, we demonstrate with examples the high performance of the FOM, in particular, when combined with the FEM. Source codes are given for the purpose of wide utility. In Section 2, the algorithms to compute the spectral density, eigenvalues and their eigenvectors are introduced, as well a method to map a general eigenvalue problem for a matrix with real eigenvalues onto a lattice dynamical problem. Section 3 describes in detail the algorithm of the FEM which accelerates remarkably the calculation of eigenvalues and eigenvectors. Section 4 describes the implementation of the FOM, i.e., how to evaluate the purity and accuracy of calculated eigenvectors and eigenvalues, how to choose the most efficient time-interval T. Fortran source codes are then given. Algorithms for computing linear response functions of both classical and quantum systems are presented in the context of the FOM in Sections 5 and 6, respectively. Section 7 shows that the finite-time scaling approach for the FOM provides a useful way to study the critical dynamics near quantum phase transitions. The extension to non-Hermitian matrices is straightforward and its applications are demonstrated in Section 8. Section 9 gives a brief description of the unstable-oscillator method (UOM) [20,21], which enables us to calculate the extreme eigenvalues and their eigenvectors with high computational performance. Although the UOM does not belong to the family of the FOM, there exists a close relationship between these two methods. Conclusions are given in the final section. This article will be a good introduction of the FOM to researchers and students engaging in numerical studies in physical science and engineering.

2. Eigenvalue problem

2.1. Mapping onto lattice dynamical equations of motion

In this section, we focus our attention on a real symmetric matrix $\{D_{mn}\}$. The generalization to a general type of Hermitian matrices with complex elements is straightforward. This will be presented in Section 6 together with a scheme for computing linear response function for quantum systems. The extension to non-Hermitian matrices is given in Section 8.

Assume that the matrix $\{D_{mn}\}$ has a set of eigenvectors $\boldsymbol{e}(\lambda)$ belonging to eigenvalue ε_λ defined by

$$\varepsilon_\lambda e_m(\lambda) = \sum_n D_{mn} e_n(\lambda) . \tag{2.1}$$

Since the matrix $\{D_{mn}\}$ is real symmetric, all eigenvalues are real and eigenvectors belonging to different eigenvalues are orthogonal. Eigenvector $\boldsymbol{e}(\lambda)$ is chosen to satisfy the orthonormal condition $\sum_m e_m(\lambda)e_m(\lambda') = \delta_{\lambda\lambda'}$. If the matrix $\{D_{mn}\}$ has negative eigenvalues, we add an appropriate amount of $\varepsilon_0 \geq |\varepsilon_{\min}|$ to the diagonal elements of the matrix $\{D_{mn}\}$ so that the minimum eigenvalue $\varepsilon_{\min} + \varepsilon_0$ can be always positive. This is due to the fact that the eigenfrequencies of mapped lattice dynamical systems should be real as understood below.

The mapping of Eq. (2.1) onto the equations of lattice dynamics is done by

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} x_m(t) = -\sum_n D'_{mn} x_n(t) , \tag{2.2}$$

where

$$D'_{mn} = D_{mn} + \delta_{mn}\varepsilon_0 \tag{2.3}$$

and $x_m(t)$ denotes the displacement of the particle on the mth site. In Eq. (2.2), the mass of the particle at the mth site is supposed to be unity and D'_{mn} plays a role of the force constant between mth and nth particles. For standard lattice dynamics, the condition between force constants $\sum_n D'_{mn} = 0$ should hold from the infinitesimal translational-symmetry of the system as a whole, which is found by putting $x_m(t) = \text{const.}$ for any m in Eq. (2.2) [10]. It is not necessary, in general, to impose this condition in our cases so that we can treat an arbitrary set of the force constants D_{mn} depending on the problem in hand.

Each displacement $x_m(t)$ can be decomposed into a sum of normal modes as

$$x_m(t) = \sum_\lambda Q_\lambda(t) e_m(\lambda) , \tag{2.4}$$

where $Q_\lambda(t)$ is the time-dependent amplitude with which the mode λ contributes to $x_m(t)$, and varies as $\sim \exp(-\mathrm{i}\mu_\lambda t)$ $(\mu_\lambda^2 \equiv \varepsilon_\lambda + \varepsilon_0)$. The squared μ_λ^2 comes from the second derivative with respect to time t of Eq. (2.2), as seen by substituting Eq. (2.4) into Eq. (2.2). Since μ_λ^2 should be positive, the matrix D_{mn} should be modified into D'_{mn} expressed by Eq. (2.3). Hereafter we call μ_λ and ε_λ, respectively, as eigenfrequency and eigenvalue for clarity.

2.2. Spectral density

The spectral density is calculated from Eq. (2.2) by the following procedures [7]. The displacement $x_m(t)$ and the velocity $\dot{x}_m(t)$ are set to be zero at $t = 0$ in Eq. (2.2). Then the periodic force $F_m \cos(\Omega t)$ is imposed on the mth site. Here F_m should be chosen as

$$F_m = F_0 \cos(\phi_m) , \tag{2.5}$$

where ϕ_m is a random quantity distributed uniformly in the range $0 \le \phi_m \le 2\pi$, and F_0 is a constant.

As a next step, we introduce the energy function $E(t)$ of the system given by

$$\begin{aligned} E(t) &= \frac{1}{2}\left\{\sum_m \dot{x}_m^2(t) + \sum_m \sum_n x_m(t) D'_{mn} x_n(t)\right\} \\ &= \frac{1}{2}\sum_\lambda \{\dot{Q}_\lambda^2(t) + \mu_\lambda^2 Q_\lambda^2(t)\} . \end{aligned} \tag{2.6}$$

In deriving the last relation from Eq. (2.4), the orthonormal condition between eigenvectors $\boldsymbol{e}(\lambda)$ is used. Let us define the quantity $\xi_\lambda(t)$ expressed by

$$\xi_\lambda(t) \equiv \dot{Q}_\lambda(t) + \mathrm{i}\mu_\lambda Q_\lambda(t) . \tag{2.7}$$

Using this quantity, the r.h.s. of Eq. (2.6) leads to

$$E(t) = \frac{1}{2}\sum_\lambda |\xi_\lambda(t)|^2 . \tag{2.8}$$

From Eqs. (2.2), (2.4) and (2.5), the equation of motion under the external periodic force,

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} x_m(t) = -\sum_n D'_{mn} x_n(t) + F_m \cos(\Omega t) \tag{2.9}$$

becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_\lambda(t) - \mathrm{i}\mu_\lambda \xi_\lambda(t) = \sum_m F_m \cos(\Omega t) e_m(\lambda) \ . \tag{2.10}$$

Solving analytically this equation under the initial condition $\xi_\lambda(0) = 0$, the quantity ξ_λ at $t = T$ is expressed by

$$\xi_\lambda(T) = \frac{\mathrm{e}^{\mathrm{i}\mu_\lambda T}}{2}\left\{\sum_m F_m e_m(\lambda)\right\}\frac{\mathrm{e}^{\mathrm{i}(\Omega-\mu_\lambda)T} - 1}{\mathrm{i}(\Omega - \mu_\lambda)} \ . \tag{2.11}$$

Thus, one has the energy function of the system $E(T)$ by combining Eqs. (2.8) and (2.11) as

$$E(T) = \frac{1}{2}\sum_\lambda \left\{\sum_m F_m e_m(\lambda)\right\}^2 \frac{\sin^2\{(\mu_\lambda - \Omega)T/2\}}{(\mu_\lambda - \Omega)^2} \ . \tag{2.12}$$

The averaged value of $E(T)$ over ϕ_m provides

$$\begin{aligned}\langle E(T)\rangle &= \frac{F_0^2}{2}\sum_\lambda \frac{\sin^2\{(\mu_\lambda - \Omega)T/2\}}{(\mu_\lambda - \Omega)^2}\left\langle \sum_m \sum_n e_m(\lambda) e_n(\lambda)\cos(\phi_m)\cos(\phi_n)\right\rangle \\ &= \frac{F_0^2}{4}\sum_\lambda \frac{\sin^2\{(\mu_\lambda - \Omega)T/2\}}{(\mu_\lambda - \Omega)^2} \ ,\end{aligned} \tag{2.13}$$

where the terms satisfying $m = n$ remain in the summations over m and n. After a sufficiently large time T, only modes λ's belonging to eigenfrequencies μ_λ in the vicinity of Ω contribute to the sum in Eq. (2.13). For a large system size N (the size of the matrix $\{D_{mn}\}$), *it is not necessary to average over all possible ensemble* $\{\phi_m\}$ *explicitly*. It suffices to choose a single configuration of $\{\phi_m\}$. For a proper time-interval T, Eq. (2.13) yields

$$\begin{aligned}\langle E(T)\rangle &\approx \frac{\pi T F_0^2}{8}\sum_\lambda \delta(\mu_\lambda - \Omega) \\ &= \frac{\pi T N F_0^2}{8}\tilde{\mathscr{D}}(\Omega) \ ,\end{aligned} \tag{2.14}$$

where $\tilde{\mathscr{D}}(\Omega)$ is the density of states for the mapped system characterized by D'_{mn}. Thus, we can relate the energy function $E(T)$ to the density of states $\tilde{\mathscr{D}}(\mu)$.

The spectral density $\mathscr{D}(\varepsilon)$ for the original matrix D_{mn} is obtained by multiplying $\tilde{\mathscr{D}}$ by the Jacobian

$$\begin{aligned}\mathscr{D}(\varepsilon) &= \left|\frac{\mathrm{d}\mu(\varepsilon)}{\mathrm{d}\varepsilon}\right|\tilde{\mathscr{D}}(\mu) \\ &= \frac{4}{\pi T N F_0^2\sqrt{\varepsilon + \varepsilon_0}}\langle E(T)\rangle \ .\end{aligned} \tag{2.15}$$

The calculated spectral density $\mathscr{D}(\varepsilon)$ should be normalized as

$$\int_{-\infty}^{\infty} \mathscr{D}(\varepsilon)\,\mathrm{d}\varepsilon = 1 . \tag{2.16}$$

To summarize this section, we should point out the following advantages for computing the spectral density: (i) The spectral density can be obtained within a required *arbitrary* energy range. (ii) The energy resolution can be controlled by choosing the time-interval T [see Eq. (2.13)]. This is particularly efficient for calculating the smooth part of the spectral density. (iii) The computing time scales linearly with the matrix size N in the case of sparse matrices.

2.3. Eigenvalues and their eigenvectors

By solving Eq. (2.9) under the initial conditions $x_m(t=0)=0$ and $\dot{x}_m(t=0)=0$, the displacement $x_m(t)$ after the time-interval T is expressed by

$$x_m(T) = \sum_{\lambda} F_\lambda h(\Omega, \mu_\lambda, T) e_m(\lambda) , \tag{2.17}$$

where

$$F_\lambda = \sum_m F_m e_m(\lambda) \tag{2.18}$$

and

$$h(\Omega, \mu_\lambda, t) = \frac{2\sin\{(\Omega + \mu_\lambda)t/2\}\sin\{(\Omega - \mu_\lambda)t/2\}}{\Omega^2 - \mu_\lambda^2} . \tag{2.19}$$

For a sufficiently large time T, only a few eigenmodes with eigenfrequencies μ_λ close to Ω contribute to the λ summation in Eq. (2.17). One can accelerate the calculation by, after the time interval-T, replacing the amplitude of the periodic force F_m at the mth site by

$$F_m = x_m(T) . \tag{2.20}$$

The initial values $x_m(t=0)$ and $\dot{x}_m(t=0)$ at the mth site are set to be zero again, and we follow the time development of Eq. (2.9). After p iterations of this procedure, the displacement $x_m(T)$ becomes

$$x_m^{(p)}(T) = \sum_{\lambda} F_\lambda h^p(\Omega, \mu_\lambda, T) e_m(\lambda) . \tag{2.21}$$

After a sufficiently large p, only a single eigenmode λ_1 ($\mu_{\lambda_1} \approx \Omega$) survives such as

$$x_m^{(p)}(T) \approx C e_m(\lambda_1) , \tag{2.22}$$

where C is a constant. This relation is used to calculate a precise eigenvalue μ_{λ_1} as shown in Section 4.

3. Time development of large-scale dynamical systems

3.1. Standard numerical methods

It is necessary, in order to solve eigenvalue problems in terms of the FOM, to calculate the time development of coupled equations of motion of forced oscillators given by Eq. (2.9). Since the most time-consuming part of the FOM lies in this, the efficient routine for Eq. (2.9) is crucial for high performance.

The simplest way to integrate an ordinary differential equation such as Eq. (2.9) is the Euler method [12,13]. In this method, Eq. (2.9) is divided into a set of two equations,

$$\frac{\mathrm{d}}{\mathrm{d}t} z_m(t) = -\sum_n D'_{mn} x_n(t) + F_m \cos(\Omega t) ,$$
$$\frac{\mathrm{d}}{\mathrm{d}t} x_m(t) = z_m(t) . \tag{3.1}$$

By discretizing time t into a step τ, Eq. (3.1) becomes

$$z_m(l+1) = z_m(l) - \tau\left[\sum_n D'_{mn} x_n(l) - F_m \cos(\Omega\tau l)\right] ,$$
$$x_m(l+1) = x_m(l) + \tau z_m(l) , \tag{3.2}$$

where $x_m(l)$ and $z_m(l)$ are the displacement and the velocity of the mth particle with mass $m = 1$ at the time $t = l\tau$ with integer l. For simplicity, we illustrate here a system with a single degree of freedom, oscillating with the frequency μ in the absence of the external force. In this case, one can write down Eq. (3.2) with $F_m = 0$ in the matrix form

$$\begin{pmatrix} z(l+1) \\ x(l+1) \end{pmatrix} = \begin{pmatrix} 1 & -\mu^2\tau \\ \tau & 1 \end{pmatrix} \begin{pmatrix} z(l) \\ x(l) \end{pmatrix} . \tag{3.3}$$

The determinant J of the matrix in Eq. (3.3) is $J = 1 + \mu^2\tau^2$ being always larger than unity. This means that the amplitude of the oscillation eventually diverges even if τ is chosen to be arbitrarily small.

Williams and Maris [7] have employed a modified version of the discretization scheme as follows:

$$z_m(l+1) = z_m(l) - \tau\left[\sum_n D'_{mn} x_n(l) - F_m \cos(\Omega\tau l)\right] ,$$
$$x_m(l+1) = x_m(l) + \tau z_m(l+1) . \tag{3.4}$$

This modified Euler method [12,13], sometimes called the Verlet method [22], yields a matrix representation of coupled equations corresponding to Eq. (3.3) of the form

$$\begin{pmatrix} z(l+1) \\ x(l+1) \end{pmatrix} = \begin{pmatrix} 1 & -\mu^2\tau \\ \tau & 1-\mu^2\tau^2 \end{pmatrix} \begin{pmatrix} z(l) \\ x(l) \end{pmatrix} , \tag{3.5}$$

for which one has the Jacobian determinant $J = 1$ for an arbitrary τ. One should remark that τ is bounded within a finite range. Solving Eq. (3.5), one finds that $x(l)$ oscillates with the frequency (see Section 9 and Ref. [7]).

$$\mu^* = \frac{2}{\tau}\sin^{-1}\left(\frac{\mu\tau}{2}\right) . \qquad (3.6)$$

From Eq. (3.6), the effective frequency μ^* is close to the true frequency μ only for $\mu\tau \ll 2$, but μ^* becomes complex if $\mu\tau > 2$. Thus, τ should be chosen less than $2/\mu$. For a system with many degrees of freedom, this condition should be understood as

$$\mu_{\max}\tau < 2 , \qquad (3.7)$$

where $\mu_{\max}$ is the maximum frequency of a mapped dynamical system. When $\mu_{\max}\tau \lesssim 2$, we must correct the frequency in accordance with Eq. (3.6). It should be emphasized that the accuracy of x_m or z_m is $O(\tau^2)$ after the correction by Eq. (3.6). This is because the modified Euler method is based on the second-order expansion of Eq. (3.1). Therefore, we must take a quite small value of τ to obtain accurate results, which implies that even the modified Euler method consumes a large amount of computing time.

The Runge–Kutta method provides a higher-order expansion of Eq. (3.1) [1,23,24]. This method makes it possible to calculate x_m and z_m very accurately (the nth-order Runge–Kutta method gives the accuracy of $O(\tau^n)$). However, the computing time of the fourth-order Runge–Kutta method, as an example, takes at least four times larger than the case using the Euler method with the same time step τ. This is not suitable for evaluating the time evolution in the FOM. Any other numerical technique to integrate ordinary differential equations also requires a longer computing time for obtaining reasonable accuracy. The common feature in these methods is that equations of motion are expanded in the time domain [1]. Thus, the dynamical variables (x_m and z_m) at every time step up to $t = T$ are computed. For the FOM, one does not require the time development of dynamical variables *during* the time interval $0 < t < T$. We need only the displacements and the velocities at $t = T$. A numerical method remarkably suitable for this purpose has been proposed [14–19]. By incorporating this method, which we call hereafter the fast time-evolution method (FEM) after Ref. [19], the efficiency of the FOM is extraordinarily enhanced as shown in the next subsection.

3.2. Fast time-evolution method

The fast time-evolution method (FEM) [14–19] enables us to obtain directly displacements and velocities of a dynamical system at an arbitrary time t, without pursuing displacements (or velocities) during the time interval $0 < t < T$. The method is based on the Chebyshev polynomial expansion of the formal solution of the ordinary differential equations in the *eigenfrequency* domain. It is possible to compute very accurately displacements and velocities by the FEM with an extremely short computing time which scales linearly with the number of variables and evolution time. Although the FEM is applicable to any type of Sturm–Liouville differential equations, we focus our attention onto the lattice dynamical equations of motion such as Eq. (2.9). We describe here the details according to Ref. [19].

Eq. (2.9) can be symbolically written as

$$\left(\frac{\mathrm{d}^2}{\mathrm{d}t^2} + \boldsymbol{D}\right)\boldsymbol{x} = \boldsymbol{f}\cos(\Omega t)\,, \tag{3.8}$$

where $\boldsymbol{D}$, $\boldsymbol{x}$, and $\boldsymbol{f}$ are the matrix and the vectors whose elements are D'_{mn}, x_m, and F_m, respectively. Here, we denote matrices and vectors by capital and lower-case boldface letters, respectively. The initial conditions are set as

$$\begin{aligned} &\boldsymbol{x}(0) = \boldsymbol{x}_0\,, \\ &\dot{\boldsymbol{x}}(0) = \boldsymbol{z}_0\,. \end{aligned} \tag{3.9}$$

The formal solution of Eq. (3.8) under the initial conditions Eq. (3.9) is given by

$$\boldsymbol{x}(t) = \cos(\sqrt{\boldsymbol{D}}t)\,x_0 + \frac{\sin(\sqrt{\boldsymbol{D}}t)}{\sqrt{\boldsymbol{D}}}z_0 + \left[\frac{\cos(\Omega t)\boldsymbol{I} - \cos(\sqrt{\boldsymbol{D}}t)}{\boldsymbol{D} - \Omega^2\boldsymbol{I}}\right]\boldsymbol{f}\,, \tag{3.10}$$

where $\boldsymbol{I}$ is the unit matrix. Functions of the matrix $\boldsymbol{D}$ should be understood as power series such as a Taylor expansion.

In the FEM the time t is regarded as a parameter and we expand Eq. (3.10) in terms of a complete set of functions $\phi_p(\boldsymbol{D})$. The functions $\phi_p(\boldsymbol{D})$ should satisfy the following conditions: (i) Expansions of functions $\cos(\sqrt{\boldsymbol{D}}t)$, $\sin(\sqrt{\boldsymbol{D}}t)/\sqrt{\boldsymbol{D}}$, and $[\cos(\Omega t)\boldsymbol{I} - \cos(\sqrt{\boldsymbol{D}}t)]/(\boldsymbol{D} - \Omega^2\boldsymbol{I})$ by $\{\phi_p(\boldsymbol{D})\}$ should rapidly converge for the purpose of saving computer resources such as computing time and memory. (ii) The functions $\phi_p(\boldsymbol{D})$ are orthogonal polynomials of $\boldsymbol{D}$, because $\phi_p(\boldsymbol{D})$ should be easily calculated from $\boldsymbol{D}$, e.g. using the recursion relation. (iii) The functions $\phi_p(\boldsymbol{D})$ have to be defined within a finite eigenvalue domain, because the spectrum of $\boldsymbol{D}$ is bounded. The Chebyshev polynomials $T_p(\boldsymbol{D})$ satisfy all these conditions. The condition (i) is, in particular, guaranteed by the fact that the Chebyshev polynomials are almost the *minimax* polynomials [1] and the error is uniformly distributed in the eigenvalue domain.

Since the domain of the Chebyshev polynomials is $[-1,1]$, the matrix $\boldsymbol{D}$ should be converted to the matrix $\boldsymbol{G}$ having a spectrum bounded in the range $[-1,1]$. Assuming $\varepsilon_{\max}$ and $\varepsilon_{\min}$ to be the maximum and the minimum eigenvalues of the matrix $\boldsymbol{D}$, we have

$$\boldsymbol{G} = \frac{2}{\varepsilon_{\max} - \varepsilon_{\min}}\boldsymbol{D} - \frac{\varepsilon_{\max} + \varepsilon_{\min}}{\varepsilon_{\max} - \varepsilon_{\min}}\boldsymbol{I}\,, \tag{3.11}$$

where both $\varepsilon_{\max}$ and $\varepsilon_{\min}$ are estimated by the Gerschgorin's theorem [25,26] as will be mentioned in Section 4 [$\varepsilon_{\max}$ and $\varepsilon_{\min}$ are positive because of Eq. (2.3)]. The functions of the matrix $\boldsymbol{D}$ in Eq. (3.10) are expanded as follows:

$$\cos(\sqrt{\boldsymbol{D}}t) = \sum_{p=0}^{\infty} a_p(t)T_p(\boldsymbol{G})\,, \tag{3.12}$$

$$\frac{\sin(\sqrt{\boldsymbol{D}}t)}{\sqrt{\boldsymbol{D}}} = \sum_{p=0}^{\infty} b_p(t)T_p(\boldsymbol{G})\,, \tag{3.13}$$

$$\frac{\cos(\Omega t)\boldsymbol{I} - \cos(\sqrt{\boldsymbol{D}}t)}{\boldsymbol{D} - \Omega^2\boldsymbol{I}} = \sum_{p=0}^{\infty} c_p(t)T_p(\boldsymbol{G})\,. \tag{3.14}$$

In actual calculations, the infinite upper limit of the summations in Eqs. (3.12)–(3.14) should be truncated up to $P-1$, where P is an appropriate cutoff so that $|a_p(t)|$, $|b_p(t)|$, and $|c_p(t)|$ with $p \geq P$ are much smaller than unity. As a result, Eq. (3.10) becomes

$$\boldsymbol{x}(t) \approx \left[\sum_{p=0}^{P-1} a_p(t) T_p(\boldsymbol{G})\right]\boldsymbol{x}_0 + \left[\sum_{p=0}^{P-1} b_p(t) T_p(\boldsymbol{G})\right]\boldsymbol{z}_0 + \left[\sum_{p=0}^{P-1} c_p(t) T_p(\boldsymbol{G})\right]\boldsymbol{f}\,. \tag{3.15}$$

The Chebyshev polynomials satisfy the discrete orthogonality relation given by [1,27,28]

$$\sum_{k=1}^{m} T_p(x_k) T_q(x_k) = \frac{m}{2}\delta_{pq}(1+\delta_{p0})\,, \tag{3.16}$$

where x_k ($k = 1, 2, \ldots, m$) are the kth zeros of $T_m(x)$, namely,

$$x_k = \cos\left[\frac{\pi(k-\frac{1}{2})}{m}\right], \tag{3.17}$$

and $p, q < m$. Using Eq. (3.16), the formula $T_p(x) = \cos[p\cos^{-1}(x)]$, and the orthonormal conditions for $T_p(x)$ expressed by [28]

$$\int_{-1}^{1} \frac{T_p(x) T_q(x)}{\sqrt{1-x^2}}\,\mathrm{d}x = \frac{\pi}{2}\delta_{pq}(1+\delta_{p0})\,, \tag{3.18}$$

one can obtain the explicit forms of coefficients $a_p(t), b_p(t)$, and $c_p(t)$ defined by Eqs. (3.12)–(3.14) as follows:

$$a_p(t) = \frac{2}{(1+\delta_{p0})P}\sum_{q=0}^{P-1} \cos\left[\frac{\pi p(q+\frac{1}{2})}{P}\right]\cos(\xi_q t)\,, \tag{3.19}$$

$$b_p(t) = \frac{2}{(1+\delta_{p0})P}\sum_{q=0}^{P-1} \cos\left[\frac{\pi p(q+\frac{1}{2})}{P}\right]\frac{\sin(\xi_q t)}{\xi_q}\,, \tag{3.20}$$

$$c_p(t) = \frac{2}{(1+\delta_{p0})P}\sum_{q=0}^{P-1} \cos\left[\frac{\pi p(q+\frac{1}{2})}{P}\right]\frac{\cos(\Omega t) - \cos(\xi_q t)}{\xi_q^2 - \Omega^2}\,, \tag{3.21}$$

where

$$\xi_q = \left\{\frac{\varepsilon_{\max}+\varepsilon_{\min}}{2} + \frac{\varepsilon_{\max}-\varepsilon_{\min}}{2}\cos\left[\frac{\pi(q+\frac{1}{2})}{P}\right]\right\}^{1/2}\,. \tag{3.22}$$

The velocity $\boldsymbol{z}$ is directly obtained from Eqs. (3.15) and (3.19)–(3.21):

$$\boldsymbol{z}(t) \approx \left[\sum_{p=0}^{P-1} \dot{a}_p(t) T_p(\boldsymbol{G})\right]\boldsymbol{x}_0 + \left[\sum_{p=0}^{P-1} \dot{b}_p(t) T_p(\boldsymbol{G})\right]\boldsymbol{z}_0 + \left[\sum_{p=0}^{P-1} \dot{c}_p(t) T_p(\boldsymbol{G})\right]\boldsymbol{f}\,, \tag{3.23}$$

where

$$\dot{a}_p(t) = -\frac{2}{(1+\delta_{p0})P}\sum_{q=0}^{P-1} \cos\left[\frac{\pi p(q+\frac{1}{2})}{P}\right]\xi_q \sin(\xi_q t)\,, \tag{3.24}$$

$$\dot{b}_p(t) = a_p(t)\,, \tag{3.25}$$

$$\dot{c}_p(t) = \frac{2}{(1+\delta_{p0})P}\sum_{q=0}^{P-1}\cos\left[\frac{\pi p(q+\frac{1}{2})}{P}\right]\frac{\xi_q\sin(\xi_q t) - \Omega\sin(\Omega t)}{\xi_q^2 - \Omega^2} \ . \tag{3.26}$$

It should be noted that these coefficients reflect the properties of the matrix $\boldsymbol{G}$ (then $\boldsymbol{D}$) only via $\varepsilon_{\max}$ and $\varepsilon_{\min}$.

In calculating $\boldsymbol{x}(t)$ or $\boldsymbol{z}(t)$ by Eq. (3.15) or (3.23), two parts of computations seem to require long computing times. One is the calculation of the Chebyshev polynomials $T_p(\boldsymbol{G})$ of the matrix $\boldsymbol{G}$. The recurrence formulae of the Chebyshev polynomials, however, make this calculation very fast, which are given by [28]

$$T_1(x) = xT_0(x) \ ,$$

$$T_{p+1}(x) = 2xT_p(x) - T_{p-1}(x) \ . \tag{3.27}$$

Actually, what we need is not $T_p(\boldsymbol{G})$ itself, but the vectors such as $\boldsymbol{x}_p \equiv T_p(\boldsymbol{G})\boldsymbol{x}_0$. From Eq. (3.27), these vectors are given by

$$\boldsymbol{x}_1 = \boldsymbol{G}\boldsymbol{x}_0 \ , \tag{3.28a}$$

$$\boldsymbol{x}_{p+1} = 2\boldsymbol{G}\boldsymbol{x}_p - \boldsymbol{x}_{p-1} \ . \tag{3.28b}$$

Notice that there are no matrix–matrix multiplications in Eq. (3.28) which take considerable computing time. Calculations of the coefficients $a_p(t), b_p(t)$, and $c_p(t)$ in Eqs. (3.19)–(3.21) seem to require P^2 operations. However, all of these expressions given by Eqs. (3.19)–(3.21) and (3.24)–(3.26) take the form of

$$W_p = \sum_{q=0}^{P-1} w_q \cos\left[\frac{\pi p(q+\frac{1}{2})}{P}\right] . \tag{3.29}$$

This is the Fourier cosine transform. The fast Fourier transform (FFT) technique reduces the number of operations in computing the coefficients from P^2 to $P\log_2 P$, which implies that the coefficients are calculated in quite a short time.

In order to estimate the appropriate value of the cutoff parameter P, we illustrate with a very simple situation, i.e., a single pendulum described by $\ddot{x} + \varepsilon x = 0$ with the initial conditions of $x(0) = x_0$ and $\dot{x}(0) = 0$. Assume that ε can take values in the range $[\varepsilon_{\min}, \varepsilon_{\max}]$. Corresponding to Eq. (3.10), we have the solution $x(t) = x_0\cos(\sqrt{\varepsilon}t)$. The Chebyshev expansion of $\cos(\sqrt{\varepsilon}t)$ is written as $\cos(\sqrt{\varepsilon}t) = \sum_{p=0}^{\infty} a_p(t)T_p(\varepsilon')$, where $\varepsilon' = 2\varepsilon/(\varepsilon_{\max} - \varepsilon_{\min}) - (\varepsilon_{\max} + \varepsilon_{\min})/(\varepsilon_{\max} - \varepsilon_{\min})$. Using the orthonormal conditions Eq. (3.18), we have

$$a_p(t) = \frac{2}{(1+\delta_{p0})\pi}\int_{-1}^{1}\frac{c(\varepsilon', t)T_p(\varepsilon')}{\sqrt{1-\varepsilon'^2}}\,\mathrm{d}\varepsilon' \ , \tag{3.30}$$

where

$$c(\varepsilon', t) = \cos\left(\sqrt{\frac{\varepsilon_{\max} - \varepsilon_{\min}}{2}\varepsilon' + \frac{\varepsilon_{\max} + \varepsilon_{\min}}{2}}t\right) . \tag{3.31}$$

The integrand of Eq. (3.30) contains the product of two oscillating functions. The function $c(\varepsilon', t)$ oscillates $(\sqrt{\varepsilon_{\max}} - \sqrt{\varepsilon_{\min}})t/2\pi$ times in the ε' domain of $[-1, 1]$. The Chebyshev polynomial $T_p(\varepsilon')$ oscillates with p zeros. If p becomes much larger than $(\sqrt{\varepsilon_{\max}} - \sqrt{\varepsilon_{\min}})t/2\pi$, $T_p(\varepsilon')$ oscillates rapidly in a period of $c(\varepsilon', t)$ as a function of ε' and the integral of Eq. (3.30) becomes negligible. This argument can be extended to a system with many degrees of freedom described by Eq. (3.8). Therefore, the cutoff parameter is estimated by

$$P = \chi(\sqrt{\varepsilon_{\max}} - \sqrt{\varepsilon_{\min}})t \; . \tag{3.32}$$

The coefficient χ is of the order of unity in actual cases.

Finally, we compare the number of operations in the (modified) Euler method and the FEM. For simplicity, we assume $\varepsilon_{\min} = 0$. In the Euler method, the most time-consuming part is in the calculation of Eq. (3.2) or (3.4). The calculation of each $z_m(l+1)$ contains the summation of N (system size) terms. Therefore, it requires roughly N^2T/τ operations to obtain displacements and velocities of all sites at the time T. Since the matrix $\{D'_{mn}\}$ is usually sparse, the number of operations is actually reduced to NT/τ. If τ is chosen as $\sqrt{\varepsilon_{\max}}\tau = \text{const.}$, this becomes $\gamma_{\text{Euler}}N\sqrt{\varepsilon_{\max}}T$, where γ_{Euler} is a proportionality coefficient. On the other hand, the most time-consuming part of the FEM is the calculation of Eq. (3.28). In this procedure, we have to calculate P matrix–vector multiplications in N dimensions. This requires NP operations. Since P is proportional to $\sqrt{\varepsilon_{\max}}T$ as given by Eq. (3.32), the number of operations for the FEM becomes proportional to $N\sqrt{\varepsilon_{\max}}T$ and the computing time is given by $\gamma_{\text{FEM}}N\sqrt{\varepsilon_{\max}}T$, where γ_{FEM} is a proportionality coefficient for the FEM. Although the computing times for both methods have similar forms, i.e., $T^{\text{CPU}}_{\text{Euler}} = \gamma_{\text{Euler}}N\sqrt{\varepsilon_{\max}}T$ and $T^{\text{CPU}}_{\text{FEM}} = \gamma_{\text{FEM}}N\sqrt{\varepsilon_{\max}}T$, the coefficient γ_{FEM} is much smaller than γ_{Euler} in actual calculations. For a problem of the time evolution of a lattice-vibrational system, for example, γ_{FEM} is 12 orders of magnitude smaller than γ_{Euler}, if we compare these coefficients under the condition that both methods give the same precision of $x(t)$. Even for the largest time step τ for the modified Euler method (e.g., $\mu_{\max}\tau = 1.99$) with which the FOM presents a reliable result, the coefficient γ_{Euler} is about ten times larger than γ_{FEM}.

4. Implementation of the FOM

4.1. Evaluating purity and accuracy of calculated eigenvectors and eigenvalues

In most conventional eigenvalue analyses, eigenvalues are directly obtained by a diagonalization of a given matrix itself, and then eigenvectors belonging to these eigenvalues are computed. In contrast to this scheme, the FOM offers a quite different one. Namely eigenvectors are calculated at first as described in Section 2.3, and then we compute corresponding eigenvalues by using these eigenvectors. This section gives the scheme for evaluating the purity of eigenvectors and the accuracy of eigenvalues obtained from those eigenvectors. This routine has been described by Yakubo et al. [8].

In order to evaluate the purity of the eigenvector calculated by the FOM, we introduce the quantity δ_m given by

$$\delta_m \equiv a_m - \tilde{\mu}^2 b_m \ , \tag{4.1}$$

where a_m and b_m are

$$a_m = \sum_n D'_{mn} x_n^{(p)}(T) \tag{4.2}$$

and

$$b_m = x_m^{(p)}(T) \ . \tag{4.3}$$

We see from Eqs. (4.1)–(4.3) that δ_m vanishes for any m if $x_m^{(p)}(T) = e_m(\lambda)$. The quantity $\tilde{\mu}$ in Eq. (4.1) will be defined later.

Let us introduce the deviation δ defined by

$$\delta \equiv \frac{\sum_m \delta_m^2}{\sum_m a_m^2} \ . \tag{4.4}$$

Using Eqs. (4.1)–(4.3), we have

$$\delta^2 = \frac{\Gamma_4 - 2\tilde{\mu}^2 \Gamma_2 + \tilde{\mu}^4 \Gamma_0}{\Gamma_4} \ , \tag{4.5}$$

where

$$\Gamma_0 = \sum_m b_m^2 \ , \tag{4.6a}$$

$$\Gamma_2 = \sum_m a_m b_m \ , \tag{4.6b}$$

$$\Gamma_4 = \sum_m a_m^2 \ . \tag{4.6c}$$

By differentiating Eq. (4.5) with respect to $\tilde{\mu}^2$, the deviation δ^2 is minimized when

$$\tilde{\mu}^2 = \frac{\Gamma_2}{\Gamma_0} \ , \tag{4.7}$$

and the minimum value of δ^2 becomes

$$\delta^2 = \frac{\Gamma_0 \Gamma_4 - \Gamma_2^2}{\Gamma_0 \Gamma_4} \ . \tag{4.8}$$

Using Eqs. (2.1), (2.3), (2.21), and (4.6), Γ_ν $(\nu = 0, 2, 4)$ can be alternatively expressed as

$$\Gamma_\nu = \sum_\lambda A_\lambda^2 \mu_\lambda^\nu \ , \tag{4.9}$$

where A_λ is the amplitude of the mode λ given by

$$A_\lambda = F_\lambda h^p(\Omega, \mu_\lambda, t) , \tag{4.10}$$

and $\mu_\lambda^2 = \varepsilon_\lambda + \varepsilon_0$. In order to examine $\tilde{\mu}$ and δ, we suppose that the displacement pattern $\{x_m^{(p)}(T)\}$ primarily consists of two adjacent eigenvectors ($\lambda = \lambda_1, \lambda_2$). For this situation, one finds from Eqs. (4.7) and (4.8)

$$\tilde{\mu}^2 = \frac{A_{\lambda_1}^2 \mu_{\lambda_1}^2 + A_{\lambda_2}^2 \mu_{\lambda_2}^2}{A_{\lambda_1}^2 + A_{\lambda_2}^2} , \tag{4.11a}$$

$$\delta^2 = \frac{A_{\lambda_1}^2 A_{\lambda_2}^2 (\mu_{\lambda_1}^2 + \mu_{\lambda_2}^2)^2}{(A_{\lambda_1}^2 + A_{\lambda_2}^2)(A_{\lambda_1}^2 \mu_{\lambda_1}^4 + A_{\lambda_2}^2 \mu_{\lambda_2}^4)} . \tag{4.11b}$$

If $|\mu_{\lambda_1} - \mu_{\lambda_2}| \ll (\mu_{\lambda_1} + \mu_{\lambda_2})/2$ and $A_{\lambda_1} \gg A_{\lambda_2}$, one has

$$\delta \simeq \frac{2\Delta\mu}{\tilde{\mu}} \left| \frac{A_{\lambda_2}}{A_{\lambda_1}} \right| . \tag{4.12}$$

The frequency difference $\Delta\mu$ ($\equiv |\mu_{\lambda_1} - \mu_{\lambda_2}|$) roughly equals to $[N\tilde{\mathscr{D}}(\mu)]^{-1}$, where $\tilde{\mathscr{D}}(\mu)$ is the density of states of the mapped system. Namely, one can estimate the mixing ratio of modes ($\equiv A_{\lambda_2}/A_{\lambda_1}$) from the value of δ calculated by Eq. (4.8).

The quantity $\tilde{\mu}$ becomes very close to μ_{λ_1} in the case of $A_{\lambda_1} \gg A_{\lambda_2}$. From Eq. (4.11a), we find

$$\mu_{\lambda_1} = \tilde{\mu} + \Delta\mu \left(\frac{A_{\lambda_2}}{A_{\lambda_1}} \right) . \tag{4.13}$$

We see that the difference between $\tilde{\mu}$ and μ_{λ_1} depends on the product of two small quantities. Hence, $\tilde{\mu}$ and $\tilde{\mu}^2 - \varepsilon_0$ calculated by Eq. (4.7) are extremely good approximations to the true eigenfrequency μ_{λ_1} and the true eigenvalue ε_{λ_1}, respectively.

It is worthwhile noting that the selected mode λ_1 is not always the mode with the eigenfrequency closest to the external frequency Ω. Due to the oscillating property of the function $h(\Omega, \mu_\lambda, t)$ given by Eq. (2.19), the selected mode depends on the choice of the time interval T. An efficient way to choose the value of T will be discussed in the Section 4.3.

4.2. Choice of the efficient time-interval T for computing the spectral density

One of the merits of the FOM is that we can calculate the spectral density with an *arbitrary* resolution $\delta\varepsilon$ by choosing the proper time interval T. Let us describe the criterion for the choice of the time T to control the resolution $\delta\varepsilon$. The frequency width $\delta\mu$ for the *mapped* system (corresponding to $\delta\varepsilon$) should be chosen as

$$\delta\mu = \frac{d\mu(\varepsilon)}{d\varepsilon} \delta\varepsilon , \tag{4.14}$$

where $\delta\varepsilon$ is the eigenvalue resolution required for the eigenvalue problem of the original matrix. Eq. (2.13) indicates that the frequency width $\delta\mu$ is inversely proportional to the time T, as given by

$$\delta\mu = \frac{4\pi}{T} . \tag{4.15}$$

Since $\mu_\lambda^2 = \varepsilon_\lambda + \varepsilon_0$, for the required resolution $\delta\varepsilon$, the time interval T should be taken as

$$T = \frac{4\pi}{(|\mathrm{d}\mu(\varepsilon)/\mathrm{d}\varepsilon|\delta\varepsilon)} = \frac{8\pi\sqrt{\varepsilon + \varepsilon_0}}{\delta\varepsilon} . \tag{4.16}$$

The resolution $\delta\varepsilon$ must satisfy the condition

$$\delta\varepsilon \gg \Delta\varepsilon , \tag{4.17}$$

where $\Delta\varepsilon = 1/N\mathscr{D}(\varepsilon)$ is the mean level spacing of the eigenvalue sequence. This condition requires that there should exist a sufficient number of eigenvalues within the spectral range $\varepsilon \sim \varepsilon + \delta\varepsilon$.

As is mentioned in Section 3.2, the computing time of the FEM (and then the FOM) is proportional to T. Therefore, Eq. (4.16) suggests that the computing time becomes shorter when choosing ε_0 to be smaller. However, if we set ε_0 to be the smallest value for which all eigenvalues are positive, $\varepsilon_{\min}$ becomes zero. In this case, the spectral density in the vicinity of $\varepsilon_{\min}$ has poor accuracy because $\mathscr{D}(\varepsilon)$ is proportional to $1/\sqrt{\varepsilon + \varepsilon_0}$ as given by Eq. (2.15). The parameter ε_0 should be carefully chosen by considering these competitive conditions (see Section 4.4.1). The time interval T determined by Eq. (4.16) does not depend on N. This implies that the computing time for the spectral density is proportional to N, and then the FOM for the spectral density is an O(N) method.

4.3. Choice of the optimal time-interval T for eigenvalue analysis

As seen from the preceding discussion, the calculation of an eigenvector by the FOM includes two key parameters, namely, the driving time interval T and the number of iterations p. It is crucial for efficient calculations of eigenvectors to choose the optimal time interval T [8]. In this section, we give a discussion concerning the most efficient choice of this parameter. If $\{x_m^{(p)}(T)\}$ consists of two modes λ_1 and λ_2 as considered below in Eq. (4.10), we obtain, from Eqs. (2.20), (4.10), and (4.13), the expression for p required for a given accuracy δ_0 as

$$p \simeq \log\left|\frac{\delta_0 \mu F_{\lambda_1}}{2\Delta\mu F_{\lambda_2}}\right| \Bigg/ \left|\log\left|\frac{\sin[(\Omega - \mu_{\lambda_2})T/2]\sin[(\Omega + \mu_{\lambda_2})T/2](\Omega^2 - \mu_{\lambda_1}^2)}{\sin[(\Omega - \mu_{\lambda_1})T/2]\sin[(\Omega + \mu_{\lambda_1})T/2](\Omega^2 - \mu_{\lambda_2}^2)}\right|\right| . \tag{4.18}$$

Eq. (4.18) indicates that the value of p diverges at times when the argument of the logarithm of the denominator in Eq. (4.18) equals to ± 1. At these times, h_{λ_1} and h_{λ_2} of Eq. (2.19) become identical, and the ratio of A_{λ_1} to A_{λ_2} remains constant independent of the number of iterations p. On the other hand, the number of iterations p vanishes when the denominator becomes infinite. These cases correspond to either h_{λ_1} or h_{λ_2} being zero, or equivalently either A_{λ_1} or A_{λ_2} vanishing after one iteration [Eq. (4.10)]. This argument is not valid in general because of the neglect of other modes ($\lambda_3, \lambda_4, \dots$) excited by the external force. In fact $h_{\lambda_3}, h_{\lambda_4}, \dots$ take finite values at T.

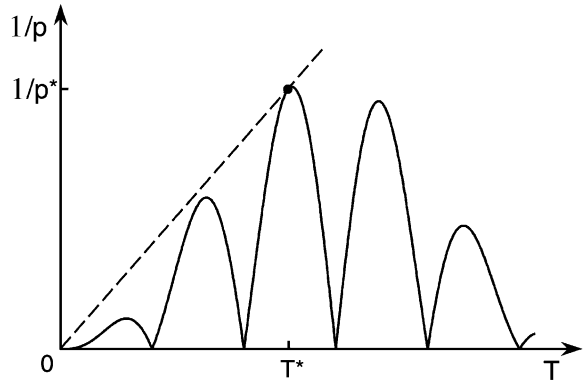


Fig. 4.1. Schematic plot for $1/p$ versus T for a fixed δ value. The solid line shows the fixed-δ curve and the dashed straight line has the largest inclination among lines intersecting the solid curve. The intersecting point provides the optimum time interval T^*.

When taking into account every excited mode, the value of p required for the given δ is schematically plotted as a function of the time interval T in Fig. 4.1 (the fixed-δ curve). Since the computing time varies as pT, we can draw a straight line for a fixed computing time in the $T - 1/p$ plane passing through the origin. The optimum time interval T is given by the point at which the fixed-δ curve intersects the straight line with the maximum slope (corresponding to the minimum computing time).

It is not practical to calculate all eigenvalues of the system in order to obtain p for fixed δ as a function of T. The fixed-δ curve, however, has some common features. At first, this curve rapidly oscillates with a period of the order of π/Ω, as mentioned below Eq. (4.18). The second is that the envelope of this curve slowly oscillates with a period of $\pi/\Delta\mu$ (see Fig. 4.1). Therefore, the optimum value of T is in the vicinity of T_0 given by

$$\begin{aligned} T_0 &= \frac{\pi}{2\Delta\mu} \\ &= \pi N \mathscr{D}(\varepsilon)\sqrt{\varepsilon + \varepsilon_0}\ . \end{aligned} \tag{4.19}$$

In actual calculations, we must find the optimum time-interval $\tilde{T}^*$ around T_0, at which δ becomes a minimum. The time interval $\tilde{T}^*$ is not the true optimum parameter, but is a good approximation for our purpose.

It should be noted that calculations done by using different values of T do not always give the same eigenvector, even though the calculated eigenvectors have the same degree of purity. This is due to the oscillating property of the function $h(\Omega, \mu_\lambda, T)$. Even if $|A_{\lambda_2}/A_{\lambda_1}|$ takes the same values, it depends on T which mode belongs to the main mode λ_1. For large T, the resonance width of $h(\Omega, \mu_\lambda, T)$ becomes narrow, and two modes with frequencies closest to Ω are extracted alternatively with a period of π/Ω. For example, if we obtain an eigenvector $\boldsymbol{e}_{\lambda_1}$ by using T^* shown in Fig. 4.1, the same eigenvector $\boldsymbol{e}_{\lambda_1}$ will be calculated by choosing other values of T within the peak containing T^*. On the contrary, if one chooses T in the adjacent peak, another eigenvector $\boldsymbol{e}_{\lambda_2}$ with the frequency closest to Ω will be extracted.

To summarize this subsection, it should be emphasized that (i) we can compute quite accurately a specific eigenvalue and its eigenvector even for that in the central energy range, and (ii) we can determine the degree of accuracy of eigenvalues and evaluate the purity of eigenvectors.

4.4. Codings for actual computations

In this subsection, we give Fortran source codes for the calculations of the spectral density and an eigenvector of a general type of large matrix. Although a real symmetric matrix is illustrated, the extension to Hermitian matrices with complex elements is straightforward. In Section 8, we describe in detail how the FOM is extended for the case of non-Hermitian matrices.

Let us consider the following $N \times N$ sparse matrix:

$$D_{mn} = \begin{cases} a_m & \text{for } m = n \ , \\ b_m & \text{for } m = n \pm 1 \ , \\ c_m & \text{for } m = n \pm L \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{4.20}$$

where a_m, b_m, and c_m are real and $N = L^2$. This type of matrix appears often in physical problems, such as the lattice vibrations of two-dimensional (2D) systems and the 2D tight-binding model of electrons. In order to eliminate the peculiarity of the sample matrix, we set a_m, b_m, and c_m to be random variables distributed uniformly within the range $[-1, 1]$.

4.4.1. Spectral density

A Fortran source code to compute the spectral density of the matrix $\{D_{mn}\}$ defined by Eq. (4.20) is presented in Fig. 4.2. This program calculates spectral densities at NE (defined in line 24) points in a region covering all eigenvalues. The resolution $\delta\varepsilon$ (DE in the program) of the spectral density is RES_FACT times of the width of the eigenvalue space division (lines 25 and 74). From lines 41 to 55, matrix elements are defined. Since this matrix has negative eigenvalues, we should modify the original matrix $\{D_{mn}\}$ to $\{D'_{mn}\}$ as described by Eq. (2.3). According to the Gerschgorin's theorem [25,26], all eigenvalues of the matrix $\{D_{mn}\}$ are involved in the region which is the union of closed sets, defined by

$$S = \bigcup_{m=1}^{N} S_m \ , \tag{4.21}$$

where

$$S_m = \left\{ \varepsilon \,\middle|\, |\varepsilon - D_{mm}| \leq \sum_{n \neq m}^{N} |D_{mn}| \right\} . \tag{4.22}$$

In this program, the lower and the upper bounds of eigenvalues are estimated by this theorem (lines 59–66). The quantity ε_0 appearing in Eq. (2.3) is chosen to be a sum between the negative sign of this lower bound and an additional positive quantity E_SHIFT (lines 26 and 67). If E_SHIFT $= 0$, the lower bound of eigenvalues of the modified matrix $\{D'_{mn}\}$ becomes zero and the spectral density has poor accuracy near the lower bound, because $\mathscr{D}(\varepsilon)$ given by Eq. (2.15) is proportional to

```
 1  C SAMPLE PROGRAM FOR SPECTRAL DENSITY BY THE FOM
 2  C
 3        PARAMETER (L=1000,NPMAX=1024,N=L*L,N2=N+1,N3=N+L)
 4        PARAMETER (NW=N3+L)
 5        IMPLICIT REAL*8 (A-H,O-Z)
 6        REAL*8 MU,MU_MAX,LG2INV
 7        DIMENSION A(N),B(0:N),C(-L+1:N)
 8        DIMENSION A1(N),B1(0:N),C1(-L+1:N)
 9        DIMENSION X(-L+1:N3),Z(-L+1:N3)
10        DIMENSION F(-L+1:N3)
11        DIMENSION F0(-L+1:N3),F1(-L+1:N3),F2(-L+1:N3)
12        DIMENSION COEF_LF(NPMAX),COEF_LFD(NPMAX)
13        CHARACTER*40 FILENAME
14  C
15        DATA A    /N*0.D0/, B   /N2*0.D0/, C   /N3*0.D0/
16        DATA F    /NW*0.D0/
17        DATA ZERO /0.0D0/
18        DATA ONE  /1.0D0/
19        DATA TWO  /2.0D0/
20        DATA PI   /3.141592653589793D0/
21  C
22  C << Set parameters >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
23  C
24        NE=100                ! Discretization number in eigenvalue space
25        RES_FACT=3.0D0        ! Resolution factor
26        E_SHIFT=1.0D0         ! Additional eigenvalue shift
27        CHI=1.0D0             ! Parameter for Max order of Chebyshev
28        CONV=1.0D-10          ! Convergence parameter for Chebyshev exp.
29        IX=2139127681         ! Seed of random number
30        FEXT=1.0D0            ! Amplitude of external force
31        FILENAME='spectral_density.data'    ! Output file name
32  C
33  C << Sizes of matrix >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
34  C
35        NN1=N                 ! Number of matrix element A
36        NN2=N-1               ! Number of matrix element B
37        NN3=N-L               ! Number of matrix element C
38  C
39  C << Definition of matrix >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
40  C
41        DO 100 I=1,NN1
42           CALL RANDOM(IX,R)
43           RND=TWO*R-ONE
44           A(I)=RND              ! Value of matrix element A
45    100 CONTINUE
46        DO 110 I=1,NN2
47           CALL RANDOM(IX,R)
48           RND=TWO*R-ONE
49           B(I)=RND              ! Value of matrix element B
50    110 CONTINUE
51        DO 120 I=1,NN3
52           CALL RANDOM(IX,R)
53           RND=TWO*R-ONE
54           C(I)=RND              ! Value of matrix element C
55    120 CONTINUE
56  C
57  C << Parameters for calculations >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
58  C
59        E_MIN0=1.0D10         ! Eigenvalue lower bound of original matrix
60        E_MAX0=-1.0D10        ! Eigenvalue upper bound of original matrix
61        DO 200 I=1,NN1        ! Gerschgorin's theorem
62           R1=A(I) -ABS(B(I))-ABS(B(I-1))-ABS(C(I))-ABS(C(I-L))
63           R2=A(I) +ABS(B(I))+ABS(B(I-1))+ABS(C(I))+ABS(C(I-L))
64           IF(R1.LT.E_MIN0) E_MIN0=R1
65           IF(R2.GT.E_MAX0) E_MAX0=R2
66    200 CONTINUE
67        E0=-E_MIN0+E_SHIFT    ! Total energy shift
```

Fig. 4.2. Fortran source code for calculation of the spectral density of the matrix defined by Eq. (4.20). The leftmost number of each line is labeled for convenience to explain the program.

```
      E_MIN=E_MIN0+E0       ! Eigenvalue lower bound of modified matrix
      E_MAX=E_MAX0+E0       ! Eigenvalue upper bound of modified matrix
      DEL_E=E_MAX-E_MIN     ! Auxiliary parameter
      AVE_E=E_MAX+E_MIN     ! Auxiliary parameter
      MU_MAX=SQRT(E_MAX)    ! Freq. corresponding to E_MAX
      MU_MIN=SQRT(E_MIN)    ! Freq. corresponding to E_MIN
      DE=RES_FACT*DEL_E/NE  ! Resolution of spec. density
C
C << Reforming matrix to one with positive eigenvalues >>>>>>>>>>>>>>>>
C
      DO 210 I=1,NN1
         A(I)=A(I)+E0       ! Shifted value of matrix element A
  210 CONTINUE
C
C << Reforming matrix for Chebyshev expansion >>>>>>>>>>>>>>>>>>>>>>>>>
C
      D_EINV=ONE/DEL_E
      DO 220 I=1,NN1
         A1(I)=(TWO*A(I)-AVE_E)*D_EINV
         B1(I)= TWO*B(I)*D_EINV
         C1(I)= TWO*C(I)*D_EINV
  220 CONTINUE
C
C << Definition of external force >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
      DO 300 I=1,NN1
         CALL RANDOM(IX,R)
         RND=TWO*PI*R
         F(I)=FEXT*COS(RND)          ! External force
  300 CONTINUE
C
C << Forced Oscillator Method >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
      OPEN(10,FILE=FILENAME)
      RNEINV=DEL_E/NE
      DEINV=8.0D0*PI/DE
      LG2INV=ONE/LOG(2.0D0)
      DO 1000 IE=1,NE
         E=E_MIN+IE*RNEINV
         MU=SQRT(E)                      ! Frequency of external force
         T=MU*DEINV                      ! Time interval
         NP=MAX(10,INT(CHI*T*(MU_MAX-MU_MIN))) ! Order of T_p for T
         NP=2**(INT(LOG(DBLE(NP+1))*LG2INV)+1) ! NP --> NP=2**NL
         DO 400 I=-L+1,N3                ! Initialization
            X(I)=ZERO                    ! Displacement
            Z(I)=ZERO                    ! Velocity
  400    CONTINUE
C
         CALL COEF_L(MU,T,NP,NPMAX,E_MAX,E_MIN,CONV,COEF_LF,COEF_LFD)
         CALL FEM_L(A1,B1,C1,F,X,Z,L,N,N3,NP,NPMAX,
     &              COEF_LF,COEF_LFD,F0,F1,F2)
C
         E_KIN=ZERO                       ! Kinetic energy
         E_POT=ZERO                       ! Potential energy
         DO 600 I=1,NN1
            E_KIN=E_KIN+Z(I)*Z(I)
            E_POT=E_POT+( A(I)*X(I)+
     &                    B(I)*X(I+1)+B(I-1)*X(I-1)+
     &                    C(I)*X(I+L)+C(I-L)*X(I-L)  )*X(I)
  600    CONTINUE
C
C << Output >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
C     Total energy
         E_TOT=0.5D0*(E_KIN+E_POT)
C     Variable in original eigenvalue space
         E_ORG=E-E0
C     Spectral density as a function of MU
         SD_MU=8.D0*E_TOT/(PI*T*NN1*FEXT*FEXT)
```

Fig. 4.2. Continued.

```
137    C     Spectral density as a function of E_ORG
138             SD_E=0.5D0*SD_MU/MU
139             WRITE(10,2000) E_ORG,SD_E
140     1000 CONTINUE
141          CLOSE (10)
142     2000 FORMAT(D19.14,3X,D19.14)
143    C
144          STOP
145          END
```

Fig. 4.2. Continued.

$1/\sqrt{\varepsilon + \varepsilon_0} = 1/\mu$ (line 138). The modified matrix $\{D'_{mn}\}$ and the external force are defined in lines 78–80 and lines 93–97, respectively. The FOM for the spectral density starts from the line 101. The time interval T is chosen in accordance with Eq. (4.16) (line 108). The energy-function calculation is done in lines 120–127.

The time evolution by the FEM is calculated by two subroutines COEF_L (line 116) and FEM_L (lines 117–118). The subroutine COEF_L provides coefficients $c_p(t)$ and $\dot{c}_p(t)$ (represented by the arrays COEF_LF and COEF_LFD, respectively, in the program) given by Eqs. (3.21) and (3.26). Since $\boldsymbol{x}_0 = 0$ and $\boldsymbol{z}_0 = 0$ in the calculation of the spectral density, coefficients other than $c_p(t)$ and $\dot{c}_p(t)$ are not necessary. The computation of $\boldsymbol{x}(t)$ and $\boldsymbol{z}(t)$ using these coefficients are performed in the subroutine FEM_L. Details of these subroutines will be explained in Section 4.4.3.

We should mention here the cutoff parameter P. As given by Eq. (3.32), P is estimated as $P = \chi(\sqrt{\varepsilon_{\max}} - \sqrt{\varepsilon_{\min}})T$. Since the time interval T is an increasing function of ε as expressed by Eq. (4.16), the maximum value $P_{\max}$ of the cutoff parameter is $8\pi\chi(\varepsilon_{\max} - \sqrt{\varepsilon_{\max}\varepsilon_{\min}})/\delta\varepsilon$, where χ, $\varepsilon_{\min}$, and $\varepsilon_{\max}$ correspond to CHI (line 27), E_MIN (line 68), and E_MAX (line 69) in the program, respectively. Considering the definition of $\delta\varepsilon$ in this sample, i.e., $\delta\varepsilon = \mathrm{DE} = \mathrm{RES\_FACT} \cdot (\varepsilon_{\max} - \varepsilon_{\min})/\mathrm{NE}$, $P_{\max}$ is written as

$$P_{\max} = \frac{8\pi\chi \cdot \mathrm{NE}\sqrt{\varepsilon_{\max}}}{\mathrm{RES\_FACT} \cdot (\sqrt{\varepsilon_{\max}} + \sqrt{\varepsilon_{\min}})} \,. \tag{4.23}$$

The size of the arrays COEF_LF and COEF_LFD is at least larger than $P_{\max}$. As will be mentioned in Section 4.4.3, coefficients $c_p(t)$ and $\dot{c}_p(t)$ are calculated by using the fast Fourier transform (FFT) technique. Thus, the size of these arrays should be a number such as 2^m (m is a certain integer). The least value of m satisfying $2^m \geq P_{\max}$ can be evaluated from Eq. (4.23) (line 110). However, at the stage of the dimension declaration in the program, values of $\varepsilon_{\max}$ and $\varepsilon_{\min}$ (then $P_{\max}$ also) are still unknown. Therefore, we should calculate these parameters, at first, for the given matrix by the Gerschgorin's theorem [25,26], and then determine the required size of the arrays COEF_LF and COEF_LFD (NPMAX in line 3).

The most time-consuming part in this program is in the subroutine FEM_L. The computing time for FEM_L is proportional to $\mathrm{NE}^2 \cdot \mathrm{N}/\mathrm{RES\_FACT}$. In the case that $N = 10^6$ ($\mathrm{L} = 1000$), $\mathrm{NE} = 100$, and $\mathrm{RES\_FACT} = 3$, for example, it totally takes 585 s to obtain the spectral density by using the FACOM VPP500 (the program is executed in a single processor). The required memory space is proportional to $N + \sqrt{N}$ in this program. In order to avoid the use of IF statements in the deepest loop (lines 31–33 of the subroutine FEM_L shown in Fig. 4.7), the sizes of some arrays

(B, C, X, Z, etc.) are larger than the minimum required. If we do not use this trick, the required memory is simply proportional to N.

4.4.2. Eigenvalues and eigenvectors

A sample program to compute an eigenvalue and its eigenvector is presented in Fig. 4.3. A part of the program (lines 43–83 and 96–115) to define matrix elements and amplitudes of the external force and to modify the matrix is the same with that for the spectral density (lines 35–73 and 78–97 in Fig. 4.2). In this program, an eigenvalue closest to $\mathrm{E} = 0.2$ (line 30) and the corresponding eigenvector are calculated. The value of the spectral density at $\mathrm{E} = 0.2$ given in the line 31 is assumed to have been already calculated by the FOM for the spectral density (Fig. 4.2). The program repeats iterations (the number of iterations is bounded by NIT_MAX defined in the line 33) until the quantity $|A_{\lambda_2}/A_{\lambda_1}|$ given by Eq. (4.12) becomes less than PURITY defined in the line 32. The iteration procedure starts from the line 125. For an efficient calculation, the time interval in each iteration should be chosen to be a value close to $T_0 = \pi/2\Delta\mu$ [see Eq. (4.19)], at which δ becomes a local minimum as a function of time. In order to find this time interval, the time development in each iteration is divided into two parts. In the first part (lines 133–134), the FEM gives $\boldsymbol{x}(t)$ and $\boldsymbol{z}(t)$ at the time $T_0 = \pi/2\Delta\mu$ (TO in line 86). In the second part (lines 155–182), the time-evolution calculation is continued up to TO + TS with a small time step TDIV (line 88), where TS (line 87) is half the period of the external force and the quantity δ at each time step is calculated (lines 163–181). The optimum time interval $\tilde{T}^*$ [see below Eq. (4.19)] in each iteration is the time at which δ becomes minimum in the range $\mathrm{TO} < t < \mathrm{TO}+\mathrm{TS}$ (line 177). If $|A_{\lambda_2}/A_{\lambda_1}|$ at $t = \tilde{T}^*$ is smaller than PURITY, the program quits the iteration and jump to the output routine (line 186). Otherwise, the iteration procedure is repeated with new external forces (lines 187–194).

In the subroutines COEF_L and FEM_L, the time interval is set to be T_0 given by Eq. (4.19). From Eq. (3.32) and $\varepsilon_{\min} = 0$ (because of $\mathrm{E\_SHIFT} = 0$) in the eigenvector calculation, the maximum value of the cutoff parameter $P_{\max}$ for these subroutines is expressed by

$$P_{\max} = \pi N \chi \mathscr{D}(\varepsilon)\sqrt{\varepsilon_{\max}(\varepsilon + \varepsilon_0)}\ . \tag{4.24}$$

As in the case of the spectral density, the arrays COEF_LF and COEF_LFD should be declarated with a size of 2^m (due to the FFT) larger than $P_{\max}$ given by Eq. (4.24) (NPLMAX in lines 3 and 14). The time interval in the subroutines COEF_S and FEM_S is quite short (TDIV in line 88). The value of $P_{\max}$ for these subroutines is also estimated by Eq. (3.32) as

$$P_{\max} = \chi\sqrt{\varepsilon_{\max}}\cdot \mathrm{TDIV}\ , \tag{4.25}$$

and the size of the arrays COEF_SF, COEF_SXD, COEF_SZ, COEF_SF, and COEF_SFD is determined by Eq. (4.25) (NPSMAX in line 3 and lines 15–17).

It should be noted that the FOM for eigenvectors and eigenvalues is an $O(N^2)$ method, while the FOM for the spectral density is an $O(N)$ method as mentioned in Section 4.2. This is because the computing time of the FEM is proportional to NT_0 as mentioned below Eq. (3.32) and $T_0 \propto N$ [see Eq. (4.19)]. If we perform this program with $\mathrm{L} = 500$ and $\mathrm{E} = 0.2$ in a single processor of the FACOM VPP500, it takes about 3300 s per iteration. The total computing time is given by multiplying this time by the number of iterations, which is usually less than 10.

```
 1   C SAMPLE PROGRAM FOR EIGENVALUES AND EIGENVECTORS BY THE FOM
 2   C
 3         PARAMETER (L=500,NPLMAX=2097152,NPSMAX=16,N=L*L,N2=N+1,N3=N+L)
 4         PARAMETER (NW=N3+L)
 5         IMPLICIT REAL*8 (A-H,O-Z)
 6         REAL*8 MU,MU_MAX
 7         DIMENSION A(N),B(0:N),C(-L+1:N)
 8         DIMENSION A1(N),B1(0:N),C1(-L+1:N)
 9         DIMENSION X(-L+1:N3),Z(-L+1:N3)
10         DIMENSION F(-L+1:N3)
11         DIMENSION X0(-L+1:N3),X1(-L+1:N3),X2(-L+1:N3)
12         DIMENSION Z0(-L+1:N3),Z1(-L+1:N3),Z2(-L+1:N3)
13         DIMENSION F0(-L+1:N3),F1(-L+1:N3),F2(-L+1:N3)
14         DIMENSION COEF_LF(NPLMAX),COEF_LFD(NPLMAX)
15         DIMENSION COEF_SX(NPSMAX),COEF_SXD(NPSMAX)
16         DIMENSION COEF_SZ(NPSMAX)
17         DIMENSION COEF_SF(NPSMAX),COEF_SFD(NPSMAX)
18         CHARACTER*40 FILENAME
19   C
20         DATA A    /N*0.0D0/, B  /N2*0.0D0/, C  /N3*0.0D0/
21         DATA A1   /N*0.0D0/, B1 /N2*0.0D0/, C1 /N3*0.0D0/
22         DATA F    /NW*0.0D0/
23         DATA ZERO /0.0D0/
24         DATA ONE  /1.0D0/
25         DATA TWO  /2.0D0/
26         DATA PI   /3.141592653589793D0/
27   C
28   C << Set parameters >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
29   C
30         E=0.2D0                    ! Calculate the eigenvalue closest to E
31         SD=0.23D0                  ! Spectral density at E
32         PURITY=1.0D-3              ! Criterion for the conversion
33         NIT_MAX=50                 ! Maximum number of iterations
34         NTS=20                     ! Number of suppl. time int. division
35         CHI=1.0D0                  ! Parameter for Max order of Chebyshev
36         CONV=1.0D-10               ! Convergence param. for Chebyshev exp.
37         IX=2139127681              ! Seed of random number
38         FEXT=1.0D0                 ! Amplitude of external force
39         FILENAME='eigenvector.data'       ! Output file name
40   C
41   C << Sizes of matrix >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
42   C
43         NN1=N                      ! Number of matrix element A
44         NN2=N-1                    ! Number of matrix element B
45         NN3=N-L                    ! Number of matrix element C
46   C
47   C << Definition of matrix >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
48   C
49         DO 100 I=1,NN1
50            CALL RANDOM(IX,R)
51            RND=TWO*R-ONE
52            A(I)=RND                ! Value of matrix element A
53     100 CONTINUE
54         DO 110 I=1,NN2
55            CALL RANDOM(IX,R)
56            RND=TWO*R-ONE
57            B(I)=RND                ! Value of matrix element B
58     110 CONTINUE
59         DO 120 I=1,NN3
60            CALL RANDOM(IX,R)
61            RND=TWO*R-ONE
62            C(I)=RND                ! Value of matrix element C
63     120 CONTINUE
64   C
65   C << Parameters for calculations >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
66   C
67         E_MIN0=1.0D10          ! Eigenvalue lower bound of original matrix
68         E_MAX0=-1.0D10         ! Eigenvalue upper bound of original matrix
```

Fig. 4.3. Fortran source code for calculation of the eigenvalue and eigenvector of the matrix defined by Eq. (4.20). The leftmost number of each line is labeled for convenience to explain the program.

```
      DO 200 I=1,NN1        ! Gerschgorin's theorem
         R1=A(I) -ABS(B(I))-ABS(B(I-1))-ABS(C(I))-ABS(C(I-L))
         R2=A(I) +ABS(B(I))+ABS(B(I-1))+ABS(C(I))+ABS(C(I-L))
         IF(R1.LT.E_MIN0) E_MIN0=R1
         IF(R2.GT.E_MAX0) E_MAX0=R2
  200 CONTINUE
C
      E0=-E_MIN0                      ! Eigenvalue shift
      E_MIN=E_MIN0+E0                 ! Lower bound of modified matrix
      E_MAX=E_MAX0+E0                 ! Upper bound of modified matrix
      DEL_E=E_MAX-E_MIN               ! Auxiliary parameter
      AVE_E=E_MAX+E_MIN               ! Auxiliary parameter
      MU=SQRT(E+E0)                   ! Frequency of external force
      MU_MAX=SQRT(E_MAX)              ! Freq. corresponding to E_MAX
      MU_MIN=SQRT(E_MIN)              ! Freq. corresponding to E_MIN
      SD_MU=TWO*MU*SD                 ! SD at frequency MU
      DEL_MU=ONE/(NN1*SD_MU)          ! Eigenfrequency spacing
      T0=0.5D0*PI/DEL_MU              ! Initial time interval
      TS=1.1D0*PI/MU                  ! Supplemental time interval
      TDIV=TS/NTS                     ! Time step in TS
      NPL=MAX(10,INT(CHI*T0*(MU_MAX-MU_MIN)))  ! Order of T_p for T
      NPS=MAX(10,INT(CHI*TDIV*(MU_MAX-MU_MIN)))! Order of T_p for TDIV
      NPL=2**(INT(LOG(DBLE(NPL+1))/LOG(2.0D0))+1)   ! NPL --> NPL=2**NL
      NPS=2**(INT(LOG(DBLE(NPS+1))/LOG(2.0D0))+1)   ! NPS --> NPS=2**NS
C
C << Reforming matrix to one with positive eigenvalues >>>>>>>>>>>>>>>>>
C
      DO 210 I=1,NN1
         A(I)=A(I)+E0       ! Shifted value of matrix element A
  210 CONTINUE
C
C << Reforming matrix for Chebyshev expansion >>>>>>>>>>>>>>>>>>>>>>>>>>
C
      D_EINV=ONE/DEL_E
      DO 220 I=1,NN1
         A1(I)=(TWO*A(I)-AVE_E)*D_EINV
         B1(I)= TWO*B(I)*D_EINV
         C1(I)= TWO*C(I)*D_EINV
  220 CONTINUE
C
C << Definition of external force >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
      DO 300 I=1,NN1
         CALL RANDOM(IX,R)
         RND=TWO*PI*R
         F(I)=FEXT*COS(RND)           ! External force
  300 CONTINUE
C
C << Coefficients of Chebyshev expansion >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
      CALL COEF_L(MU,T0,NPL,NPLMAX,E_MAX,E_MIN,CONV,COEF_LF,COEF_LFD)
      CALL COEF_S(MU,TDIV,NPS,NPSMAX,E_MAX,E_MIN,CONV,
     &           COEF_SX,COEF_SXD,COEF_SZ,COEF_SF,COEF_SFD)
C
C << Iteration for Forced Oscillator Method >>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
      D_MINV=ONE/DEL_MU
      DO 1000 IIT=1,NIT_MAX           ! Iteration loop
         DO 400 I=-L+1,N3             ! Initialization
            X(I)=ZERO                 ! Displacement at t
            Z(I)=ZERO                 ! Velocity at t
 400     CONTINUE
C
C  Long time evolution
         CALL FEM_L(A1,B1,C1,F,X,Z,L,N,N3,NPL,NPLMAX,
     &              COEF_LF,COEF_LFD,F0,F1,F2)
C
C  Purity evaluation
         GAMMA0=ZERO
         GAMMA2=ZERO
         GAMMA4=ZERO
         DO 500 I=1,NN1
            AI=A(I)*X(I)+
```

Fig. 4.3. Continued.

```
     &         B(I)*X(I+1)+B(I-1)*X(I-1)+
     &         C(I)*X(I+L)+C(I-L)*X(I-L)
            BI=X(I)
            GAMMA0=GAMMA0+BI*BI
            GAMMA2=GAMMA2+AI*BI
            GAMMA4=GAMMA4+AI*AI
  500    CONTINUE
         DELTA1=SQRT(ABS(ONE-GAMMA2*GAMMA2/(GAMMA4*GAMMA0)))
         EIGN_MU1=SQRT(ABS(GAMMA2/GAMMA0))
         DELTA2=DELTA1*1.01D0
         EIGN_MU2=EIGN_MU1
C
C
         DO 600 ITS=1,NTS          ! Supplemental time interval loop
C
C  Short time evolution
            CALL FEM_S(A1,B1,C1,F,X,Z,L,N,N3,NPS,NPSMAX,
     &                COEF_SX,COEF_SXD,COEF_SZ,COEF_SF,COEF_SFD,
     &                X0,X1,X2,Z0,Z1,Z2,F0,F1,F2)
C
C  Purity evaluation
            GAMMA0=ZERO
            GAMMA2=ZERO
            GAMMA4=ZERO
            DO 610 I=1,NN1
              AI=A(I)*X(I)+
     &           B(I)*X(I+1)+B(I-1)*X(I-1)+
     &           C(I)*X(I+L)+C(I-L)*X(I-L)
              BI=X(I)
              GAMMA0=GAMMA0+BI*BI
              GAMMA2=GAMMA2+AI*BI
              GAMMA4=GAMMA4+AI*AI
  610       CONTINUE
            DELTA3=SQRT(ABS(ONE-GAMMA2*GAMMA2/(GAMMA4*GAMMA0)))
            EIGN_MU3=SQRT(ABS(GAMMA2/GAMMA0))
            IF(DELTA1.GT.DELTA2.AND.DELTA3.GT.DELTA2) GOTO 700
            DELTA1=DELTA2
            DELTA2=DELTA3
            EIGN_MU1=EIGN_MU2
            EIGN_MU2=EIGN_MU3
  600    CONTINUE                  ! End of supplemental time interval
C
  700    RATIO=0.5D0*EIGN_MU2*DELTA2*D_MINV     ! A2/A1
         write(6,5100) iit,ratio
         IF(RATIO.LT.PURITY) GOTO 2000
         X_MAX=ZERO
         DO 800 I=1,NN1
            IF(X_MAX.LT.ABS(X(I))) X_MAX=X(I)   ! Rescale X1
  800    CONTINUE
         X_MINV=FEXT/X_MAX
         DO 810 I=1,NN1
            F(I)=X(I)*X_MINV                    ! Ext. force iteration
  810    CONTINUE
 1000 CONTINUE                                  ! End of iteration
C
C << Output >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
 2000 WRITE(6,5200) RATIO,EIGN_MU2*EIGN_MU2-E0
      V_NOR=ZERO
      DO 900 I=1,NN1
         V_NOR=V_NOR+X(I)*X(I)                  ! Normalization
  900 CONTINUE
      SQVINV=ONE/SQRT(V_NOR)
      OPEN(10,FILE=FILENAME)
      DO 910 I=1,NN1
         WRITE(10,5300) I,X(I)*SQVINV
  910 CONTINUE
      CLOSE (10)
 5100 FORMAT("Number of Iterations=",I3,4X,"A2/A1=",D19.14)
 5200 FORMAT("a2/a1=",D19.14,4X,"Eigenvalue=",D19.14)
 5300 FORMAT(I6,3X,D19.14)
C
      STOP
      END
```

Fig. 4.3. Continued.

4.4.3. Time evolution routine

The time evolution based on the FEM is separated from the main program as subroutines. The time evolution routine consists of two subroutines. One (COEF_L or COEF_S) is for the calculation of coefficients of the Chebyshev expansion and the other (FEM_L or FEM_S) gives $\boldsymbol{x}(t)$ and $\boldsymbol{z}(t)$ from these coefficients. The subroutines COEF_L and FEM_L are provided for a time evolution starting with the initial conditions of $\boldsymbol{x}(0) = 0$ and $\boldsymbol{z}(0) = 0$, while COEF_S and FEM_S are for $\boldsymbol{x}(0) \neq 0$ and $\boldsymbol{z}(0) \neq 0$. (Indices L and S stand for *long* and *short* time intervals, respectively.) Since the latter set of subroutines includes the former one as a special case, we explain only the subroutines COEF_S and FEM_S (Figs. 4.4–4.9).

The subroutine COEF_S to calculate the Chebyshev expansion coefficients is shown in Fig. 4.4. Variables COEF_SX, COEF_SZ, COEF_SF, COEF_SXD, and COEF_SFD correspond to $a_p(t)$, $b_p(t)$, $c_p(t)$, $\dot{a}_p(t)$, and $\dot{c}_p(t)$ given by Eqs. (3.19)–(3.21), (3.24), and (3.26), respectively. These coefficients are computed by the FFT. Transformed functions are calculated in lines 22–33. The quantity ξ_q defined by Eq. (3.22) is given in the line 23 (XIQ). Trigonometric calculations in lines 16–21 and 30–32 are the trick for fast computations of trigonometric functions. This trick is based on the recurrence [1],

$$\cos(\theta + \varphi) = \cos\theta - \alpha\cos\theta - \beta\sin\theta \,,$$

$$\sin(\theta + \varphi) = \sin\theta - \alpha\sin\theta + \beta\cos\theta \,, \tag{4.26}$$

where

$$\alpha = 2\sin^2\left(\frac{\varphi}{2}\right) \tag{4.27}$$

and

$$\beta = \sin\varphi \,. \tag{4.28}$$

The Fourier transform of these functions are performed by the subroutine FFT (lines 34–38). The subroutine FFT is not a standard FFT routine, but provides the Fourier transform given by Eq. (3.29). The program list of FFT is presented in Fig. 4.8, which is essentially the same with the program given in Ref. [1]. According to Eqs. (3.19)–(3.21) and (3.24)–(3.26), we obtain the final results for the Chebyshev expansion coefficients by multiplying these Fourier transforms by $2/(1 + \delta_{p0})P$ (lines 39–55). The actually required order of the Chebyshev expansion [the cutoff parameter P in Eq. (3.15) or (3.23)] is determined in lines 52–54. The convergence criterion (tolerance) CONV is given in the line 28 of Fig. 4.2 or the line 36 of Fig. 4.3.

The subroutine FEM_S is shown in Fig. 4.6. In this subroutine, we calculate $\boldsymbol{x}(t)$ and $\boldsymbol{z}(t)$ by Eqs. (3.15) and (3.23), respectively. Products between the Chebyshev polynomial of the matrix and a vector are calculated by using the recurrence formula Eq. (3.28). Arrays X0, X1, and X2 provide three terms $\boldsymbol{x}_{p-1}, \boldsymbol{x}_p$, and $\boldsymbol{x}_{p+1}$ in Eq. (3.28), respectively. Other arrays (Z0, F0, ...) present similar variables $(\boldsymbol{z}_{p-1}, \boldsymbol{f}_{p-1}, \dots)$. Eq. (3.28a) is calculated in lines 29–42, while Eq. (3.28b) is evaluated in lines 52–71. Summations in Eqs. (3.15) and (3.23) for $p = 0$ and 1 are performed in lines 43–50. The rest of the summations is done in lines 72–79. The most time-consuming part which governs the efficiency of the total program lies in lines 54–62 and 73–78. The computing time of this part is proportional to NP.

```
      SUBROUTINE COEF_S(MU,T,NP,NPMAX,E_MAX,E_MIN,CONV,
     &           COEF_SX,COEF_SXD,COEF_SZ,COEF_SF,COEF_SFD)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MU
      DIMENSION COEF_SX(NPMAX),COEF_SXD(NPMAX)
      DIMENSION COEF_SZ(NPMAX)
      DIMENSION COEF_SF(NPMAX),COEF_SFD(NPMAX)
C
      DATA ONE  /1.0D0/
      DATA TWO  /2.0D0/
      DATA PI   /3.141592653589793D0/
C
      DEL_E=E_MAX-E_MIN
      AVE_E=E_MAX+E_MIN
      ALPHA=-TWO*(SIN(0.5D0*PI/NP))**2
      BETA=SIN(PI/NP)
      COSQ=COS(0.5D0*PI/NP)
      SINQ=SIN(0.5D0*PI/NP)
      COSMUT=COS(MU*T)
      SINMUT=SIN(MU*T)
      DO 100 IP=1,NP
         XIQ=SQRT(DEL_E*((AVE_E/DEL_E)+COSQ)*0.5D0)
         DENOM=XIQ*XIQ-MU*MU
         COEF_SX(IP)=COS(XIQ*T)
         COEF_SZ(IP)=SIN(XIQ*T)/XIQ
         COEF_SF(IP)=(COSMUT-COEF_SX(IP))/DENOM
         COEF_SXD(IP)=-XIQ*XIQ*COEF_SZ(IP)
         COEF_SFD(IP)=(-COEF_SXD(IP)-MU*SINMUT)/DENOM
         COSQ_TMP=COSQ
         COSQ=COSQ+ALPHA*COSQ-BETA*SINQ
         SINQ=SINQ+ALPHA*SINQ+BETA*COSQ_TMP
  100 CONTINUE
      CALL FFT(COEF_SX,NP)
      CALL FFT(COEF_SZ,NP)
      CALL FFT(COEF_SF,NP)
      CALL FFT(COEF_SXD,NP)
      CALL FFT(COEF_SFD,NP)
      FACT1=ONE/NP
      FACT2=TWO/NP
      DO 110 IP=1,NP
         IF(IP.EQ.1) THEN
            FACT=FACT1
         ELSE
            FACT=FACT2
         END IF
         COEF_SX(IP)=FACT*COEF_SX(IP)
         COEF_SZ(IP)=FACT*COEF_SZ(IP)
         COEF_SF(IP)=FACT*COEF_SF(IP)
         COEF_SXD(IP)=FACT*COEF_SXD(IP)
         COEF_SFD(IP)=FACT*COEF_SFD(IP)
         IF(MAX(ABS(COEF_SX(IP)),ABS(COEF_SZ(IP)),ABS(COEF_SF(IP))
     &         ,ABS(COEF_SXD(IP)),ABS(COEF_SFD(IP)))
     &     .LT.CONV) GOTO 200
  110 CONTINUE
C
  200 NP=IP
C
      RETURN
      END
```

Fig. 4.4. The program list of the subroutine COEF_S.

```
      SUBROUTINE COEF_L(MU,T,NP,NPMAX,E_MAX,E_MIN,CONV,
     &                  COEF_LF,COEF_LFD)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MU
      DIMENSION COEF_LF(NPMAX),COEF_LFD(NPMAX)
C
      DATA ONE  /1.0D0/
      DATA TWO  /2.0D0/
      DATA PI   /3.141592653589793D0/
C
      DEL_E=E_MAX-E_MIN
      AVE_E=E_MAX+E_MIN
      ALPHA=-TWO*(SIN(0.5D0*PI/NP))**2
      BETA=SIN(PI/NP)
      COSQ=COS(0.5D0*PI/NP)
      SINQ=SIN(0.5D0*PI/NP)
      COSMUT=COS(MU*T)
      SINMUT=SIN(MU*T)
C
      DO 100 IP=1,NP
         XIQ=SQRT(DEL_E*((AVE_E/DEL_E)+COSQ)*0.5D0)
         DENOM=XIQ*XIQ-MU*MU
         COEF_LF(IP)=(COSMUT-COS(XIQ*T))/DENOM
         COEF_LFD(IP)=(XIQ*SIN(XIQ*T)-MU*SINMUT)/DENOM
         COSQ_TMP=COSQ
         COSQ=COSQ+ALPHA*COSQ-BETA*SINQ
         SINQ=SINQ+ALPHA*SINQ+BETA*COSQ_TMP
  100 CONTINUE
      CALL FFT(COEF_LF,NP)
      CALL FFT(COEF_LFD,NP)
      FACT1=ONE/NP
      FACT2=TWO/NP
      DO 110 IP=1,NP
         IF(IP.EQ.1) THEN
            FACT=FACT1
         ELSE
            FACT=FACT2
         END IF
         COEF_LF(IP)=FACT*COEF_LF(IP)
         COEF_LFD(IP)=FACT*COEF_LFD(IP)
         IF(MAX(ABS(COEF_LF(IP)),ABS(COEF_LFD(IP))).LT.CONV) GOTO 200
  110 CONTINUE
C
  200 NP=IP
C
      RETURN
      END
```

Fig. 4.5. The program list of the subroutine COEF_L.

5. Computing linear response functions for classical systems

The purpose of this section is to show how to compute linear response functions for classical systems in the context of the FOM. The facets of the FOM we have developed in previous sections can be combined into a new scheme for computing linear response functions. In Sections 5–7, we use the symbol ω instead of μ, because it represents an *actual* frequency of a physical system. The FOM makes it possible to compute directly $S(\boldsymbol{q}, \omega)$ for physical systems described by very large matrices, in particular, without performing the Fourier transform of the spatio-temporal correlation function $S(\boldsymbol{r}, t)$ [29]. This scheme is an efficient O(N) method and quite different from those

```
      SUBROUTINE FEM_S(A,B,C,F,X,Z,L,N,N3,NP,NPMAX,
     &                 COEF_SX,COEF_SXD,COEF_SZ,COEF_SF,COEF_SFD,
     &                 X0,X1,X2,Z0,Z1,Z2,F0,F1,F2)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(N),B(0:N),C(-L+1:N)
      DIMENSION X(-L+1:N3),Z(-L+1:N3),F(-L+1:N3)
      DIMENSION X0(-L+1:N3),X1(-L+1:N3),X2(-L+1:N3)
      DIMENSION Z0(-L+1:N3),Z1(-L+1:N3),Z2(-L+1:N3)
      DIMENSION F0(-L+1:N3),F1(-L+1:N3),F2(-L+1:N3)
      DIMENSION COEF_SX(NPMAX),COEF_SXD(NPMAX)
      DIMENSION COEF_SZ(NPMAX)
      DIMENSION COEF_SF(NPMAX),COEF_SFD(NPMAX)
C
      DATA ZERO /0.0D0/
      DATA TWO  /2.0D0/
C
      DO 100 I=-L+1,N3                  ! Initialization
         X0(I)=ZERO                     ! T_{n-2}(D)*X
         X1(I)=ZERO                     ! T_{n-1}(D)*X
         X2(I)=ZERO                     ! T_{n}  (D)*X
         Z0(I)=ZERO                     ! T_{n-2}(D)*Z
         Z1(I)=ZERO                     ! T_{n-1}(D)*Z
         Z2(I)=ZERO                     ! T_{n}  (D)*Z
         F0(I)=ZERO                     ! T_{n-2}(D)*F
         F1(I)=ZERO                     ! T_{n-1}(D)*F
         F2(I)=ZERO                     ! T_{n}  (D)*F
  100 CONTINUE
      DO 110 I=1,N
         X0(I)=X(I)
         Z0(I)=Z(I)
         F0(I)=F(I)
         X1(I)=A(I)*X(I)+
     &         B(I)*X(I+1)+B(I-1)*X(I-1)+
     &         C(I)*X(I+L)+C(I-L)*X(I-L)
         Z1(I)=A(I)*Z(I)+
     &         B(I)*Z(I+1)+B(I-1)*Z(I-1)+
     &         C(I)*Z(I+L)+C(I-L)*Z(I-L)
         F1(I)=A(I)*F(I)+
     &         B(I)*F(I+1)+B(I-1)*F(I-1)+
     &         C(I)*F(I+L)+C(I-L)*F(I-L)
  110 CONTINUE
      DO 120 I=1,N
         X(I)=COEF_SX(1) *X0(I)+COEF_SX(2) *X1(I)+
     &        COEF_SZ(1) *Z0(I)+COEF_SZ(2) *Z1(I)+
     &        COEF_SF(1) *F0(I)+COEF_SF(2) *F1(I)
         Z(I)=COEF_SXD(1)*X0(I)+COEF_SXD(2)*X1(I)+
     &        COEF_SX(1) *Z0(I)+COEF_SX(2) *Z1(I)+
     &        COEF_SFD(1)*F0(I)+COEF_SFD(2)*F1(I)
  120 CONTINUE
C
      DO 200 IP=3,NP
         DO 210 I=1,N
            X2(I)=TWO*(A(I)*X1(I)+
     &                 B(I)*X1(I+1)+B(I-1)*X1(I-1)+
     &                 C(I)*X1(I+L)+C(I-L)*X1(I-L))-X0(I)
            Z2(I)=TWO*(A(I)*Z1(I)+
     &                 B(I)*Z1(I+1)+B(I-1)*Z1(I-1)+
     &                 C(I)*Z1(I+L)+C(I-L)*Z1(I-L))-Z0(I)
            F2(I)=TWO*(A(I)*F1(I)+
     &                 B(I)*F1(I+1)+B(I-1)*F1(I-1)+
     &                 C(I)*F1(I+L)+C(I-L)*F1(I-L))-F0(I)
  210    CONTINUE
         DO 220 I=1,N
            X0(I)=X1(I)
            X1(I)=X2(I)
            Z0(I)=Z1(I)
```

Fig. 4.6. The program list of the subroutine FEM_S.

```
            Z1(I)=Z2(I)
            F0(I)=F1(I)
            F1(I)=F2(I)
  220    CONTINUE
         DO 230 I=1,N
            X(I)=X(I)+COEF_SX(IP) *X2(I)+
     &                COEF_SZ(IP) *Z2(I)+
     &                COEF_SF(IP) *F2(I)
            Z(I)=Z(I)+COEF_SXD(IP)*X2(I)+
     &                COEF_SX(IP) *Z2(I)+
     &                COEF_SFD(IP)*F2(I)
  230    CONTINUE
  200 CONTINUE
C
      RETURN
      END
```

Fig. 4.6. Continued.

```
      SUBROUTINE FEM_L(A,B,C,F,X,Z,L,N,N3,NP,NPMAX,
     &                 COEF_LF,COEF_LFD,F0,F1,F2)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(N),B(0:N),C(-L+1:N)
      DIMENSION X(-L+1:N3),Z(-L+1:N3),F(-L+1:N3)
      DIMENSION F0(-L+1:N3),F1(-L+1:N3),F2(-L+1:N3)
      DIMENSION COEF_LF(NPMAX),COEF_LFD(NPMAX)
C
      DATA ZERO /0.0D0/
      DATA TWO  /2.0D0/
C
      DO 100 I=-L+1,N3                  ! Initialization
         F0(I)=ZERO                     ! T_{n-2}(D)*F
         F1(I)=ZERO                     ! T_{n-1}(D)*F
         F2(I)=ZERO                     ! T_{n}  (D)*F
  100 CONTINUE
      DO 110 I=1,N
         F0(I)=F(I)
         F1(I)=A(I)*F(I)+
     &         B(I)*F(I+1)+B(I-1)*F(I-1)+
     &         C(I)*F(I+L)+C(I-L)*F(I-L)
  110 CONTINUE
      DO 120 I=1,N
         X(I)=COEF_LF(1) *F0(I)+COEF_LF(2) *F1(I)
         Z(I)=COEF_LFD(1)*F0(I)+COEF_LFD(2)*F1(I)
  120 CONTINUE
C
      DO 200 IP=3,NP
         DO 210 I=1,N
            F2(I)=TWO*(A(I)*F1(I)+
     &                 B(I)*F1(I+1)+B(I-1)*F1(I-1)+
     &                 C(I)*F1(I+L)+C(I-L)*F1(I-L))-F0(I)
  210    CONTINUE
         DO 220 I=1,N
            F0(I)=F1(I)
            F1(I)=F2(I)
  220    CONTINUE
         DO 230 I=1,N
            X(I)=X(I)+COEF_LF(IP) *F2(I)
            Z(I)=Z(I)+COEF_LFD(IP)*F2(I)
  230    CONTINUE
  200 CONTINUE
C
      RETURN
      END
```

Fig. 4.7. The program list of the subroutine FEM_L.

```
      SUBROUTINE FFT(Y,N)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(N)
      PI=3.141592653589793D0
      C1=0.5D0
      C2=-0.5D0
      J=1
      MMAX=2
      N2P3=N+3
C
      THETA=0.5D0*PI/N
      WR=COS(THETA)
      WI=SIN(THETA)
      WPR0=-2.0D0*WI**2
      WPI0=SIN(2.0D0*THETA)
      DO 100 I=1,N/2
         Y1=0.5D0*(Y(I)+Y(N-I+1))
         Y2=WI*(Y(I)-Y(N-I+1))
         Y(I)=Y1+Y2
         Y(N-I+1)=Y1-Y2
         WTEMP=WR
         WR=WR*WPR0-WI*WPI0+WR
         WI=WI*WPR0+WTEMP*WPI0+WI
  100 CONTINUE
C
      DO 110 I=1,N,2
         IF(J.GT.I) THEN
           TEMPR=Y(J)
           TEMPI=Y(J+1)
           Y(J)=Y(I)
           Y(J+1)=Y(I+1)
           Y(I)=TEMPR
           Y(I+1)=TEMPI
         ENDIF
         M=N/2
  120    IF((M.GE.2).AND.(J.GT.M)) THEN
            J=J-M
            M=M/2
            GOTO 120
         ENDIF
         J=J+M
  110 CONTINUE
  130 IF(N.GT.MMAX) THEN
        ISTEP=2*MMAX
        THETA=2.D0*PI/(MMAX)
        WPR=-2.D0*SIN(0.5D0*THETA)**2
        WPI=SIN(THETA)
        WR=1.D0
        WI=0.D0
        DO 140 M=1,MMAX,2
           DO 150 I=M,N,ISTEP
              J=I+MMAX
              TEMPR=WR*Y(J)-WI*Y(J+1)
              TEMPI=WR*Y(J+1)+WI*Y(J)
              Y(J)=Y(I)-TEMPR
              Y(J+1)=Y(I+1)-TEMPI
              Y(I)=Y(I)+TEMPR
              Y(I+1)=Y(I+1)+TEMPI
  150      CONTINUE
           WTEMP=WR
           WR=WR*WPR-WI*WPI+WR
           WI=WI*WPR+WTEMP*WPI+WI
  140   CONTINUE
        MMAX=ISTEP
      GOTO 130
      ENDIF
C
      THETA=PI/DBLE(N/2)
      WPR=-2.D0*SIN(0.5D0*THETA)**2
      WPI=SIN(THETA)
      WR=1.D0+WPR
      WI=WPI
      DO 200 I=2,N/4
         I1=2*I-1
         I2=I1+1
         I3=N2P3-I2
         I4=I3+1
         H1R=C1*(Y(I1)+Y(I3))
         H1I=C1*(Y(I2)-Y(I4))
         H2R=-C2*(Y(I2)+Y(I4))
         H2I=C2*(Y(I1)-Y(I3))
         Y(I1)=H1R+WR*H2R-WI*H2I
         Y(I2)=H1I+WR*H2I+WI*H2R
         Y(I3)=H1R-WR*H2R+WI*H2I
         Y(I4)=-H1I+WR*H2I+WI*H2R
         WTEMP=WR
         WR=WR*WPR-WI*WPI+WR
         WI=WI*WPR+WTEMP*WPI+WI
  200 CONTINUE
      H1R=Y(1)
      Y(1)=H1R+Y(2)
      Y(2)=H1R-Y(2)
C
      WR=1.0D0
      WI=0.0D0
      DO 300 I=3,N,2
         WTEMP=WR
         WR=WR*WPR0-WI*WPI0+WR
         WI=WI*WPR0+WTEMP*WPI0+WI
         Y1=Y(I)*WR-Y(I+1)*WI
         Y2=Y(I+1)*WR+Y(I)*WI
         Y(I)=Y1
         Y(I+1)=Y2
  300 CONTINUE
      SUM=0.5D0*Y(2)
      DO 310 I=N,2,-2
         SUM1=SUM
         SUM=SUM+Y(I)
         Y(I)=SUM1
  310 CONTINUE
      RETURN
      END
```

Fig. 4.8. The program list of the subroutine FFT.

```
      SUBROUTINE RANDOM(IX,RND)
C
      REAL*8 RND
      IR=48828125
      MR=2147483647
      IX=IX*IR
      IF(IX) 10,20,20
   10 IX=IX+MR+1
   20 RND=DFLOAT(IX)/DFLOAT(MR)
      RETURN
      END
```

Fig. 4.9. The program list of the subroutine RANDOM. This subroutine generates random numbers distributed uniformly in the range [0, 1].

using direct diagonalization techniques. The simplicity and versatility of the FOM make it very efficient, and its flexibility allows for wide applications in physical problems.

5.1. Dynamic structure factor $S(\boldsymbol{q}, \omega)$

The dynamic structure factor is defined, in general, by [30–32]

$$
\begin{aligned}
S(\boldsymbol{q}, \varepsilon) &= (n+1)\chi''(\boldsymbol{q}, \varepsilon) \\
&= (n+1)\pi \sum_{\lambda} \delta(\varepsilon - \varepsilon_{\lambda}) \left| \sum_{m} \mathrm{e}^{-\mathrm{i}\boldsymbol{q}\cdot\boldsymbol{R}_m} e_m(\lambda) \right|^2 ,
\end{aligned}
\tag{5.1}
$$

where the prefactor $(n+1)$ comes from the Bose distribution which is expressed by $1/(1-\mathrm{e}^{-\beta\varepsilon})$ with $\beta = 1/k_{\mathrm{B}}T$, $\boldsymbol{R}_m$ the position vector of the site m, respectively. The explicit form of Eq. (5.1) is given by

$$
S(\boldsymbol{q}, \varepsilon) = (n+1)\pi \sum_{\lambda} \delta(\varepsilon - \varepsilon_{\lambda}) \left[\left\{ \sum_{m} \cos(\boldsymbol{q}\cdot\boldsymbol{R}_m) e_m(\lambda) \right\}^2 + \left\{ \sum_{m} \sin(\boldsymbol{q}\cdot\boldsymbol{R}_m) e_m(\lambda) \right\}^2 \right] . \tag{5.2}
$$

The above definition of $S(\boldsymbol{q}, \varepsilon)$ can be related to the energy function introduced in Eq. (2.12). The energy function of Eq. (2.12) is expressed by

$$
E(\Omega, t) = \frac{1}{2} \sum_{\lambda} \left\{ \sum_{m} F_m e_m(\lambda) \right\}^2 \frac{\sin^2\{(\omega_{\lambda} - \Omega)t/2\}}{(\omega_{\lambda} - \Omega)^2} . \tag{5.3}
$$

Setting the external force in Eq. (5.3) as

$$
F_m = F_0 \cos(\boldsymbol{q}\cdot\boldsymbol{R}_m) \tag{5.4}
$$

we have the energy function $E(T, \Omega)$ of the form after a sufficient time interval T,

$$
E_{\boldsymbol{q}}^{\mathrm{c}}(\Omega, T) \equiv \frac{\pi T F_0^2}{4} \sum_{\lambda} \delta(\omega_{\lambda} - \Omega) \left\{ \sum_{m} \cos(\boldsymbol{q}\cdot\boldsymbol{R}_m) e_m(\lambda) \right\}^2 . \tag{5.5}
$$

The driving time-interval T should be chosen in the same way as the calculation of the spectral density. By setting the external force again as

$$F_m = F_0 \sin(\boldsymbol{q} \cdot \boldsymbol{R}_m) , \tag{5.6}$$

Eq. (5.3) yields

$$E_{\boldsymbol{q}}^{\mathrm{s}}(\Omega, T) \equiv \frac{\pi T F_0^2}{4} \sum_{\lambda} \delta(\omega_\lambda - \Omega) \left\{ \sum_m \sin(\boldsymbol{q} \cdot \boldsymbol{R}_m) e_m(\lambda) \right\}^2 . \tag{5.7}$$

Combining Eqs. (5.2), (5.5) and (5.7), the dynamic structure factor $S(\boldsymbol{q}, \varepsilon)$ is given by

$$S(\boldsymbol{q}, \varepsilon) = (n+1) \left| \frac{\mathrm{d}\omega(\varepsilon)}{\mathrm{d}\varepsilon} \right| \frac{4}{T F_0^2} \{ E_{\boldsymbol{q}}^{\mathrm{c}}(\omega(\varepsilon), T) + E_{\boldsymbol{q}}^{\mathrm{s}}(\omega(\varepsilon), T) \} . \tag{5.8}$$

This scheme makes it possible to calculate efficiently the ε dependence of $S(\boldsymbol{q}, \varepsilon)$. It is remarkable that this scheme requires the same computing time as that required for the spectral density, namely proportional to N as mentioned in Section 4.2.

It is straightforward to extend this algorithm to cases with vector displacements. For example, the dynamic structure factor of vibrational systems is defined by the formula

$$S(\boldsymbol{q}, \omega) = \frac{(n+1)}{\omega} \sum_{\lambda} \delta(\omega - \omega_\lambda) \left| \sum_m \{ \boldsymbol{q} \cdot \boldsymbol{e}_m(\lambda) \} \, \mathrm{e}^{-\mathrm{i}\boldsymbol{q} \cdot \boldsymbol{R}_m} \right|^2 , \tag{5.9}$$

where $\boldsymbol{e}_m(\lambda)$ is the vector displacement of the vibrational eigenmode λ at the mth site. Using this formula, one can calculate the frequency ω and wavenumber $\boldsymbol{q}$ dependencies of $S(\boldsymbol{q}, \omega)$ by setting as, instead of Eqs. (5.4) and (5.6),

$$\boldsymbol{F}_m = \boldsymbol{q} F_0 \cos(\boldsymbol{q} \cdot \boldsymbol{R}_m) \tag{5.10}$$

and

$$\boldsymbol{F}_m = \boldsymbol{q} F_0 \sin(\boldsymbol{q} \cdot \boldsymbol{R}_m) . \tag{5.11}$$

5.2. *Computing $S(\boldsymbol{q}, \omega)$ for random fractals*

We demonstrate the efficiency of this scheme by showing the calculated results of the dynamic structure factor for fracton dynamics on three-dimensional (3D) bond-percolating networks. Percolating networks [33,34] exhibit self-similar fractal geometry [35] with various physical implications [36–38], and also provide models of real fractal materials such as silica aerogels [39,40], sol–gel glasses [41], or colloids. It has been pointed out that vibrational excitations peculiar to fractal structures can be characterized by the fractal and fracton dimensionalities, D_{f} and $\tilde{d}$, respectively [42]. The excitations belonging to this class are called *fractons* [42]. Computer simulations have played an essential role for investigating fracton dynamics [43]. The dynamic structure factor provides important information on dynamic properties of percolating networks.

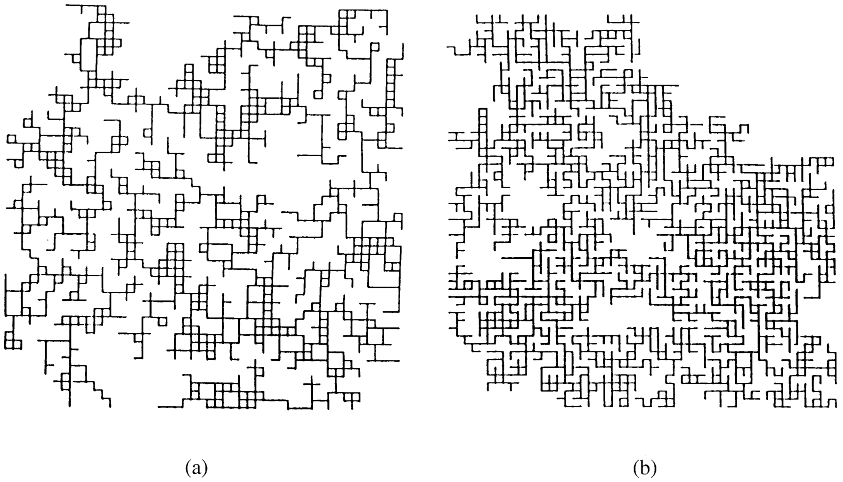


Fig. 5.1. (a) Site-percolating network and (b) bond-percolating network at the critical probabilities p_c.

We briefly introduce the essential points on dynamics of percolating networks [43]. To construct site-percolating networks, each intersection of a d-dimensional lattice is occupied by a particle at random with probability p. Bonds are said to be connected if particles are adjacent along a principal direction (see the illustration given in Fig. 5.1). In a bond-percolating network, all sites are initially occupied and bonds are randomly occupied with the probability p. A critical probability p_c exists such that, for $p \geq p_c$, a connected cluster will infinitely extend across the lattice, referred to as an infinite cluster. On the other hand, the remainder of occupied sites or bonds will form finite clusters. The probability that a site or bond belongs to the infinite cluster, $P(p)$, is characterized by the exponent β through the scaling formula [33],

$$P(p) \propto (p - p_c)^{\beta} . \tag{5.12}$$

There exists a unique length scale $\xi(p)$ in a percolating network which determines the crossover from homogeneous to fractal structures. This quantity is usually termed the percolation correlation length. For $L \gg \xi(p)$, the percolating network appears homogeneous, and the mass density ρ does not depend on L. The percolating network exhibits fractal geometry for $L \ll \xi(p)$ since the occupied mass density scales as Eq. (5.12). The correlation length $\xi(p)$ scales as

$$\xi(p) = \Xi_0 |p - p_c|^{-\nu} . \tag{5.13}$$

Thus, the network is fractal at short length scales, and Euclidean at long length scales.

It is known that the master equation for diffusion can be mapped onto a variety of physical systems [44], including elastic vibrations, spin waves in Heisenberg ferromagnets, and electrical RC circuits. This allows the vibrational problem to be mapped onto the diffusion problem. In homogeneous systems, the mean-square displacement of a random walker, $\langle R^2(t) \rangle$, is proportional to time t,

$$\langle R^2(t) \rangle \propto t \tag{5.14}$$

for any Euclidean dimension d. In percolating systems, the diffusion on a length scale $L \ll \xi(p)$ is anomalous. The mean-square displacement becomes [45]

$$\langle R^2(t)\rangle \propto t^{2/(2+\theta)} , \tag{5.15}$$

with $\theta = (\mu - \beta)/\nu$, where μ is called the conductivity exponent defined by $\sigma(p) \propto (p - p_c)^\mu$. Since $\theta > 0$, the diffusion slows down reflecting the hierarchically intricate structure of percolating networks. Eq. (5.15) holds also for the relationship between a wavelength $\Lambda(\omega)$ and frequency ω due to the mapping relationship between the diffusion equation and the vibrational equation, which is given by $\omega \propto \Lambda(\omega)^a$ with $a = 1 + (\mu - \beta)/2\nu$. Thus, one can consider Eq. (5.15) as the dispersion relation for elastic waves.

The equations of vibrational motion with *scalar* displacements on a percolating network driven by an external force with frequency Ω is

$$\frac{d^2 x_m}{dt^2} = \sum_n D_{mn} x_n + F_m \cos(\Omega t) , \tag{5.16}$$

where $x_m(t)$ is the displacement at the mth site, D_{mn} is the force constant between atoms m and n, and the mass of atoms is unity. In the percolating network model, $D_{mn} = -1$ if the mth atom is connected to the nth atom which is one of the nearest neighbors of the mth atom, the diagonal element $D_{mm} = -\sum_n D_{mn}$ and $D_{mn} = 0$ otherwise.

The dynamic structure factor $S(\boldsymbol{q}, \omega)$ for the percolating network calculated by the FOM is shown in Fig. 5.2. The percolating network is formed on a 120 × 120 × 120 cubic lattice at the percolation concentration $p = 0.31$ (the critical concentration p_c is 0.249). The number of sites is 1.7×10^6. $S(\boldsymbol{q}, \omega)$ for five different $\boldsymbol{q}$ along the [1 0 0] direction is plotted as a function of ω. The solid lines are only guides to the eye and the Bose factor is reduced. For small wave vector

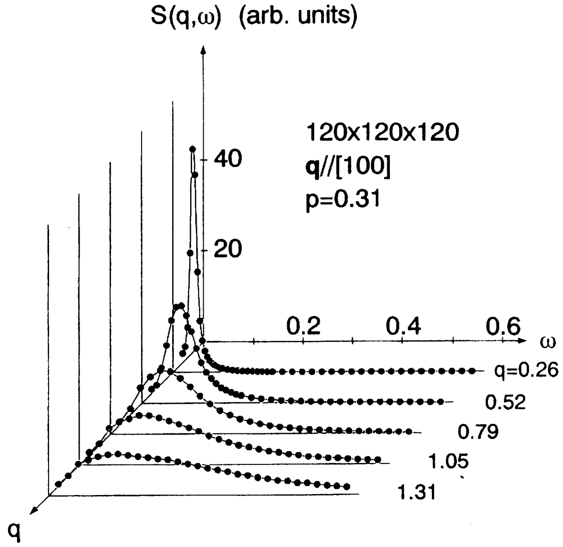


Fig. 5.2. ω dependence of the dynamical structure factor $S(\boldsymbol{q}, \omega)$ of the 3D bond-percolating network at $p = 0.31$ formed on a 120 × 120 × 120 cubic lattice. From Ref. [29].

$q(=|\boldsymbol{q}|<\xi^{-1})$, sharp peaks appear in the low-frequency region. With increasing q, peak positions shift to the higher-frequency region beyond $\omega_c \approx 0.07$, and the widths (τ^{-1}) of the peaks increase very rapidly. This indicates that the linewidth of fracton is very broad, originating from the Ioffe–Regel strong scattering limit $(\tau^{-1} \approx \omega)$. These convincing results have been the first verification of the validity of the single-length-postulate [46] for percolating systems by computer simulations.

5.3. Raman scattering intensity

The inelastic light scattering intensity such as the Raman scattering intensity presents rich information of dynamical properties of materials as well as the dynamic structure factor $S(\boldsymbol{q},\omega)$. Most of numerical studies on the Raman scattering intensity have been done by using direct diagonalization techniques [47–50] or the moment method [51] which consume a large amount of memory size. This results in a limited number of particles of the system, which causes sample-dependence or finite-size effects in the results. Due to this difficulty, no clear evidence had been given on the frequency dependence of the Raman scattering intensity for random systems.

The Raman scattering intensity for large sizes can be calculated by the FOM similar to the case of $S(\boldsymbol{q},\omega)$ with a computational effort linear in the matrix size N [29]. Taking into account the vector nature of the displacements of particles, the equation of motion with an external force applied to each site is given by

$$\frac{\mathrm{d}^2 x_m^\alpha(t)}{\mathrm{d}t^2} = \sum_{n,\beta} D_{mn}^{\alpha\beta} x_n^\beta(t) + F_m^\alpha \cos(\Omega t) , \tag{5.17}$$

where $x_m^\alpha(t)$ is the displacement of the mth atom (mass is unity) with the Cartesian component α, $D_{mn}^{\alpha\beta}$ the force constant between the mth and the nth atoms, and F_m^α the amplitude of the external force on the mth site in the α direction.

The energy function $E(\Omega,t)$ given by Eq. (2.12) should be read by taking account of the α component of F_m^α and $e_m^\alpha(\lambda)$ instead of F_m and $e_m(\lambda)$, where $e_m^\alpha(\lambda)$ is the α component of the eigenvector. After a sufficiently large time interval T, $E(\Omega,T)$ becomes

$$E(\Omega,T) = \frac{\pi T}{4} \sum_\lambda \delta(\Omega-\omega_\lambda) \left\{ \sum_{m,\alpha} F_m^\alpha e_m^\alpha(\lambda) \right\}^2 , \tag{5.18}$$

where ω_λ is the eigenfrequency of the mode λ. The r.h.s. of Eq. (5.18) is related to the definition of the Raman-scattering intensity $I_{\alpha\beta}(\omega)$ which is defined by

$$I_{\alpha\beta} \equiv \frac{1}{2\pi N} \int \mathrm{d}t\, \mathrm{e}^{\mathrm{i}\omega t} \sum_{mn} \langle \mu_m^{\alpha\beta}(t) \mu_n^{\alpha\beta}(0) \rangle , \tag{5.19}$$

where $\langle \cdots \rangle$ is the thermal average, N the number of sites, and $\mu_m^{\alpha\beta}(t)$ the polarizability at the site m. Expanding the polarizability $\mu_m^{\alpha\beta}(t)$ in terms of small displacements $x_m^\alpha(t)$, one has

$$I_{\alpha\beta}(\omega) = \frac{\hbar}{2N^2} \frac{\langle n+1 \rangle}{\omega} \sum_\lambda \delta(\omega-\omega_\lambda) \left\{ \sum_{m,\gamma} f_m^{\alpha\beta\gamma} e_m^\gamma(\lambda) \right\}^2 , \tag{5.20}$$

where $\langle n+1\rangle$ is the Bose factor, and $f_m^{\alpha\beta\gamma} \equiv \sum_n \partial\mu_m^{\alpha\beta}/\partial x_n^\gamma$. We have neglected the contribution from elastic scattering.

As an example, we consider the dipole-induced-dipole (DID) mechanism. In this case, one can express $f_m^{\alpha\beta\gamma}$ in Eq. (5.20) by using the derivative of the dipole propagator as [47,49,50,52]

$$f_m^{\alpha\beta\gamma} \equiv \sum_n \tilde{\mu}_m \tilde{\mu}_n \left[-3\frac{\delta_{\alpha\beta} r_\gamma + \delta_{\beta\gamma} r_\alpha + \delta_{\gamma\alpha} r_\beta}{r^5} + 15\frac{r_\alpha r_\beta r_\gamma}{r^7} \right], \tag{5.21}$$

where r_α is defined as $\boldsymbol{r}_{mn} \equiv \boldsymbol{R}_m - \boldsymbol{R}_n = (r_x, r_y, r_z)$, $r \equiv |\boldsymbol{r}_{mn}|$, and $\tilde{\mu}_m$ the bare polarizability of the site m. The bare polarizability at each site is taken to be isotropic. Using Eqs. (5.18) and (5.20), the Raman-scattering intensity is expressed as

$$I_{\alpha\beta}(\omega) = \frac{\hbar}{2N^2}\frac{\langle n+1\rangle}{\omega}\frac{4E_{\alpha\beta}(\omega, T)}{\pi T}\ , \tag{5.22}$$

where the amplitude of the periodic force $\{F_m^\alpha\}$ in Eq. (5.18) is taken to be $f_m^{\alpha\beta\gamma}$. This method enables us to calculate Eq. (5.19) with arbitrary resolution of frequency $\delta\omega$ by choosing the proper time interval $T = 4\pi/\delta\omega$. Thus, we can calculate the Raman scattering intensity of complex systems through minor modifications of the scheme for calculating the dynamic structure factor $S(\boldsymbol{q}, \omega)$. We omit to present examples of calculated results of the Raman scattering intensity since the procedure is completely the same as the case of the dynamic structure factor in the previous subsection, which requires the same computing time as that for calculating the density of states (namely scaling linearly with the system size N). See Ref. [29] in which details of calculated results are given. The results are completely in agreement with the scaling theory [46].

6. Computing linear response functions for quantum systems

Linear response functions are very significant for gaining insight into dynamic properties of quantum systems. Calculations of linear response functions for quantum systems described by $N \times N$ Hamiltonian matrices normally require the evaluation of all eigenvalues and corresponding eigenvectors. However, as the size N of matrices becomes large, standard diagonalization routines need a large amount of the computing time scaling with N^3 as well as memory space proportional to N^2. They remain limited to systems of modest size because of the high computational cost.

The development of an efficient algorithm for linear response functions of quantum systems is quite challenging. So far, many algorithms suitable for the calculation of linear response functions such as the Kubo–Greenwood formula [53,54] for the AC conductivity have been proposed. These include the methods based on the continued fraction technique [55], the recursion method [56,57,155], the moments method [59], the full-diagonalization technique using the Lanczos method [4], the maximum entropy method [60–62], the Chebyshev polynomial expansion [58,63–71,156,157], the conjugated gradient method [72,73], the quantum-molecular-dynamics calculation [74] and the method for direct integration of the time-dependent Schrödinger equation [75–77].

The algorithm based on the FOM proposed by Nakayama and Shima [9,78,79] is quite effective for the calculation of linear response functions for quantum systems described by large-scale Hamiltonian matrices. The advantages of this method compared to existing methods are that (i) it requires memory space of the order of N for sparse matrices, (ii) the computing time is proportional to N^2, and (iii) it is easy to vectorize and parallelize for implementations in array-processing modern supercomputers. The last point comes from the fact that the time-consuming part in computations is the solving of the time evolution of equations of motion, and so the program is easily optimized.

6.1. Kubo formula and the FOM

We consider a quantum system described by the following Hamiltonian:

$$\hat{H} = \sum_{mn} D_{mn}|m\rangle\langle n| \quad (m,n = 1,2,\ldots,N)\ , \tag{6.1}$$

where $\langle m|$ is the bra vector in the *site* notation. The ket vector is defined as well. Since the set $\{|m\rangle\}$ satisfies the closure relation $\sum_m |m\rangle\langle m| = 1$, an arbitrary state is expressed as

$$|\Psi(t)\rangle = \sum_m a_m(t)|m\rangle\ . \tag{6.2}$$

Let us impose a small perturbation $\hat{V}$ to the system given by

$$\hat{V} = -\frac{1}{2}\sum_\alpha \hat{x}_\alpha(f_0^\alpha \mathrm{e}^{-\mathrm{i}\omega t} + \text{c.c.})\ , \tag{6.3}$$

where $\hat{x}_\alpha$ is the α component of the *generalized* displacement and f_0^α is the corresponding generalized force. c.c. indicates a complex conjugate. In the spectral representation, this is written in the form

$$\begin{aligned}\hat{V} &= \sum_\alpha \sum_{mn} |m\rangle V_{mn}^\alpha(t)\langle n| \\ &= -\frac{1}{2}\sum_\alpha \sum_{mn} (|m\rangle x_{mn}^\alpha \langle n|)(f_0^\alpha \mathrm{e}^{-\mathrm{i}\omega t} + \text{c.c.})\ .\end{aligned} \tag{6.4}$$

Here, $V_{mn}^\alpha(t) = \langle m|\hat{V}_\alpha|n\rangle$ and $x_{mn}^\alpha = \langle m|\hat{x}_\alpha|n\rangle$. Substituting Eqs. (6.1) and (6.3) into the time-dependent Schrödinger equation for $|\Psi(t)\rangle$ and multiplying by $\langle k|$ from the left, we have the inhomogeneous coupled linear differential equation

$$\mathrm{i}\hbar\frac{\mathrm{d}a_m(t)}{\mathrm{d}t} - \sum_n D_{mn}a_n(t) = \sum_\alpha \sum_n V_{mn}^\alpha(t)a_n(t)\ . \tag{6.5}$$

For a small perturbation, the time-dependent first-order perturbation theory is applicable by putting $a_m(t) = a_m^{(0)}(t) + \sum_\alpha a_{m\alpha}^{(1)}(t)$ into Eq. (6.5). The zeroth-order equation becomes

$$\mathrm{i}\hbar\frac{\mathrm{d}a_m^{(0)}(t)}{\mathrm{d}t} - \sum_n D_{mn}a_n^{(0)}(t) = 0\ , \tag{6.6}$$

while the first-order term yields

$$\mathrm{i}\hbar\frac{\mathrm{d}a_{m\alpha}^{(1)}(t)}{\mathrm{d}t}-\sum_n D_{mn}a_{n\alpha}^{(1)}(t)=\sum_n V_{mn}^{\alpha}a_n^{(0)}(t)\ . \tag{6.7}$$

From Eqs. (6.4) and (6.7), one has the first-order linear differential equation with the periodic external force

$$\mathrm{i}\hbar\frac{\mathrm{d}a_{m\alpha}^{(1)}(t)}{\mathrm{d}t}-\sum_n D_{mn}a_{n\alpha}^{(1)}(t)=\ -\frac{\hbar}{2}(F_{m\alpha}\mathrm{e}^{-\mathrm{i}\omega t}+\tilde{F}_{m\alpha}\mathrm{e}^{\mathrm{i}\omega t})\,\mathrm{e}^{-\mathrm{i}\omega_{\lambda_0}t}\ . \tag{6.8}$$

Here

$$F_{m\alpha}=\sum_n\frac{f_0^{\alpha}}{\hbar}x_{mn}^{\alpha}e_n(\lambda_0),\quad \tilde{F}_{m\alpha}=\sum_n\frac{f_0^{\alpha *}}{\hbar}x_{mn}^{\alpha}e_n(\lambda_0)\ , \tag{6.9}$$

where $e_m(\lambda_0)\equiv\langle m|\omega_{\lambda_0}\rangle$ is the mth element of the initial eigenvector belonging to the eigenvalue $\hbar\omega_{\lambda_0}$ of the matrix $\{D_{mn}\}$ [see Eq. (6.6)]. In the derivation of Eq. (6.8), we assume that the unperturbed state with the eigenfrequency $\hbar\omega_{\lambda_0}$ is given by

$$a_m^{(0)}(t)=e_m(\lambda_0)\,\mathrm{e}^{-\mathrm{i}\omega_{\lambda_0}t}\ .$$

The function $a_{m\alpha}^{(1)}(t)$ is expanded by a set of eigenvectors $\{e_m(\lambda)\}$ as

$$a_{m\alpha}^{(1)}(t)=\sum_\lambda \xi_\lambda^\alpha(t)e_m(\lambda)\ , \tag{6.10}$$

where $\xi_\lambda^\alpha(t)$ is the amplitude of the mode λ. From Eq. (6.8), one has the equation for $\xi_\lambda^\alpha(t)$,

$$\mathrm{i}\frac{\mathrm{d}\xi_\lambda^\alpha(t)}{\mathrm{d}t}-\omega_\lambda\xi_\lambda^\alpha(t)=\ -\frac{1}{2}\left\{\sum_n e_n^*(\lambda)[F_{n\alpha}\mathrm{e}^{-\mathrm{i}(\omega_{\lambda_0}+\omega)t}+\tilde{F}_{n\alpha}\mathrm{e}^{-\mathrm{i}(\omega_{\lambda_0}-\omega)t}]\right\}\ , \tag{6.11}$$

where the orthonormal condition $\sum_m e_m(\lambda)e_m^*(\lambda')=\delta_{\lambda\lambda'}$ is used. Under the initial condition $\xi_\lambda^\alpha(0)=0$, the solution of Eq. (6.11) becomes

$$\xi_\lambda^\alpha(t)=\ -\frac{\mathrm{e}^{-\mathrm{i}\omega_\lambda t}}{2}\left\{\sum_n e_n^*(\lambda)\left[F_{n\alpha}\frac{\mathrm{e}^{\mathrm{i}(\omega_\lambda-\omega_{\lambda_0}-\omega)t}-1}{\omega_\lambda-\omega_{\lambda_0}-\omega}+\tilde{F}_{n\alpha}\frac{\mathrm{e}^{\mathrm{i}(\omega_\lambda-\omega_{\lambda_0}+\omega)t}-1}{\omega_\lambda-\omega_{\lambda_0}+\omega}\right]\right\}\ . \tag{6.12}$$

The second term in the square brackets is negligible since we consider the case of zero temperature and treat the Fermi distribution function with the Fermi frequency $\omega_\mathrm{F}=E_\mathrm{F}/\hbar$ as a step function. This implies that $\omega_\lambda>\omega_\mathrm{F}\geq\omega_{\lambda_0}$, namely $\omega_\lambda-\omega_{\lambda_0}>0$, indicating that the contribution from the first term with $\omega_{\lambda_0}+\omega\approx\omega_\lambda$ is dominant.

Let us introduce the *resonance* function defined by

$$E_{\alpha\beta}(\Omega,t)=\sum_m a_{m\alpha}^{(1)*}(t)a_{m\beta}^{(1)}(t)=\sum_\lambda \xi_\lambda^{\alpha *}(t)\xi_\lambda^\beta(t)\ . \tag{6.13}$$

Substituting Eq. (6.12) into Eq. (6.13), one has, at $t=T$,

$$E_{\alpha\beta}(\Omega,T)=\sum_\lambda\left|\sum_m F_{m\alpha}e_m^*(\lambda)\right|^2\frac{\sin^2\{(\omega_\lambda-\Omega)T/2\}}{(\omega_\lambda-\Omega)^2}\ , \tag{6.14}$$

where $\Omega = \omega_{\lambda_0} + \omega$. The orthogonality condition for eigenvectors $\{e_m(\lambda)\}$ is used to derive Eq. (6.14). Eigenvectors contributing to the sum in Eq. (6.14) are those whose frequencies lie within the width of π/T around Ω. Suppose that the following conditions are satisfied as mentioned in Section 4.2,

$$\frac{1}{\Omega} \ll T \ll \frac{4\pi}{\Delta\omega} , \tag{6.15}$$

where $\Delta\omega$ is the average eigenfrequency spacing. Taking a proper time interval T satisfying the condition Eq. (6.15), Eq. (6.14) gives

$$E_{\alpha\beta}(\Omega, T) = \frac{\pi T}{2}\sum_{\lambda}\left|\sum_{m} F_{m\alpha} e_m^*(\lambda)\right|^2 \delta(\omega_\lambda - \Omega) . \tag{6.16}$$

A straightforward calculation leads to the following representation, using the expression for $F_{m\alpha}$ introduced by Eq. (6.9),

$$\begin{aligned}\sum_m F_{m\alpha} e_m^*(\lambda) &= \frac{f_0^\alpha}{\hbar}\sum_{mn} x_{mn}^\alpha e_n(\lambda_0) e_m^*(\lambda) \\ &= \frac{f_0^\alpha}{\hbar}\sum_{mn} \langle\omega_\lambda|m\rangle\langle m|\hat{x}_\alpha|n\rangle\langle n|\omega_{\lambda_0}\rangle \\ &= \frac{f_0^\alpha}{\hbar}\langle\omega_\lambda|\hat{x}_\alpha|\omega_{\lambda_0}\rangle .\end{aligned} \tag{6.17}$$

Substituting Eq. (6.17) into Eq. (6.16), we have

$$E_{\alpha\beta}(\Omega, T) = \frac{\pi T f_0^{\alpha *} f_0^\beta}{2\hbar^2}\sum_\lambda \langle\omega_{\lambda_0}|\hat{x}_\alpha|\omega_\lambda\rangle\langle\omega_\lambda|\hat{x}_\beta|\omega_{\lambda_0}\rangle\delta(\omega_\lambda - \Omega) . \tag{6.18}$$

The generalized susceptibility $\chi_{\alpha\beta}(\omega)$ under the generalized external force defined in Eq. (6.3) is given by the Kubo formula [53],

$$\chi_{\alpha\beta}(\omega) = \frac{\mathrm{i}}{\hbar}\int_0^\infty \mathrm{e}^{\mathrm{i}\omega t}\langle[\hat{x}_\alpha(t), \hat{x}_\beta(0)]\rangle\, \mathrm{d}t , \tag{6.19}$$

where angular brackets denote the quantum and thermal averages. At zero temperature, the imaginary part of the generalized susceptibility for a given initial state $|\omega_{\lambda_0}\rangle$ is expressed by

$$\begin{aligned}\chi''_{\alpha\beta}(\omega) = &\frac{\pi}{\hbar}\sum_\lambda \mathrm{Re}[\langle\omega_{\lambda_0}|\hat{x}_\alpha|\omega_\lambda\rangle\langle\omega_\lambda|\hat{x}_\beta|\omega_{\lambda_0}\rangle][\delta(\omega_{\lambda\lambda_0} - \omega) - \delta(\omega_{\lambda\lambda_0} + \omega)] \\ &- \frac{1}{\hbar}P\sum_\lambda \mathrm{Im}[\langle\omega_{\lambda_0}|\hat{x}_\alpha|\omega_\lambda\rangle\langle\omega_\lambda|\hat{x}_\beta|\omega_{\lambda_0}\rangle]\left(\frac{1}{\omega_{\lambda\lambda_0} - \omega} - \frac{1}{\omega_{\lambda\lambda_0} + \omega}\right) ,\end{aligned} \tag{6.20}$$

where the symbol P means a principal value integral, and $\omega_{\lambda\lambda_0} = \omega_\lambda - \omega_{\lambda_0}$. If $\alpha = \beta$, the second term of Eq. (6.20) vanishes. In this case, choosing $f_0^\alpha = f_0^\beta = 1$, $\chi''_{\alpha\alpha}(\omega)$ can be expressed by the resonance function given by Eq. (6.18) as

$$\chi''_{\alpha\alpha}(\omega) = \frac{2\hbar E_{\alpha\alpha}(\Omega, T)}{T} . \tag{6.21}$$

This is the key relation between the resonance function $E_{\alpha\alpha}(\Omega, T)$ [Eq. (6.13)] and the imaginary part of the generalized susceptibility $\chi''_{\alpha\alpha}(\omega)$. From this relation, we can compute $\chi''_{\alpha\alpha}(\omega)$ from calculations of eigenvectors $e_m(\lambda)$ in Eq. (6.9) and the time-development of $a^{(1)}_{m\alpha}(t)$ governed by Eq. (6.8).

6.2. Computing the Kubo–Greenwood formula

This section describes the relationship between the Kubo–Greenwood formula [53,54] for the AC conductivity as a special case of the generalized susceptibility $\chi''_{\alpha\beta}(\omega)$ and the resonance function defined by Eq. (6.13). A small perturbation due to the vector potential $\boldsymbol{A}(t)$ applied to an electronic system is expressed as $\hat{V} = -\hat{\boldsymbol{J}}\cdot\boldsymbol{A}(t)$, where $\hat{\boldsymbol{J}}$ is the current operator. Since the conductivity is defined as the response to the electric field $\boldsymbol{E}(t)$, the generalized conductivity $\Sigma_{\alpha\beta}(\omega)$ is related to the generalized susceptibility $\chi_{\alpha\beta}(\omega)$ via the relation $\Sigma_{\alpha\beta}(\omega) = \chi_{\alpha\beta}(\omega)/\mathrm{i}\omega L^d$, where L is the linear size of the d-dimensional system.

The generalized conductivity is given by [53]

$$\Sigma_{\alpha\beta}(\omega) = \frac{ine^2}{m\omega}\delta_{\alpha\beta} + \frac{1}{\hbar\omega L^d}\lim_{\varepsilon\to 0}\int_0^\infty \mathrm{e}^{\mathrm{i}\omega t-\varepsilon t}\langle[\hat{J}_\alpha(t), \hat{J}_\beta(0)]\rangle\,\mathrm{d}t , \tag{6.22}$$

where $\hat{J}_\alpha(t)$ is the α component of the current operator. The first term comes from field-free electrons. From this, one can derive the *longitudinal* component of the AC conductivity $\sigma(\omega) \equiv \mathrm{Re}[\Sigma_{\alpha\alpha}(\omega)]$ by setting $\alpha = \beta$ and $\hat{x}_\alpha = \hat{J}_\alpha$ in Eq. (6.3). Considering the Fermi distribution function $f(\omega)$ at zero temperature, the longitudinal AC conductivity $\sigma(\omega)$ is written by the imaginary part of the generalized susceptibility $\chi''(\omega)[=\chi''_{\alpha\alpha}(\omega)]$ as

$$\begin{aligned}\sigma(\omega) &= \frac{2}{\omega L^d}\sum_{\omega_{\lambda_0}}\chi''(\omega)[f(\omega_{\lambda_0}) - f(\omega_{\lambda_0}+\omega)]\\ &= \frac{2}{\omega L^d}\sum_{\omega_{\lambda_0}=\omega_{\mathrm{F}}-\omega}^{\omega_{\mathrm{F}}}\chi''(\omega) ,\end{aligned} \tag{6.23}$$

where the spin freedom is taken into account and the definition of the Fermi frequency is $\omega_{\mathrm{F}} = E_{\mathrm{F}}/\hbar$. The meaning of $\sum_{\omega_{\lambda_0}=\omega_{\mathrm{F}}-\omega}^{\omega_{\mathrm{F}}}$ is the sum over the initial state $|\omega_{\lambda_0}\rangle$ at zero temperature. From Eqs. (6.21) and (6.23), the longitudinal AC conductivity expressed by the resonance function given by Eq. (6.13) becomes

$$\sigma(\omega) = \frac{4\hbar}{\omega T L^d}\sum_{\omega_{\lambda_0}=\omega_{\mathrm{F}}-\omega}^{\omega_{\mathrm{F}}} E(\Omega, T) , \tag{6.24}$$

where the time interval T satisfies the condition Eq. (6.15).

Eq. (6.23) is equivalent to the Kubo–Greenwood formula as verified below. Eq. (6.23) can be rewritten as

$$\sigma(\omega) = \frac{2}{\omega} \int_{E_{\mathrm{F}} - \hbar\omega}^{E_{\mathrm{F}}} \mathrm{d}E_{\lambda_0} \mathscr{D}(E_{\lambda_0}) \chi''(\omega) \,, \tag{6.25}$$

where $\mathscr{D}(E_{\lambda_0})$ means the spectral density of states at the eigenenergy $E_{\lambda_0} = \hbar\omega_{\lambda_0}$. Substituting Eq. (6.20) into Eq. (6.25) and taking account of $\omega_\lambda - \omega_{\lambda_0} > 0$ as explained below Eq. (6.12), one has

$$\sigma(\omega) = \frac{2\pi e^2}{\hbar\omega} \sum_{\omega_\lambda} \left\{ \int_{E_{\mathrm{F}} - \hbar\omega}^{E_{\mathrm{F}}} \mathrm{d}E_{\lambda_0} |\langle \omega_\lambda | \hat{v} | \omega_{\lambda_0} \rangle|^2 \mathscr{D}(E_{\lambda_0}) \delta(\omega_{\lambda_0} + \omega - \omega_\lambda) \right\} , \tag{6.26}$$

where $\hat{v}$ is the velocity operator. The use of the relation $\Sigma_\lambda = L^d \int \mathscr{D}(E_\lambda) \, \mathrm{d}E_\lambda$ in Eq. (6.26) yields

$$\begin{aligned} \sigma(\omega) &= \frac{2\pi e^2 L^d}{\omega} \int \mathrm{d}E_\lambda \int_{E_{\mathrm{F}} - \hbar\omega}^{E_{\mathrm{F}}} \mathrm{d}E_{\lambda_0} |\langle \omega_\lambda | \hat{v} | \omega_{\lambda_0} \rangle|^2 \mathscr{D}(E_{\lambda_0}) \mathscr{D}(E_\lambda) \delta(E_{\lambda_0} + \hbar\omega - E_\lambda) \\ &= 2\pi e^2 \hbar L^d \int_{E_{\mathrm{F}} - \hbar\omega}^{E_{\mathrm{F}}} \mathrm{d}E_{\lambda_0} \frac{|\langle \omega_{\lambda_0} + \omega | \hat{v} | \omega_{\lambda_0} \rangle|^2}{\hbar\omega} \mathscr{D}(E_{\lambda_0}) \mathscr{D}(E_{\lambda_0} + \hbar\omega) \,. \end{aligned} \tag{6.27}$$

This is the Kubo–Greenwood formula [53,54]. We have proved the equivalency between Eq. (6.23) and the Kubo–Greenwood formula.

The calculation of the resonance function in Eq. (6.13) is reduced to the numerical solution of the first-order coupled linear differential equation with a periodic external force, expressed by Eq. (6.8). By decomposing the function $a_{m\alpha}^{(1)}(t)$ in Eq. (6.8) into a real part $x_m(t)$ and an imaginary part $y_m(t)$, one has

$$\hbar \frac{\mathrm{d}x_m(t)}{\mathrm{d}t} - \sum_n D_{mn} y_n(t) = \frac{1}{2} F_m \sin(\Omega t) \,, \tag{6.28}$$

$$\hbar \frac{\mathrm{d}y_m(t)}{\mathrm{d}t} - \sum_n D_{mn} x_n(t) = \frac{1}{2} F_m \cos(\Omega t) \,. \tag{6.29}$$

Here we have assumed, for simplicity, that the elements $\{D_{mn}\}$ and $\{F_m\}$ are real. These equations of motion can be numerically solved by the FEM described in Section 3.2.

Details of the implementation of our algorithm are as follows. (i) We prepare the initial states $\{e_m(\lambda_0)\}$ belonging to the eigenfrequency ω_{λ_0} by applying the FOM described in Section 2.3. (ii) The matrix elements x_{mn}^α in Eq. (6.4) are given analytically, and the external force $F_{m\alpha}$ defined by Eq. (6.9) is determined. (iii) Taking the time interval T as $T = 4\pi/\Delta\Omega_{\mathrm{R}}$ ($\Delta\Omega_{\mathrm{R}} \gg \delta\omega$), we calculate $x_m(t)$ and $y_m(t)$ described by Eqs. (6.28) and (6.29) with the initial conditions $x_m(0) = y_m(0) = 0$ and $\dot{x}_m(0) = \dot{y}_m(0) = 0$. We can finally obtain the resonance function by using Eq. (6.13). Note that the frequency resolution $\Delta\Omega_{\mathrm{R}}$ can be determined and controlled by the time interval T.

6.3. AC conductivity of 1D chain

In order to assess the efficiency of this algorithm, we illustrate a one-dimensional (1D) tight-binding Hamiltonian with N sites given by

$$\hat{H} = \sum_m \varepsilon_m |m\rangle\langle m| - \sum_{mn} t_{mn} |m\rangle\langle n| \,, \tag{6.30}$$

where we set $\varepsilon_m = 2$ and $t_{mm\pm 1} = 1$ for the hopping term between nearest neighbors. The matrix elements for the current operator $\hat{J} = e\hat{v}$ are obtained from the Heisenberg equation of motion as $J_{mm\pm 1} = \langle m|\hat{J}|m \pm 1\rangle = \mp \mathrm{i}e/\hbar$ by taking a lattice spacing $a = 1$. We have calculated the resonance function defined by Eq. (6.13) with $\alpha = \beta$ under the fixed boundary condition.

Fig. 6.1 represents the comparison of numerical results (solid circles) with the analytic solution (solid line) for the 1D chain with $N = 10{,}000$ on a double logarithmic scale. The system of units used here is $e = \hbar = 1$. For these calculations, we have chosen the resonance width in Eq. (6.14) as $\Delta\Omega_\mathrm{R} = 4\pi/T$ with $T = 4\pi \times 100$. There exist about 25 modes within this resonance width $\Delta\Omega_\mathrm{R}$. The initial eigenvector $\{e_m(\lambda_0)\}$ calculated by the FOM has a purity of $A_2/A_1 \approx 10^{-3}$. We see from Fig. 6.1 that the calculated result agrees fairly well with the analytic result over an order of magnitude in frequency.

6.4. Critical behavior of AC conductivity near the Anderson transition

The scaling arguments of localization [80,81] have stimulated many works on both static [82] and dynamic critical behavior [83,84] for disordered electron systems, especially on the Anderson

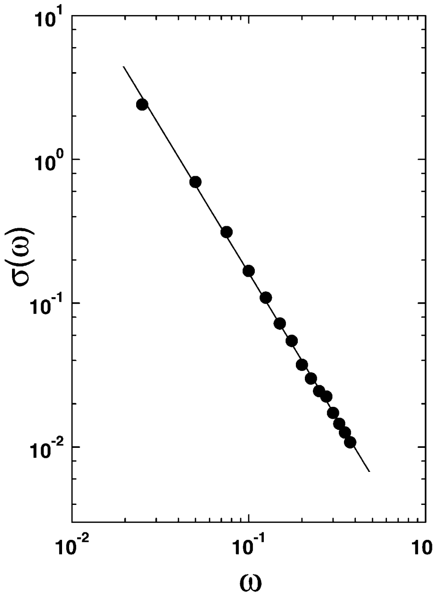


Fig. 6.1. AC conductivity $\sigma(\omega)$ of the 1D tight-binding chain with the system size $N = 10{,}000$. The resonance width $\Delta\Omega_\mathrm{R}$ is taken to be $\Delta\Omega_\mathrm{R} = 0.01$ in the frequency range $\omega = 0.02$–0.4 in the system of units given in the text. The solid line indicates the analytic solution showing the ω^{-2} dependence. From Ref. [9].

transition. The existence of this transition depends on the dimensionality and the symmetry of the system. Three-dimensional (3D) systems may show the Anderson transition, and their critical behaviors are classified into three universality classes according to the basic symmetry of the Hamiltonian [82,85,86]. Systems being invariant under spin rotation in addition to time reversal constitute the orthogonal class, while systems being invariant under time reversal but having no spin-rotational symmetry belong to the symplectic class. Systems without time-reversal symmetry forms the unitary class.

Many numerical works have contributed to reveal both the static and dynamic behaviors of the transition through the investigations of localization length [87–89], diffusion of wave packets [82,90–93], and level statistics [94–98]. The value of the critical disorder W_c above which all states are localized is known to be $W_c = 16.5$ for 3D orthogonal systems [87,99,100]. Wegner [83] has predicted in terms of a scaling argument that the AC conductivity $\sigma(\omega)$ in 3D systems near the transition obeys the power law $\sigma(\omega) \propto \omega^{1/3}$. This behavior was not numerically verified until the numerical analysis by Lambrianides and Shore [101]. They have evaluated the Kubo–Greenwood formula [53,54] by calculating directly eigenvectors of the system with the diagonalization method, so the system sizes $N(= L^3)$ treated were very limited ($L = 6$–14), indicating the relevance of the finite size effect.

We show here the calculated results of AC conductivities $\sigma(\omega)$ for 3D orthogonal systems to demonstrate the efficiency of our numerical method. The system is described by a Hamiltonian similar to Eq. (6.30) except that the site indices m and n run over three-dimensional lattice points. The on-site potential ε_m is uniformly distributed in the range $[-W, W]$. The hopping energy t_{mn} is chosen as unity for the nearest neighbor sites and zero otherwise. The densities of states calculated by the FOM described in Section 2.2 for the system size $N = 30^3$ are shown in Fig. 6.2. We consider the case of critical disorder $W_c(= 16.5)$, for which the mobility edge appears at $\omega = 0$. The Fermi energy $\hbar\omega_F$ is set to be zero, namely at the center of the flat band (see Fig. 6.2). Fig. 6.3 shows the calculated AC conductivity for 3D systems with $N = 30^3$ averaged over 20 samples with various realizations of the random potential $\{\varepsilon_m\}$. The result clearly shows the power law $\sigma(\omega) \propto \omega^{1/3}$ as predicted by Wegner [83] for 3D orthogonal systems. The error is within 12%.

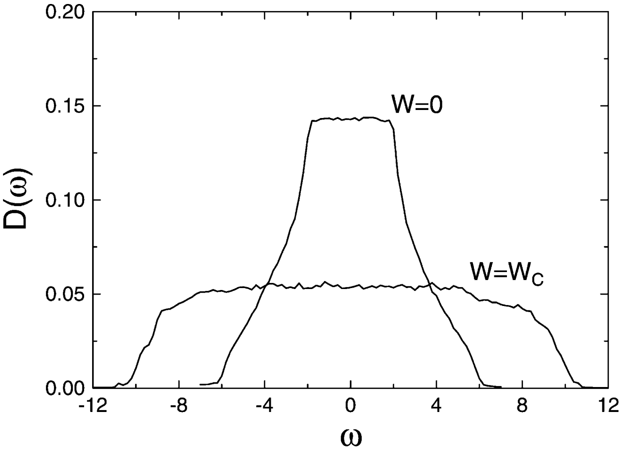
7. Finite-time scaling method for the FOM

7.1. Finite-time scaling

The method described in Section 2.2 is quite effective for calculating densities of states (DOS) of large-scale systems. However, it should be borne in mind that the DOS depends on the resolution $\delta\varepsilon$ [thus on the time interval T in Eq. (4.16)]. If the DOS of an infinite system has a power-law dependence near a critical energy ε_c such as

$$\mathscr{D}(\varepsilon) \propto (\varepsilon_c - \varepsilon)^{\alpha} , \tag{7.1}$$

one has to take great care of choosing the value of T to determine the exponent α and the critical energy ε_c, because the spectrum becomes dilute continuously on approaching ε_c. To avoid this difficulty, a scaling approach combined with the FOM proposed by Hukushima and Nemoto [102] is very practical.



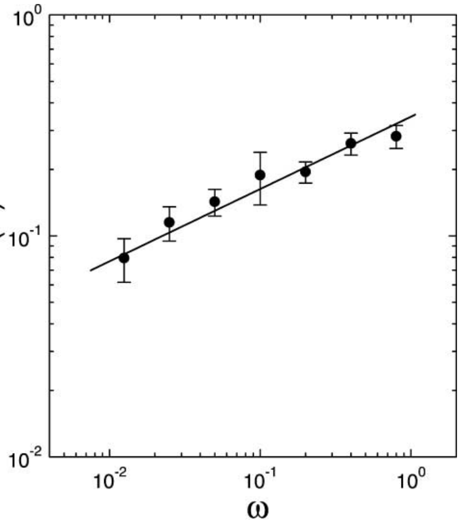


Fig. 6.2. Densities of states for the 3D Anderson model of noninteracting electrons in a uniformly distributed random potential with $W = W_c$ and $W = 0$. The system size is $N = 30^3$. The resonance width is taken as $\Delta\Omega_R = 0.2$. The data are averaged over 10 samples. From Ref. [9].

Fig. 6.3. AC conductivity $\sigma(\omega)$ for the 3D Anderson model of noninteracting electrons in a uniformly distributed random potential with $W = W_c$. The system size is $N = 30^3$. The resonance width is taken as $\Delta\Omega_R = 0.01$ for the system size $N = 30^3$. The data are averaged over 20 samples. The solid line is drawn by a least squares fit and each of the error bars is defined as a standard deviation. From Ref. [9].

Since the energy distance $\varepsilon_c - \varepsilon$ provides the unique characteristic time scale of the system if the DOS obeys Eq. (7.1), the DOS calculated by the FOM with a finite time interval T is expressed in terms of the finite-time scaling form such as

$$\mathscr{D}(\varepsilon, T) = (\varepsilon_c - \varepsilon)^{\alpha} f[(\varepsilon_c - \varepsilon)T] , \tag{7.2}$$

where the scaling function should have the asymptotic form:

$$f(x) \propto \begin{cases} \text{const.} & \text{for } x \gg 1 , \\ x^{-\alpha} & \text{for } x \ll 1 . \end{cases} \tag{7.3}$$

The last form reflects the fact that the DOS becomes constant if the time interval T is *short* and $\delta\varepsilon$ becomes *large*. The validity of this form can be verified by substituting Eq. (2.13) into Eq. (2.15) and performing the integration using the formula $\sum_\lambda = \int \mathrm{d}\varepsilon \mathscr{D}(\varepsilon)$. Hukushima and Nemoto [102] have applied this argument to investigate the band edge structure of the $\pm J$ spin glass model. They have succeeded in obtaining the exponent α and the energy ε_c by fitting the data for various T to the scaling function $f(x)$.

The idea of this technique is applicable to determine a dynamic exponent near a quantum phase transition [9,78,79]. As mentioned in Section 6.4, the 3D Anderson model exhibits the metal-insulator transition, and the AC conductivity $\sigma(\omega)$ at the transition point ($W = W_c$ and $\hbar\omega_F = 0$)

follows:

$$\sigma(\omega) \propto \omega^{\delta} , \tag{7.4}$$

with $\delta = \frac{1}{3}$ [83]. The exponent δ can be determined accurately by using the finite-time scaling technique. Since the DOS near $\omega_{\mathrm{F}} = 0$ (the critical point) is almost constant as shown in Fig. 6.2, there is no characteristic energy at criticality. Therefore, only the frequency ω characterizes the time scale of the system. Thus, the AC conductivity calculated by the FOM with a finite time interval T is written in the scaling form:

$$\sigma(\omega, T) = T^{-\delta} G(\omega T) , \tag{7.5}$$

where the asymptotic form of $G(z)$ should be

$$G(z) \propto \begin{cases} z^{\delta} & \text{for } z \gg 1 , \\ \text{const.} & \text{for } z \ll 1 . \end{cases} \tag{7.6}$$

The asymptotic form for $z \ll 1$ is due to the fact that the resonance function given by Eq. (6.14) does not depend on Ω if the time interval T is short because the sine function in Eq. (6.14) has a broad peak for small T.

The above asymptotic forms can be also confirmed by Eq. (6.14) and using the constant DOS near $\omega_{\mathrm{F}} = 0$. From Eqs. (6.14) and (6.17), the explicit form of the resonance function is given by

$$E(\omega_{\lambda_0}, \omega, T) = \frac{1}{\hbar} \int \mathrm{d}\omega_{\lambda} \mathscr{D}(\omega_{\lambda}) |\langle \omega_{\lambda} | \hat{J} | \omega_{\lambda_0} \rangle|^2 \frac{\sin^2\{(\omega_{\lambda} - \omega_{\lambda_0} - \omega)T/2\}}{(\omega_{\lambda} - \omega_{\lambda_0} - \omega)^2} , \tag{7.7}$$

where the density of states $\mathscr{D}(\omega_{\lambda})$ is introduced by the definition $\sum_{\lambda} = L^d \int \mathrm{d}(\hbar\omega_{\lambda}) \mathscr{D}(\omega_{\lambda})$. Since the AC conductivity of an infinite system is expressed by Eq. (7.4) for a sufficiently large time interval T, Eq. (7.7) for $T(\omega_{\lambda_0} + \omega) \gg 1$ should be

$$\frac{E(\omega_{\lambda_0}, \omega, T)}{T} \propto \int \mathrm{d}\omega_{\lambda} (\omega_{\lambda} - \omega_{\lambda_0})^{\delta} \delta(\omega_{\lambda} - \omega_{\lambda_0} - \omega) \propto \omega^{\delta} . \tag{7.8}$$

We have used the fact that $\mathscr{D}(\omega_{\lambda})$ is nearly constant in the band center. For the short time interval T, the resonance width in Eq. (7.7) becomes wider than the bandwidth and one yields

$$\frac{E(\omega_{\lambda_0}, \omega, T)}{T} \propto T^{-\delta} . \tag{7.9}$$

From these two extreme cases, the scaling form of the resonance function, or equivalently the AC conductivity, obeys Eqs. (7.5) and (7.6).

7.2. Results for unitary and symplectic systems

In this section, we show the calculations of the dynamic exponents for 3D unitary and symplectic systems in terms of the finite-time scaling method. Unitary and symplectic systems are actually realized by a magnetic field and a spin–orbit interaction, respectively. For these systems, the Hamiltonian matrices become complex and quaternion real. Therefore, it is not easy to calculate $\sigma(\omega)$ by conventional methods.

The Hamiltonian of the system is given by

$$H = \sum_{m\sigma} \varepsilon_{m\sigma} |m\sigma\rangle\langle m\sigma| + \sum_{m\sigma,n\sigma'} t_{m\sigma,n\sigma'} |m\sigma\rangle\langle n\sigma'| \;, \tag{7.10}$$

where m and σ denote the lattice site and the spin, respectively. We again set the lattice constant to be unity and only the nearest neighbor coupling is taken into account. The on-site potentials $\{\varepsilon_m\}$ are uniformly distributed in the range $[-W/2, W/2]$. In the unitary case, $t_{m\sigma,n\sigma'}$ is given by

$$t_{m\sigma,n\sigma'} = t \exp(\mathrm{i}\phi_{mn})\delta_{\sigma\sigma'} \tag{7.11}$$

The Peierls phase factor ϕ_{mn} depends on the magnetic field and the gauge. In the symplectic case, the hopping energy is described by [103,104]

$$t_{m\sigma,m-k\sigma'} = t[\exp(-\mathrm{i}\theta\boldsymbol{\sigma}_k)]_{\sigma\sigma'}, \quad k \equiv \hat{x}, \hat{y}, \hat{z} \;, \tag{7.12}$$

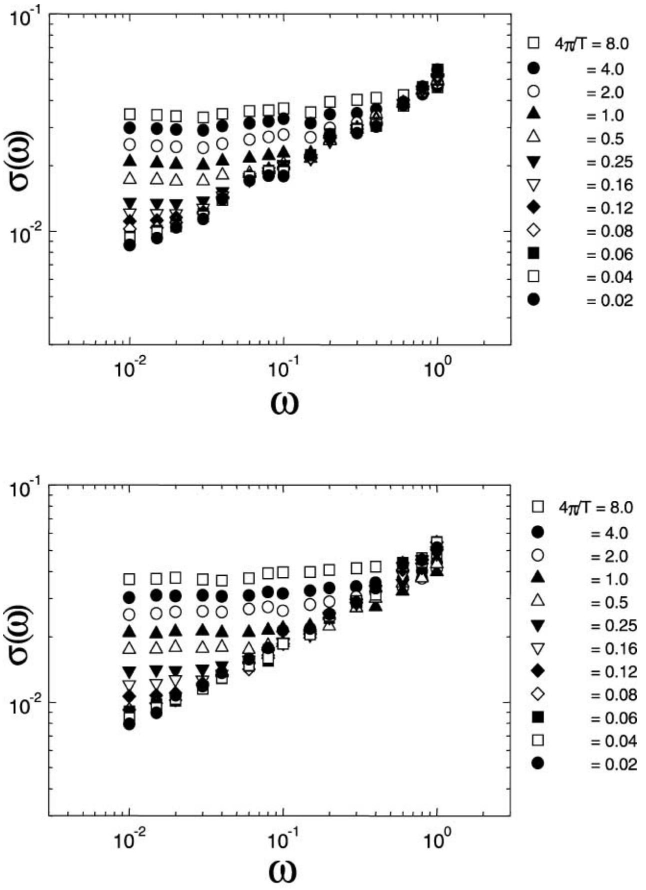
where $\boldsymbol{\sigma}_k$ are Pauli matrices. We choose the hopping amplitude t as the energy unit. The orthogonal system can be also described by Eq. (7.10) by taking $t_{m\sigma,n\sigma'} = t\delta_{\sigma\sigma'}$.

We set the disorder strength $W = W_c = 17.9$ [105] for the unitary case, assuming that a uniform magnetic field is applied parallel to the z-direction and the magnetic flux through a plaquette of the x–y plane is set to be 0.2 flux quanta. For this strength of the magnetic field, the Peierls phase in Eq. (7.11) becomes, in the Landau gauge, $2\pi/5$ for $n = m + \hat{x}$ and zero otherwise. For the symplectic case, we set $\theta = \pi/6$ in Eq. (7.12), and W is set to the critical value $W_c = 19.0$ [98]. The Fermi energy E_F is fixed to the band center. Actual calculations have been performed for systems with $30 \times 30 \times 30$ lattice sites for both cases. In each case, averaging over 20 independent realizations of random potentials has been performed.

Fig. 7.1 presents the calculated results of $\sigma(\omega)$ for both cases taking various time intervals $T = \pi/2 - 200\pi$ [79]. The corresponding resonance widths become $4\pi/T = 0.02 \sim 8.0$. We see from Fig. 7.1 that the calculated results follow the $\omega^{1/3}$-behavior with increasing time interval T over two orders of magnitude in frequency. Fig. 7.2 shows the scaling function $G(z)$ defined in Eq. (7.5). The most likely fit is determined by the χ^2-statistic, and confidence intervals for fitting parameters were estimated from the Bootstrap procedure [1]. The calculated results of the exponents are $\delta = 0.34 \pm 0.02$ for the unitary case and $\delta = 0.34 \pm 0.01$ for the symplectic case. These values agree well with the prediction of the scaling theory for the AC conductivity $\sigma(\omega)$ [83]. The errors in these exponents are much smaller than those in the results given in Fig. 6.3.

8. Extension to non-Hermitian matrices

The eigenvalue analysis for *non-Hermitian* matrices becomes important in many areas of condensed matter physics such as antiferromagnets [106,107], spin-glasses [108,109], electronic structure [110], the master equation in nonequilibrium thermodynamics [111], and the vector-mode analysis of optical waveguides [112–114]. The standard method for treating the eigenvalue problem of $N \times N$ non-Hermitian matrices is a diagonalization technique such as the QR method or the Arnoldi method [115,116]. These have, however, the serious drawback requiring a large amount of computer memory space, which makes it difficult to analyze *very large* non-Hermitian matrices. Another difficulty arises from, in general, the eigenvalues of non-Hermitian matrices



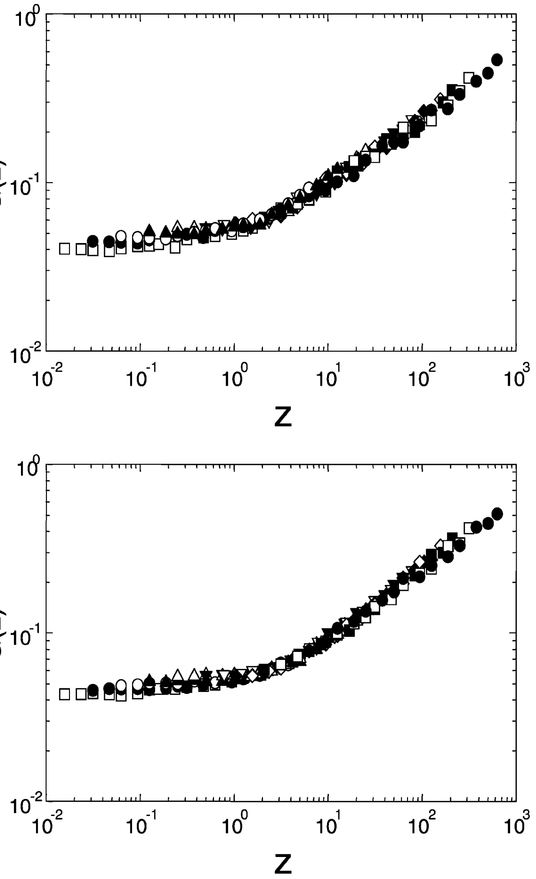


Fig. 7.1. AC conductivity $\sigma(\omega)$ for 3D (a) unitary and (b) symplectic systems for various time intervals T. The system size is taken as $N = 30^3$. From Ref. [79].

Fig. 7.2. Scaling function $G(z)$ introduced in Eq. (7.5) for 3D (a) unitary and (b) symplectic systems. The estimated values of dynamical exponents are $\delta = 0.34 \pm 0.02$ for unitary case, and $\delta = 0.34 \pm 0.01$ for symplectic case. From Ref. [79].

being sensitive to small changes in matrix elements. The difficulty is due to the lack of orthogonality among eigenvectors for non-Hermitian matrices. From these mathematical difficulties, practical algorithms have not yet been developed for the analysis of large non-Hermitian matrices. The basic idea of the FOM is applicable to large-scale non-Hermitian matrices. We describe here the algorithm analyzing large-scale non-Hermitian matrices.

8.1. Mapping onto lattice dynamical equations of motion

An asymmetric (as well as non-Hermitian) matrix has two sets of eigenvectors called the *right* eigenvector $\boldsymbol{u}(\lambda)$ defined by

$$\varepsilon_\lambda u_m(\lambda) = \sum_n D_{mn} u_n(\lambda) \tag{8.1}$$

and the *left* eigenvector $\boldsymbol{v}(\lambda)$ given by

$$\varepsilon_\lambda v_m(\lambda) = \sum_n v_n(\lambda) D^*_{nm} \ . \tag{8.2}$$

These eigenvectors belong to the same eigenvalue ε_λ. Though left (or right) eigenvectors do not form an orthogonal set themselves due to the asymmetric nature of the matrix $\{D_{mn}\}$, biorthogonality conditions are found between them [3,117]. These are written as

$$\sum_\lambda u_m(\lambda) v^*_n(\lambda) = \delta_{mn} \tag{8.3}$$

and

$$\sum_m u_m(\lambda) v^*_m(\lambda') = \delta_{\lambda\lambda'} \ . \tag{8.4}$$

The mapping of Eqs. (8.1) and (8.2) onto the equations of motion is done by, as in Eqs. (2.2) and (2.3),

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} x_m(t) = -\sum_n D'_{mn} x_n(t) \ , \tag{8.5}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} y_m(t) = -\sum_n D'_{nm}{}^* y_n(t) \ , \tag{8.6}$$

where D'_{mn} is defined as $D'_{mn} = D_{mn} + \delta_{mn}\varepsilon_0$. Since $\boldsymbol{u}(\lambda)$ forms a complete set of vectors [note that $\boldsymbol{u}(\lambda)$ does not form an orthogonal set, but they are *linearly independent*], the displacements $x_m(t)$ and $y_m(t)$ can be decomposed into a set of right and left eigenvectors $\boldsymbol{u}(\lambda)$ and $\boldsymbol{v}(\lambda)$ as

$$x_m(t) = \sum_\lambda Q_\lambda(t) u_m(\lambda) \ , \tag{8.7}$$

$$y_m(t) = \sum_\lambda R_\lambda(t) v_m(\lambda) \ . \tag{8.8}$$

8.2. Spectral density

We treat hereafter real asymmetric matrices $\{D_{mn}\}$. The generalization to non-Hermitian matrices with complex elements is straightforward. We introduce the quantity $E(t)$, corresponding to the energy function Eq. (2.6), defined by

$$\begin{aligned} E(t) &= \frac{1}{2}\left\{\sum_m \dot{x}_m(t)\dot{y}_m(t) + \sum_{mn} y_m(t) D'_{mn} x_n(t)\right\} \\ &= \frac{1}{2}\sum_\lambda \{\dot{Q}_\lambda(t)\dot{R}_\lambda(t) + \mu^2_\lambda Q_\lambda(t) R_\lambda(t)\} \ , \end{aligned} \tag{8.9}$$

where the biorthogonality condition Eq. (8.4) and $\mu_\lambda^2 \equiv \varepsilon_\lambda + \varepsilon_0$ are used. Using the quantities $\xi_\lambda(t)$ and $\eta_\lambda(t)$ defined by $\xi_\lambda(t) \equiv \dot{Q}_\lambda(t) + \mathrm{i}\mu_\lambda Q_\lambda(t)$ and $\eta_\lambda(t) \equiv \dot{R}_\lambda(t) + \mathrm{i}\mu_\lambda R_\lambda(t)$, Eq. (8.9) is rewritten as $E(t) = \frac{1}{2}\sum_\lambda \xi_\lambda^*(t)\eta_\lambda(t)$. Thus, if an external periodic force $F_m \cos(\Omega t)$ is applied to the system, one has as in the case of Eq. (2.12),

$$E(T) = \frac{1}{2}\sum_\lambda \left\{\sum_m F_m v_m(\lambda)\right\}\left\{\sum_n F_n u_n(\lambda)\right\}\frac{\sin^2\{(\mu_\lambda - \Omega)T/2\}}{(\mu_\lambda - \Omega)^2} . \tag{8.10}$$

The averaged value of $E(T)$ over ϕ_m becomes

$$\begin{aligned}\langle E(T)\rangle &= \frac{F_0^2}{2}\sum_\lambda \frac{\sin^2\{(\mu_\lambda - \Omega)T/2\}}{(\mu_\lambda - \Omega)^2}\left\langle \sum_{mn} v_m(\lambda)u_m(\lambda)\cos(\phi_m)\cos(\phi_n)\right\rangle \\ &= \frac{F_0^2}{2}\sum_\lambda \frac{\sin^2\{(\mu_\lambda - \Omega)T/2\}}{(\mu_\lambda - \Omega)^2} ,\end{aligned} \tag{8.11}$$

where $\langle \cdots \rangle$ denotes the random phase average and the terms satisfying $m = n$ remain in the summation for m and n. Provided that the proper time interval T is used, Eq. (8.11) yields

$$\langle E(\Omega, T)\rangle \approx \frac{\pi T F_0^2}{8}\sum_\lambda \delta(\mu_\lambda - \Omega) = \frac{\pi T N F_0^2}{8}\tilde{\mathscr{D}}(\Omega) , \tag{8.12}$$

where $\tilde{\mathscr{D}}(\Omega)$ is the density of states for the mapped system. The spectral density of the original matrix D_{mn} is given by Eq. (2.15).

8.3. Eigenvectors

Using Eqs. (8.5) and (8.7), the equation of motion for right eigenvectors $\boldsymbol{u}$ with the external force $F_m \cos(\Omega t)$ is written as

$$\sum_\lambda \left\{\frac{\mathrm{d}^2 Q_\lambda(t)}{\mathrm{d}t^2} + \mu_\lambda^2 Q_\lambda(t)\right\} u_m(\lambda) = F_m \cos(\Omega t) . \tag{8.13}$$

Multiplying the left eigenvector $v_m(\lambda')$ and taking the sum over m and λ in Eq. (8.13) under the condition Eq. (8.4), one obtains the equation for the amplitude $Q_\lambda(t)$ as

$$\frac{\mathrm{d}^2 Q_\lambda(t)}{\mathrm{d}t} + \mu_\lambda^2 Q_\lambda(t) = \sum_m \{F_m v_m(\lambda)\}\cos(\Omega t) . \tag{8.14}$$

Eq. (8.14) is solved with the initial condition $Q_\lambda(t = 0) = 0$ as

$$Q_\lambda(t) = \left\{\sum_m F_m v_m(\lambda)\right\}\frac{2\sin\{(\Omega + \mu_\lambda)t/2\}\sin\{(\Omega - \mu_\lambda)t/2\}}{\Omega^2 - \mu_\lambda^2} . \tag{8.15}$$

Using Eq. (8.7), the amplitude of $x_m(t)$ after the time interval T is

$$x_m(T) = \sum_\lambda \left\{\sum_m F_m v_m(\lambda)\right\}\frac{2\sin\{(\Omega + \mu_\lambda)t/2\}\sin\{(\Omega - \mu_\lambda)t/2\}}{\Omega^2 - \mu_\lambda^2} u_m(\lambda) . \tag{8.16}$$

Note that this equation corresponds to Eq. (2.17). After p iterations of the procedure described in Section 2.3 [see Eq. (2.20)], the amplitude $x_m(T)$ becomes

$$x_m^{(p)}(T) = \sum_\lambda \left\{ \sum_m F_m v_m(\lambda) \right\} \left[\frac{2\sin\{(\Omega + \mu_\lambda)t/2\} \sin\{(\Omega - \mu_\lambda)t/2\}}{\Omega^2 - \mu_\lambda^2} \right]^p u_m(\lambda) . \tag{8.17}$$

Only a single eigenmode λ_1 ($\mu_{\lambda_1} \approx \Omega$) survives for sufficiently large p, namely,

$$x_m^{(p)}(T) \approx C u_m(\lambda_1) , \tag{8.18}$$

where C is a constant.

The above argument ensures that eigenvectors of non-Hermitian matrices can be calculated in exactly the same way as that for Hermitian matrices. Therefore, the eigenvalue ε_λ for $\boldsymbol{u}(\lambda)$ is obtained by the procedure described in Section 4.1.

8.4. Dynamical properties of percolating antiferromagnets

In this section, we show an example of the eigenvalue analysis for non-Hermitian matrices, i.e., the spin-wave dynamics of *percolating* antiferromagnets. Spin-wave excitations on percolating Heisenberg antiferromagnets show peculiar properties originating from geometric disorder and self-similarity [118,119]. They possess characteristics of *fractons* as vibrational excitations on percolating networks (see Section 5.2). However, the symmetry of the matrix characterizing the equations of antiferromagnetic spin-wave motion is different from those for lattice vibrations or ferromagnetic spin waves [42]. Therefore, we expect that the essential nature of antiferromagnetic spin-wave fractons differs from that of vibrational ones. The Hamiltonian of a percolating Heisenberg antiferromagnet is described by

$$H = \sum_{mn} J_{mn} \boldsymbol{S}_m \cdot \boldsymbol{S}_n , \tag{8.19}$$

where $\boldsymbol{S}_m$ denotes the classical spin vector with magnitude S at the site m, and J_{mn} is the exchange coupling between the nearest-neighbor spins at the sites m and n. Each lattice point is occupied by a spin with the probability p (see Section 5.2). J_{mn} is chosen as $J_{mn} = 1$ if both sites m and n are occupied, and $J_{mn} = 0$ otherwise. The linearized equation of motion for spin deviation S_m^+ from the perfect Néel order is expressed, in units of $S/\hbar = 1$, by

$$\mathrm{i}\frac{\partial S_m^+}{\partial t} = \sigma_m \sum_n J_{mn}(S_n^+ + S_m^+) , \tag{8.20}$$

where $S_m^+ \equiv S_m^x + \mathrm{i}S_m^y$, and σ_m is taken to be $+1$ for the site m belonging to the up-spin sublattice and -1 to the down-spin sublattice. The same equation holds for $S_m^-(\equiv S_m^x - \mathrm{i}S_m^y)$. This equation has different symmetry from the equations of motion for ferromagnetic spin waves (or lattice vibrations with scalar displacements). The dynamical matrix element D_{mn} is given by $D_{mn} = \sigma_m J_{mn}$ for $m \neq n$ and $D_{mm} = \sigma_m \sum_n J_{mn}$. Note that the dynamical matrix $\boldsymbol{D}$ is *antisymmetric*, namely $D_{mn} = -D_{nm}$, due to the coefficient σ_m. See the details in Refs. [11,120–125].

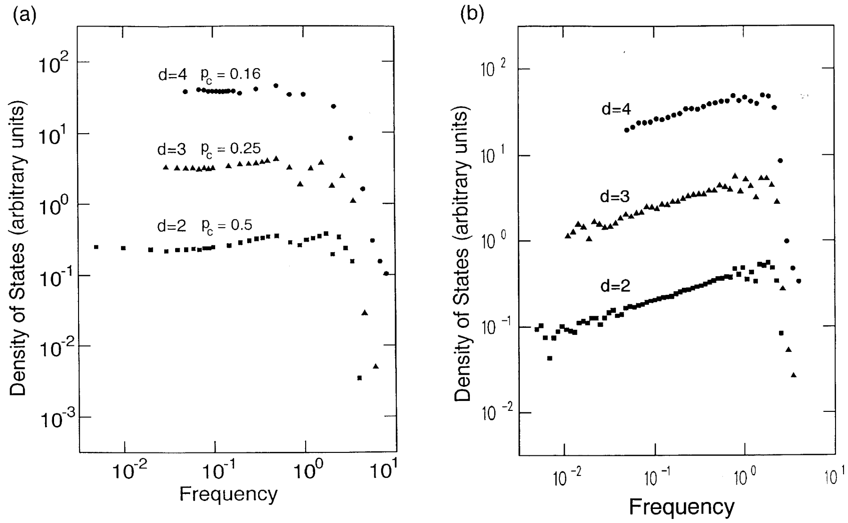


Fig. 8.1. (a) Densities of spin-wave states for 2D (squares), 3D (triangles), and 4D (circles) bond-percolating antiferromagnets at $p = p_c$. The results have been obtained by averaging over 1, 3, and 6 realizations of bond-percolating networks formed on 1100 × 1100, 100 × 100 × 100, and 28 × 28 × 28 × 28 hypercubic lattices for $d = 2$, $d = 3$, and $d = 4$, respectively. From Ref. [121]. (b) Densities of vibrational states for 2D (squares), 3D (triangles), and 4D (circles) bond-percolating networks at $p = p_c$. Conditions for calculations are the same with the case of (a). Fluctuations in densities of states at high frequencies do not represent computational errors, but indicate fine structures due to local configurations of atoms in single samples. From Ref. [120].

The results for the densities of spin wave states of bond-percolating networks at the critical concentrations (see Section 5.2) are shown in Fig. 8.1(a). System sizes are 1100^2, 100^3, and 28^4 for $d = 2$, 3, and 4, respectively. For any d, the densities of states are nearly constant at lower frequencies. This suggests that the fracton dimension $\tilde{d}_{\mathrm{AF}}$ for spin waves in percolating antiferromagnets is unity for any Euclidean dimensions, while $\tilde{d}$ for vibrational fractons becomes close to $\frac{4}{3}$ for any d [see Fig. 8.1(b)]. Since the fracton dimension is the key dynamic exponent as mentioned in Section 5.2, the fact $\tilde{d}_{\mathrm{AF}} \neq \tilde{d}$ implies that antiferromagnetic spin-wave fractons belong to a different class of universality from that for vibrational ones. The physical interpretation is given in Ref. [11].

Using the right and left eigenvectors, linear response functions of systems described by non-Hermitian matrices can be also computed by the FOM. Fig. 8.2 shows the dynamic structure factor $S(\boldsymbol{q}, \omega)$ of spin waves excited on 3D percolating antiferromagnets. Details of the algorithm are presented in Refs. [122–124]. The specific feature of $S(\boldsymbol{q}, \omega)$ obtained from these numerical calculations have been recently confirmed for percolating Heisenberg antiferromagnets $\mathrm{RbMn_{0.39}Mg_{0.61}F_3}$ in terms of inelastic neutron scattering experiments by Ikeda et al. [126].

There is another example of the eigenvalue problem of non-Hermitian matrices: The mode analysis in optical waveguides. The precise numerical analysis has become crucial to understand the propagation characteristics of light in waveguides for the optimum design of optical devices. Although a lot of efficient numerical methods for the eigenmode analysis of optical waveguides have been proposed [127–130], most of them have been performed under the scalar-wave

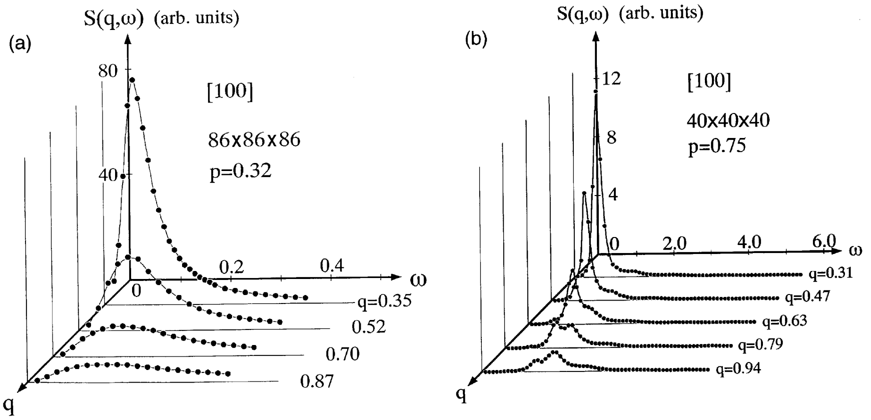


Fig. 8.2. (a) Dynamic structure factor $S(\boldsymbol{q},\omega)$ for 3D bond-percolating antiferromagnets at $p = 0.32$ ($p_c = 0.25$) formed on 86^3 cubic lattices. (b) Dynamic structure factor $S(\boldsymbol{q},\omega)$ for 3D bond-percolating antiferromagnets at $p = 0.75$ formed on 40^3 cubic lattices. The double-peak structure appears in this case. From Ref. [124].

approximation. The scalar-mode analysis is only valid for weakly guiding structures, but not accurate for waveguide structures with a large variation of refractive indices such as semiconductor waveguides. While [112–114] methods incorporating the vector nature of light waves are rigorous, these are usually complicated and require much computing time for practical calculations. This is because vector-mode analyses need to treat large-scale asymmetric (or non-Hermitian) matrices for achieving highly accurate calculations. We omit the detailed description on this subject. The reader who is interested in this can see Refs. [131,132].

9. Unstable oscillator method

In many-body problems, it is often necessary to calculate the ground state and the first few excited states and the corresponding eigenenergies of a very large-scale system. The Lanczos method is one of the most popular and powerful scheme for this purpose. Okamoto and Maris [20] have proposed an efficient method, the so-called unstable-oscillator method (UOM), for calculating the extreme (minimum or maximum) eigenvalue and the corresponding eigenvector of a large Hermitian or non-Hermitian matrix. The UOM is not appropriate for calculating arbitrary eigenvalues and eigenvectors in the central range of the spectrum as in the case of the FOM. It is, however, possible to compute extreme eigenvalues and eigenvectors much faster than in the case using the FOM. Although the method does not belong to the family of the FOM, it is deeply related to the FOM in the sense that a given matrix is mapped onto a lattice dynamical problem [20,21]. Therefore, we briefly describe the essence of the UOM in this section.

For a given Hermitian matrix $\boldsymbol{D}$, let us consider the corresponding equations of motion Eq. (2.2). In the FOM, we apply an external periodic force to the dynamical system [Eq. (2.9)], and solve Eq. (2.9) by the FEM. On the contrary, the UOM integrates the equations of motion Eq. (2.2) (the external-force free system) by the modified Euler method described in Section 3.1. Usually, the time step in the modified Euler method has to be taken as small as possible in order to minimize the error caused by discretizing time. The UOM chooses a time step τ not satisfying Eq. (3.7), and gives rise to the instability intentionally. Such a trick can provide a very effective way for the determination of the extreme eigenvalues.

Substituting Eq. (2.4) into Eq. (2.2) and using Eq. (2.1), we have the equation for the amplitude $Q_\lambda(t)$ as

$$\frac{\mathrm{d}^2 Q_\lambda(t)}{\mathrm{d}t} + \mu_\lambda^2 Q_\lambda(t) = 0 \; . \tag{9.1}$$

This equation is divided into a set of two equations,

$$\frac{\mathrm{d}}{\mathrm{d}t} P_\lambda(t) = -\mu_\lambda^2 Q_\lambda(t) \; ,$$

$$\frac{\mathrm{d}}{\mathrm{d}t} Q_\lambda(t) = P_\lambda(t) \; . \tag{9.2}$$

According to the argument in Section 3.1 [Eq. (3.4)], the modified Euler method yields the discretized coupled equations [12],

$$\begin{aligned} P_\lambda(l+1) &= P_\lambda(l) - \mu_\lambda^2 \tau Q_\lambda(l) \; , \\ Q_\lambda(l+1) &= Q_\lambda(l) + \tau P_\lambda(l+1) \; , \end{aligned} \tag{9.3}$$

where the time t is $l\tau$ with integer l. These equations lead to the relation

$$Q_\lambda(l+1) - (2 - \mu^2\tau^2) Q_\lambda(l) + Q_\lambda(l-1) = 0 \; . \tag{9.4}$$

Assuming $Q_\lambda(l) = (\beta_\lambda)^l$ with a constant β_λ, one has

$$\beta_\lambda^\pm = \frac{2 - \mu_\lambda^2\tau^2 \pm \sqrt{\mu_\lambda^2\tau^2(\mu_\lambda^2\tau^2 - 4)}}{2} \; . \tag{9.5}$$

The general solution for the amplitude is then given by

$$Q_\lambda(l) = c_\lambda^+ (\beta_\lambda^+)^l + c_\lambda^- (\beta_\lambda^-)^l \; , \tag{9.6}$$

where c_λ^+ and c_λ^- are to be determined by the initial conditions. The behavior of the solution is classified into two cases:

(A) $|\beta_\lambda| = 1$, for $0 \le \mu_\lambda\tau \le 2$,

(B) $|\beta_\lambda| \ne 1$, for $\mu_\lambda\tau > 2$.

This provides the basis of the UOM. Amplitudes of modes with $\mu_\lambda > 2/\tau$ grow *exponentially* with increasing time, while modes with $\mu_\lambda < 2/\tau$ oscillate stably. Thus, choosing the time step τ so that all eigenfrequencies μ_λ except for the maximum one $\mu_{\lambda_{\max}}$ satisfy the condition $\mu_\lambda\tau \le 2$, one can extract the growing extreme eigenvector $\boldsymbol{e}(\lambda_{\max})$.

The optimum time step τ can be determined from the following potential energy,

$$U = \frac{1}{8}\sum_{m,n} x_m(l) D'_{mn}\{x_n(l+1) + 2x_n(l) + x_n(l-1)\} \; , \tag{9.7}$$

where $x_m(l)$ is the displacement of the mth atom at $t = l\tau$ under the initial conditions of $x_m(0) = \phi_m$ and $\dot{x}_m(0) = 0$, where ϕ_m is a random variable. This yields [20]

$$U = \frac{1}{8}\sum_{\lambda} \mu_{\lambda}^2(4 - \mu_{\lambda}^2\tau^2)Q_{\lambda}^2(l) . \tag{9.8}$$

The time step τ to obtain only the mode having the maximum eigenfrequency is the smallest time step at which U for a large l changes to a negative value. The eigenvalue corresponding to this extreme eigenvector can be calculated by Eq. (4.7). The purity of the mode is evaluated by δ given by Eq. (4.8).

The next highest eigenfrequency is obtained from the initial displacement vector $\{x_m(0)\}$ without the extreme-eigenvector component. This is done by the Gram–Schmidt method:

$$x_m(0) = \phi_m - e_m(\lambda_{\max})\sum_n \phi_n e_n^*(\lambda_{\max}) , \tag{9.9}$$

where ϕ_m is the random variable defining the initial displacement vector for obtaining $\boldsymbol{e}(\lambda_{\max})$. Under the initial condition of zero-velocity vector, the maximum frequency becomes the next highest eigenfrequency $\mu_{\lambda_{\max}-1}$ of the original system. Using the quantity U defined by Eq. (9.7), we determine the new time step for obtaining the next highest eigenfrequency $\mu_{\lambda_{\max}-1}$ and its eigenvector, and so on.

As in the case of the FOM, the algorithm requires the computing time proportional to N^2, where N is the system size. However, the proportionality coefficient for the UOM is much smaller than that for the FOM, implying that the UOM is faster than the FOM. This is because the extreme eigenvector grows *exponentially* in the UOM, while the amplitude of the resonating mode increases *linearly* in time in the FOM. The drawback of the UOM is that one can calculate only a few eigenvalues (and their eigenvectors) near the spectrum edge. Due to the fact that the extreme eigenvector obtained is not completely pure, the initial displacement vector defined by Eq. (9.9) slightly contains the $\mu_{\max}$ component $\boldsymbol{e}(\lambda_{\max})$. This makes it difficult to calculate eigenvectors $\boldsymbol{e}(\lambda)$ with μ_λ far from $\mu_{\lambda_{\max}}$. This method is easily extended to non-Hermitian matrices [131,132].

10. Conclusions

Computer simulation has become a very intensive area of science and engineering with the development of new algorithms, which are originated from a number of different subjects with their own problems and goals. Though such simulations are likely to play an important role in laboratory experiments, those are often unjustified due to the complexity of their numerical algorithms. The forced oscillator method (FOM) has provided a unique and highly efficient scheme. This method is very powerful for calculating the spectral density, eigenvalues, and corresponding eigenvectors of Hermitian or non-Hermitian matrices in addition to the linear response functions of classical and quantum systems. The superiority of the FOM lies in its computational efficiency and the ability to treat large dynamical systems containing up to 10^7 particles. The method has the following advantages: (i) the memory requirement is of the order of

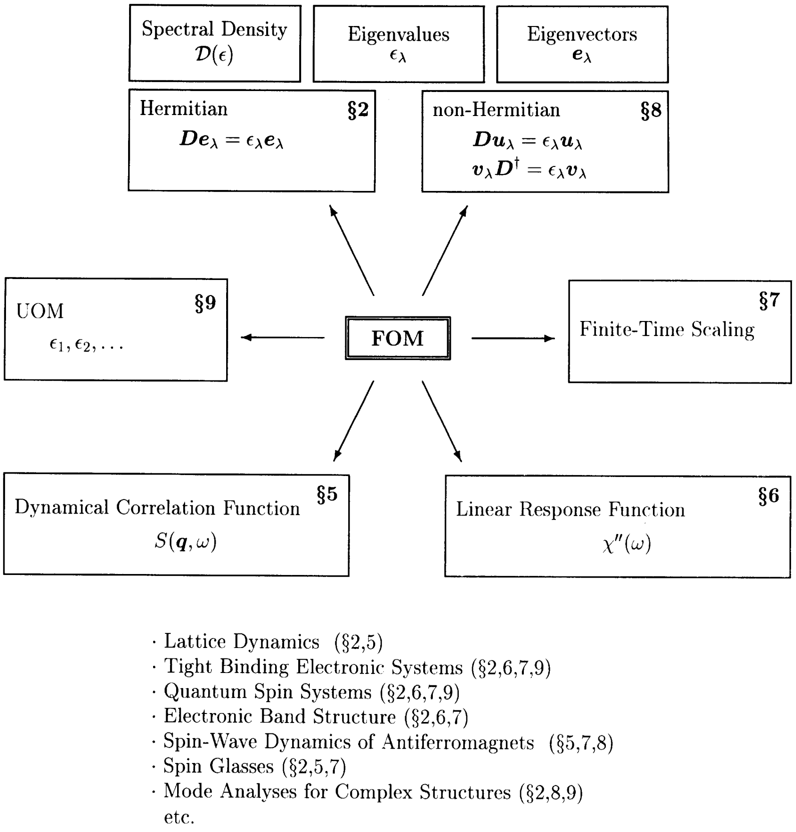

Spectral Density
$\mathcal{D}(\epsilon)$
Eigenvalues
ϵ_λ
Eigenvectors
$\boldsymbol{e}_\lambda$
Hermitian §2
$\boldsymbol{D}\boldsymbol{e}_\lambda = \epsilon_\lambda \boldsymbol{e}_\lambda$
non-Hermitian §8
$\boldsymbol{D}\boldsymbol{u}_\lambda = \epsilon_\lambda \boldsymbol{u}_\lambda$
$\boldsymbol{v}_\lambda \boldsymbol{D}^\dagger = \epsilon_\lambda \boldsymbol{v}_\lambda$
UOM §9
$\epsilon_1, \epsilon_2, \ldots$
FOM
§7
Finite-Time Scaling
Dynamical Correlation Function §5
$S(\boldsymbol{q}, \omega)$
Linear Response Function §6
$\chi''(\omega)$
· Lattice Dynamics (§2,5)
· Tight Binding Electronic Systems (§2,6,7,9)
· Quantum Spin Systems (§2,6,7,9)
· Electronic Band Structure (§2,6,7)
· Spin-Wave Dynamics of Antiferromagnets (§5,7,8)
· Spin Glasses (§2,5,7)
· Mode Analyses for Complex Structures (§2,8,9)
etc.

Fig. 10.1. Overview of developments and extensions of the forced oscillator method (FOM). Number after section at the right corner represents the related section.

N for sparse matrices, (ii) it is very suitable for implementations in parallel and vector processing supercomputers, (iii) the computations for the spectral density can be made within an arbitrary range of eigenvalues and with a given resolution, and the computing time scales linearly with the matrix size N, (iv) one can calculate quite accurately the specific eigenvalue and its eigenvector with a computing time proportional to N^2, and evaluate its accuracy and purity, and (iv) linear response functions are easily calculated in the context of the FOM. In particular, the efficiency of the FOM becomes remarkable when combined with the FEM [14–19].

The FOM has been successfully applied to a number of problems such as, for *Hermitian* matrices, dynamics of fractal and glassy systems [43,120,133–147], photon localization [148,149], electronic states in mesoscopic systems [9,79,88,150,151], electronic structures of amorphous systems [152,153], and $\pm J$ Ising spin glasses [102], and for *non-Hermitian* matrices, spin-wave dynamics of antiferromagnets [11,121–123] and waveguide analysis [131,132,154]. This situation is illustrated in Fig. 10.1.

Finally, we would like to point out that the following list of references would not do justice to the entire area of eigenvalue analysis and computing linear response functions. Readers should follow their own interests by looking at reviews and books cited in the following references.

Acknowledgements

The authors are grateful to our colleagues, T. Terao, Y. Hobiki, H. Noro, and H. Shima, who have contributed over the years to the development of the FOM. This work is supported in part by a Grant-in-Aid for Scientific Research from the Japan Ministry of Education, Science, Sports, and Culture for Scientific Research. Numerical calculations were partially performed on the FACOM VPP 500 of the Supercomputer Center, Institute of Solid State Physics, University of Tokyo.

References

[1] See, for example, W.H. Press, S.A. Tenkolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortan 77: The Art of Scientific Computing, Cambridge University Press, Cambridge, 1997.

[2] J.K. Cullum, R.A. Willoughby, Lanczos, Algorithms for Large Symmetric Eigenvalue Computations, Birkhauser, Basel, 1985.

[3] See, for example, J. Cullum, R.A. Willoughby (Eds.), Large Scale Eigenvalue Problems, North-Holland, Amsterdam, 1986.

[4] For a review see, for example, E. Dagotto, Rev. Mod. Phys. 66 (1994) 763.

[5] A. Cordelli, G. Grosso, G.P. Parravicini, Comput. Phys. Commun. 83 (1995) 255.

[6] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer, New York, 1980.

[7] M.L. Williams, H.J. Maris, Phys. Rev. B 31 (1985) 4508.

[8] K. Yakubo, T. Nakayama, H.J. Maris, J. Phys. Soc. Japan 60 (1991) 3249.

[9] T. Nakayama, H. Shima, Phys. Rev. E 58 (1998) 3984.

[10] A.A. Maradudin, E.W. Montroll, G.H. Weiss, Theory of Lattice Dynamics in the Harmonic Approximation, Academic Press, New York, 1963.

[11] K. Yakubo, T. Terao, T. Nakayama, J. Phys. Soc. Japan 62 (1993) 2196.

[12] See, for example, D. Frenkel, B. Smit, Understanding Molecular Simulations, Academic Press, San Diego, 1996.

[13] See, for example, N.I. Danilia, N.S. Dubrovskaya, O.P. Kvasha, G.L. Smirnov, Computational Mathematics, Mir Publishers, Moscow, 1988.

[14] H. Tal-Ezer, R. Kosloff, J. Chem. Phys. 81 (1984) 3967.

[15] R. Kosloff, H. Tal-Ezer, Chem. Phys. Lett. 127 (1986) 223.

[16] R. Kosloff, J. Phys. Chem. 92 (1988) 2087.

[17] R. Kosloff, Annu. Rev. Phys. Chem. 45 (1994) 145.

[18] R. Chen, H. Guo, Comput. Phys. Commun. 119 (1999) 19.

[19] Y.L. Loh, S.N. Taraskin, S.R. Elliott, Phys. Rev. Lett. 84 (2000) 2290.

[20] Y. Okamoto, H.J. Maris, Comput. Phys. Commun. 76 (1993) 191.

[21] A. Mitsutake, T. Iitaka, Y. Okamoto, Comput. Phys. Commun. 96 (1996) 217.

[22] L. Verlet, Phys. Rev. 159 (1967) 98.

[23] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[24] J.R. Rice, Numerical Methods, Software, and Analysis, McGraw-Hill, New York, 1983.

[25] R.J. Goult, R.F. Hoskins, J.A. Milner, M.J. Pratt, Computational Methods in Linear Algebra, Wiley, New York, 1974.

[26] G.H. Golub, C.F. van Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, MD, 1996.

[27] L. Fox, I.B. Parker, Chebyshev Polynomials in Numerical Analysis, Oxford University Press, London, 1968.

[28] T.J. Rivlin, The Chebyshev Polynomials, Pure and Applied Mathematics: A Wiley-Interscience Series of Monographs, and Tracts, Wiley, New York, 1974.

[29] T. Terao, T. Nakayama, Phys. Rev. B 53 (1996) R2918.

[30] W. Marchall, S.W. Lovesey, Theory of Thermal Neutron Scattering, Oxford University Press, Oxford, 1971.

[31] W.J.L. Buyers, D.E. Pepper, R.J. Elliot, J. Phys. C 5 (1972) 2611.

[32] S. Kirkpatrick, A.B. Harris, Phys. Rev. B 12 (1975) 2480.
[33] D. Stauffer, A. Aharony, Introduction to Percolation Theory, Taylor & Francis, London, 1992.
[34] M. Sahimi, Application of Percolation Theory, Taylor & Francis, London, 1994.
[35] H.E. Stanley, J. Phys. A 10 (1977) L211.
[36] B.I. Shklovskii, A.L. Efros, Electronic Properties of Doped Semiconductors, Springer, Berlin, 1984.
[37] S. Havlin, D. Ben-Avraham, Adv. Phys. 36 (1987) 695.
[38] M.B. Ishichenko, Rev. Mod. Phys. 64 (1992) 961.
[39] G. Dietler, C. Aubert, D.S. Cannell, P. Wiltzius, Phys. Rev. Lett. 57 (1986) 3117.
[40] G. Dietler, C. Aubert, D.S. Cannell, P. Wiltzius, Phys. Rev. Lett. 59 (1987) 246.
[41] F. Family, D.P. Landau, Kinetics of Aggregation and Gelation, North-Holland, Amsterdam, 1984.
[42] S. Alexander, R. Orbach, J. Phys. (Paris) Lett. 43 (1982) L625.
[43] T. Nakayama, K. Yakubo, R. Orbach, Rev. Mod. Phys. 66 (1994) 381.
[44] S. Alexander, J. Bernasconi, W. Schneider, R. Orbach, Rev. Mod. Phys. 53 (1981) 175.
[45] Y. Gefen, A. Aharony, S. Alexander, Phys. Rev. Lett. 50 (1983) 77.
[46] S. Alexander, E. Courtens, R. Vacher, Physica A 195 (1993) 286.
[47] M. Montagna, O. Pilla, G. Viliani, V. Mazzacurati, G. Ruocco, G. Signorelli, Phys. Rev. Lett. 65 (1990) 1136.
[48] O. Pilla, G. Viliani, M. Montagna, V. Mazzacurati, G. Ruocco, G. Signorelli, Phil. Mag. B 65 (1992) 243.
[49] V. Mazzacurati, M. Montagna, O. Pilla, G. Viliani, G. Ruocco, G. Signorelli, Phys. Rev. B 45 (1992) 2126.
[50] E. Stoll, M. Kolb, E. Courtens, Phys. Rev. Lett. 68 (1992) 2472.
[51] O. Pilla, G. Viliani, R. Dell'Anna, R. Ruocco, Physica A 247 (1997) 23.
[52] A. Rahmani, C. Benoit, R. Jullien, G. Poussiguer, A. Sakout, J. Phys.: Condens. Matter 9 (1997) 2149.
[53] R. Kubo, J. Phys. Soc. Japan 12 (1957) 570.
[54] A.D. Greenwood, Proc. Phys. Soc. London 71 (1958) 585.
[55] E.R. Gagliano, C.A. Balseiro, Phys. Rev. Lett. 59 (1987) 2999.
[56] See, for example, V.S. Viswanath, G. Mueller, The Recursion Method, Springer, Heidelberg, 1994.
[57] R.E. Wyatt, Phys. Rev. E 51 (1995) 3643.
[58] H. Tanaka, W. Kunishima, M. Ito, RIKEN Review 29 (2000) 20.
[59] C. Benoit, J. Phys.: Condens. Matter 6 (1994) 3137.
[60] L.R. Mead, N. Papanicolaou, J. Math. Phys. (N.Y.) 25 (1984) 2404.
[61] R.H. Brown, A.E. Carlsson, Phys. Rev. B 32 (1985) 6125.
[62] D.A. Drabold, O.F. Sankey, Phys. Rev. Lett. 70 (1993) 3631.
[63] H. Roeser, H. Fehshkle, R.N. Silver, Europhys. Lett. 28 (1994) 257.
[64] L.W. Wang, Phys. Rev. Lett. 73 (1994) 1039.
[65] S. Goedecker, L. Colombo, Phys. Rev. Lett. 73 (1994) 122.
[66] O.F. Sankey, D.A. Drabold, A. Gibson, Phys. Rev. B 50 (1994) 1376.
[67] R.N. Silver, H. Röder, A.H. Voter, J.D. Kress, J. Comp. Phys. 124 (1996) 115.
[68] A.F. Voter, J.D. Kress, R.N. Silver, Phys. Rev. B 53 (1996) 12733.
[69] R.N. Silver, H. Röder, Phys. Rev. E 56 (1997) 4822.
[70] D. Mayou, Europhys. Lett. 6 (1988) 549.
[71] S. Roche, D. Mayou, Phys. Rev. Lett. 79 (1997) 2518.
[72] M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias, J.D. Joannopoulos, Rev. Mod. Phys. 64 (1992) 1045.
[73] S. Nomura, X.W. Zhao, Y. Aoyagi, T. Sugano, Phys. Rev. B 54 (1996) 13974.
[74] K. Michielsen, H. De Raedt, Europhys. Lett. 34 (1996) 435.
[75] T. Iitaka, S. Nomura, H. Hirayama, X. Zhao, Y. Aoyagi, T. Sugano, Phys. Rev. E 56 (1997) 1222.
[76] S. Nomura, T. Iitaka, X. Zhao, T. Sugano, Y. Aoyagi, Phys. Rev. B 56 (1997) R4348.
[77] T. Iitaka, T. Ebisuzaki, Phys. Rev. E 61 (2000) R3314.
[78] H. Shima, T. Nakayama, J. Phys. Soc. Japan 67 (1998) 2189.
[79] H. Shima, T. Nakayama, Phys. Rev. B 60 (1999) 14066.
[80] P.W. Anderson, Phys. Rev. 109 (1958) 1492.
[81] E. Abrahams, P.W. Anderson, D.C. Licciardello, T.V. Ramakrishnan, Phys. Rev. Lett. 42 (1979) 673.
[82] See, for example, N. Kramer, A. Mackinnon, Rep. Prog. Phys. 56 (1993) 1469.

[83] F.J. Wegner, Z. Phys. B 25 (1976) 327.
[84] J.T. Chalker, G.J. Daniell, Phys. Rev. Lett. 61 (1988) 593.
[85] S. Hikami, A. Larkin, Y. Nagaosa, Prog. Theor. Phys. 63 (1980) 707.
[86] K.B. Efetov, Adv. Phys. 32 (1983) 53.
[87] A. Mackinnon, B. Kramer, Z. Phys. B 53 (1983) 1.
[88] K. Yakubo, Y. Goto, Phys. Rev. B 54 (1996) 13432.
[89] K. Slevin, T. Ohtsuki, Phys. Rev. Lett. 78 (1997) 4083.
[90] T. Kawarabayashi, T. Ohtsuki, Phys. Rev. B 51 (1995) 10897.
[91] T. Kawarabayashi, T. Ohtsuki, Phys. Rev. B 53 (1996) 6975.
[92] T. Ohtsuki, T. Kawarabayashi, J. Phys. Soc. Japan 66 (1997) 314.
[93] B. Huckestein, R. Klesse, Phys. Rev. B 59 (1999) 9714.
[94] Y. Huo, R.N. Bhatt, Phys. Rev. Lett. 68 (1992) 1375.
[95] B.I. Shklovskii, B. Shapiro, B.R. Sears, P. Lambrianides, H.B. Shore, Phys. Rev. B 47 (1993) 11487.
[96] E. Hofstetter, M. Schreiber, Phys. Rev. B 49 (1994) 14726.
[97] M. Batsch, L. Schweitzer, I.Kh. Zharekeshev, B. Kramer, Phys. Rev. Lett. 77 (1996) 1552.
[98] T. Kawarabayashi, T. Ohtsuki, K. Slevin, Y. Ono, Phys. Rev. Lett. 77 (1996) 3593.
[99] A. Mackinnon, B. Kramer, Phys. Rev. Lett. 47 (1981) 1546.
[100] B. Kramer, K. Broderix, A. Mackinnon, M. Schreiber, Physica A 167 (1990) 163.
[101] P. Lambrianides, H.B. Shore, Phys. Rev. B 50 (1994) 7268.
[102] K. Hukushima, K. Nemoto, J. Phys. Soc. Japan 64 (1995) 1863.
[103] T. Ando, Phys. Rev. B 40 (1989) 5325.
[104] K. Slevin, J.-L. Pichard, P.A. Mello, J. Phys. (Paris) 6 (1996) 529.
[105] M. Hemeke, B. Kramer, T. Ohtsuki, Europhys. Lett. 27 (1994) 389.
[106] G.J. Hu, D.L. Huber, Phys. Rev. B 33 (1986) 3599.
[107] D.L. Huber, W.Y. Ching, Phys. Rev. B 47 (1993) 3220.
[108] L.R. Walker, R.E. Walstedt, Phys. Rev. B 22 (1980) 3816.
[109] I. Avgin, D.L. Huber, Phys. Rev. B 48 (1993) 13625.
[110] See, for example, R. Haydock, in: H. Ehrenreich, F. Seitz, D. Turnbull (Eds.), Solid State Physics, Academic Press, New York, 1980.
[111] N.G. van Kampen, Stochastic Process in Physics and Chemistry, North-Holland, Amsterdam, 1992.
[112] M.S. Stern, IEE Proc. Pt. J 135 (1988) 56.
[113] P.L. Liu, S.L. Yang, D.M. Yuan, IEEE J. Quantum Electron. 29 (1993) 1205.
[114] W.P. Huang, C.L. Xu, IEEE J. Quantum Electron. 29 (1993) 2639.
[115] See, for example, J.H. Wilkinson, C. Reinsch, Linear Algebra, Springer, Berlin, 1971.
[116] See, for example, F. Chatelin, Valeurs Propres de Matrices, Masson, Paris, 1988.
[117] See, for example, L.E. Reichl, Modern Course in Statistical Physics, University of Texas Press, Austin, 1980.
[118] R. Orbach, K.W. Yu, J. Appl. Phys. 61 (1987) 3689.
[119] G. Polatsek, O. Entin-Wohlman, R. Orbach, Phys. Rev. B 39 (1989) 9353.
[120] T. Nakayama, Physica A 191 (1992) 386.
[121] K. Yakubo, T. Terao, T. Nakayama, J. Phys. Soc. Japan 63 (1994) 3431.
[122] T. Terao, K. Yakubo, T. Nakayama, Phys. Rev. E 50 (1994) 566.
[123] T. Terao, K. Yakubo, T. Nakayama, Phys. Rev. B 49 (1994) 12281.
[124] T. Terao, T. Nakayama, Phys. Rev. B 51 (1995) 11479.
[125] T. Nakayama, in: D.W. Hone (Ed.), Proceedings of the R.L. Orbach Symposium: 35 Years of Condensed Matter and Related Physics, World Scientific, Singapore, 1996, pp. 43–62.
[126] H. Ikeda, S. Ito, M.A. Adams, J.A. Fernandez-Baca, J. Phys. Soc. Japan 67 (1998) 3376.
[127] M.J. Robertson, S. Ritchie, P. Dayan, IEE Proc. Pt. J. 132 (1985) 336.
[128] M.D. Feit, J.A. Fleck Jr., J. Opt. Soc. Am. A 7 (1990) 73.
[129] W.P. Huang, H.A. Haus, J. Lightwave Technol. 9 (1991) 56.
[130] A.P. Ansbro, I. Montrosset, IEE Proc. Pt. J. 140 (1993) 253.
[131] H. Noro, T. Nakayama, J. Lightwave Tech. 14 (1996) 1546.

[132] H. Noro, T. Nakayama, J. Opt. Soc. Am. A 14 (1997) 1451.
[133] K. Yakubo, T. Nakayama, Phys. Rev. B 36 (1987) 8933.
[134] K. Yakubo, T. Nakayama, Phys. Rev. B 40 (1989) 517.
[135] K. Yakubo, T. Nakayama, J. Phys. Soc. Japan 58 (1989) 1504.
[136] T. Nakayama, K. Yakubo, R. Orbach, J. Phys. Soc. Japan 58 (1989) 1891.
[137] K. Yakubo, E. Courtens, T. Nakayama, Phys. Rev. B 42 (1990) 1078.
[138] K. Yakubo, K. Takasugi, T. Nakayama, J. Phys. Soc. Japan 60 (1990) 3249.
[139] T. Nakayama, K. Yakubo, J. Phys. Soc. Japan 61 (1992) 2601.
[140] Y. Hobiki, K. Yakubo, T. Nakayama, Phys. Rev. B 52 (1995) R1310.
[141] Y. Hobiki, K. Yakubo, T. Nakayama, Phys. Rev. E 54 (1996) 1997.
[142] T. Nakayama, N. Sato, J. Phys.: Condens. Matter 10 (1998) L41.
[143] T. Nakayama, J. Phys. Soc. Japan 68 (1999) 3540.
[144] T. Terao, A. Yamaya, T. Nakayama, Phys. Rev. E 57 (1998) 4426.
[145] T. Nakayama, Phys. Rev. Lett. 80 (1998) 1244.
[146] T. Nakayama, Physica B 263–264 (1999) 243.
[147] T. Terao, T. Nakayama, Physica B 263–264 (1999) 317.
[148] M. Takano, K. Yakubo, T. Nakayama, Jpn. J. Appl. Phys. 31 (1992) L839.
[149] T. Nakayama, K. Yakubo, M. Takano, Phys. Rev. B 47 (1993) 9249.
[150] T. Terao, K. Yakubo, T. Nakayama, J. Phys. Soc. Japan 61 (1992) 2173.
[151] K. Yakubo, M. Ono, Phys. Rev. B 58 (1998) 9767.
[152] H. Tanaka, T. Fujiwara, Phys. Rev. B 49 (1994) 11440.
[153] S. Yamamoto, T. Fujiwara, Phys. Rev. B 51 (1995) 8841.
[154] T. Nakayama, M. Takano, K. Yakubo, T. Yamanaka, Opt. Lett. 17 (1992) 326.
[155] D.W. Bullet, R. Haydock, V. Heine, M.J. Kelly, in: H. Ehrenreich, F. Seitz, D. Turnbull (Eds.), Solid State Physics, Vol. 35, Academic Press, New York, 1980.
[156] W. Zhu, Y. Huang, D.J. Kouri, C. Chandler, D.K. Hoffman, Chem. Phys. Lett. 217 (1994) 73.
[157] W. Kunishima, M. Ito, H. Tanaka, Prog. Theor. Phys. Suppl. 138 (2000) 149.