1 Stochastic Optimization Method for Analytic Continuation: when *a priori* Knowledge is Missing

A.S. Mishchenko
RIKEN Advanced Science Institute (ASI)
Cross-Correlated Materials Research Group (CMRG),
Wako 351-0198, Japan

Contents

1 Difficulties to solve ill-posed problems 5
1.1 Discrete form of integral equation 5
1.2 Sawtooth noise instability 5

2 Methods to solve ill-posed problems 7
2.1 Tikhonov-Phillips regularization method 8
2.2 Maximum entropy method 9
2.3 Stochastic sampling methods 9
2.4 Stochastic optimization method: relation to other stochastic sampling approaches 10

3 Stochastic optimization method: general description 11
3.1 Deviation measure 11
3.2 Parametrization of particular spectra 12
3.3 General overview: obtaining particular solution and its sum 14
3.4 General features of elementary updates 15
3.5 Global updates 16
3.6 Final solution and refinement 17
3.7 Elementary updates of class I 18
3.8 Elementary updates of class II 19

4 Practical aspects of the method 20
4.1 Choosing the number of global updates F 21
4.2 Choosing the number of particular solutions L 22

5 Tests of SOM 23
5.1 Test of SOM for imaginary time representation 24
5.2 Test of SOM for Matsubara representation 26

6 General description 28
6.1 Objective function 28

7 Available kernels and their properties 28
7.1 One particle at zero temperature 28
7.2 Optical conductivity 29
7.3 Fermi statistics at finite T 30
7.4 Bose statistics at finite T 30
7.5 Matsubara representation 31

8 Manual for operating of the set of codes 32
8.1 Description of Fortran-90 and input files in the set 32
8.2 Output files after processing 33
8.3 Description of the main control file ”control.in” 34
8.4 Description of the file ”anzac 1.in” 38
8.5 Description of the file ”manupre.in” 38
8.6 Description of the file ”precision.in” 38
8.7 Description of the file ”brek.in” 39

9 Manual for preparing tests for various kernels 39
9.1 How to test 40

10 Tutorials 41
10.1 Kernel 0 41
10.2 Kernel 1 42
10.3 Kernel 2 45
10.4 Kernel 3 46
10.5 Kernel 4 46
10.6 Kernel 5 47

Obtaining dynamical properties from the Quantum Monte Carlo (QMC) simulations is notoriously difficult problem because QMC provides a limited number of values of a dynamical correlation function

$$\{G(m), m = 1, M\} \tag{1}$$

either at Matsubara frequencies $i\omega_m$ or at imaginary time points τ_m, whereas dynamical information is associated with a spectral function $A(\omega)$ depending on the continuous energy variable ω. The procedure of obtaining the dynamical correlation function $A(\omega)$ from the known set of values $G(m)$ is called *analytic continuation* . One of the most complete overview of this problem for the case when the set of values $\{G(m), m = 1, M\}$ is obtained from numeric calculations can be found in Ref. [1]. Generally, the procedure requires solving the Fredholm integral equation of the first kind [2]

$$G(m) = \int_{-\infty}^{\infty} d\omega \; \mathcal{K}(m,\omega) \; A(\omega) \; , \; m = 1, \ldots, M \; , \tag{2}$$

where $\mathcal{K}(m,\omega)$ is some known kernel which depends on what quantities are associated with $G(m)$ and $A(\omega)$.

One of numerous examples is when one wants to determine the Lehmann spectral function [3]. This function contains a lot of important information on quasiparticles. For example, Lehmann function is proportional to the spectral response observed in experiments on angle resolved photoemission spectroscopy (ARPES) [4].

A typical quantity calculated in QMC is $G(m) = G(\tau_m)$ which is called imaginary time Green function (GF)

$$G(\tau_m) = \langle \; T_\tau c(\tau_m) c^\dagger(0) \; \rangle \; . \tag{3}$$

Here T_τ is the time ordering operator and c is annihilation operator of a quasiparticle. The imaginary time GF satisfies the periodicity (anti-periodicity) relation

$$G(\tau + \beta) = \pm G(\tau) \tag{4}$$

with the period equal to inverse temperature $\beta = 1/T$. Here upper (lower) sing is for boson (fermion) operators. Hence, there is an equivalent representation given by the values of the Fourier transform $G(m) = \mathcal{G}(i\omega_m)$ of the imaginary time GF

$$\mathcal{G}(i\omega_m) = \int_0^\beta d\tau \; e^{i\omega_m \tau} \; G(\tau) \tag{5}$$

at Matsubara frequencies $i\omega_m$ equal to $(2m+1)i\pi/\beta$ $[2im\pi/\beta]$ for fermion [boson] operators [3]. The quantity $\mathcal{G}(i\omega_m)$ is the GF in the Mastubara representation. Indeed, there is inverse Fourier transform from the Matsubara representation to the imaginary time GF

$$G(\tau) = \frac{1}{\beta} \sum_{\omega_m} \; e^{-i\omega_m \tau} \; \mathcal{G}(i\omega_m) \; . \tag{6}$$

It can be shown [3] that in the case when QMC data for GF are obtained in the Matsubara representation (5) $G(m) = \mathcal{G}(i\omega_m)$, the kernel $\mathcal{K}(m,\omega) \equiv \mathcal{K}(i\omega_m,\omega)$ of the Eq. (2) is

$$\mathcal{K}(i\omega_m,\omega) = \pm \frac{1}{i\omega_m - \omega} , \tag{7}$$

where plus (minus) sign corresponds to boson (fermion) operators. On the other hand, if the QMC data are given in terms of imaginary time GF (3), the kernel $\mathcal{K}(m,\omega) \equiv \mathcal{K}(\tau_m,\omega)$ of the analytic continuation is

$$\mathcal{K}(\tau_m,\omega) = -\frac{\exp(-\tau_m\omega)}{\exp(-\beta\omega) \pm 1} , \tag{8}$$

where the positive (negative) sign is for fermion (boson) operators.

Another example is when the quantity of interest is the optical conductivity $\sigma(\omega)$ and the quantity supplied by QMC is the imaginary time current-current correlation function $G(m) = J(\tau_m)$. The kernel $\mathcal{K}(m,\omega) \equiv \mathcal{K}(\tau_m,\omega)$ in this case is

$$\mathcal{K}(\tau_m,\omega) = \frac{1}{\pi} \frac{\omega \exp(-\tau_m\omega)}{1 - \exp(-\beta\omega)} . \tag{9}$$

Indeed, the problem of solving the Fredholm equation of the first kind is encountered in many areas which are far from the particular problem of analytic continuation. For example, one has to solve equation of the same type to restore the thermodynamic properties of the quantum systems from QMC [5] or to recover the variety of impurity traps in organic materials from the ESR spectra [6, 7]. Moreover, similar equation has to be solved for medical X-ray and impedance tomography, image deblurring, and many other practical applications [8]. Indeed, because of notorious practical importance of the problem there is a long history of the vast amount of attempts to develop methods giving solutions for this class of equations. The main difficulty with the considered above type of equations is the following: they belong to the class of the *ill-posed* problems. The main characteristic feature of this class is that there is no unique solution in the mathematical sense. Hence, to solve such equation, one has to introduce some additional information specifying what kind of solution is expected. Therefore, it is impossible to single out the best method for solving this class of equations because each specific problem requires its own approach.

In the following we give a historical, although incomplete, overview of the approaches invented to solve the Fredholm equations of the first kind and follow the developments of the methods up to the recent times. In Sec. 1, we introduce the most simple minded approach of the least-squares fit and show why it is not suitable for the *ill-posed* problems. We describe various approaches to the *ill-posed* problems in the Sec. 2. In particular, we discuss there Tikhonov-Phillips regularization method, maximum entropy method, and several variants of the stochastic sampling method. The stochastic optimization method (SOM), which is the main topic of this chapter, is compared with another stochastic sampling methods in Sec. 2.4. We give a detailed description of the SOM and some practical recipes in Secs. 3 and 4, respectively. Some tests of SOM are presented in Sec. 5 and some conclusions are given in Sec. 5.2.

1 Difficulties to solve ill-posed problems

To outline difficulties encountered in the problem of solving the Fredholm equation of the first kind it is convenient to transform the equation into its discrete analog. The discrete analog is a matrix equation which seems to be easily solvable by the least-squares fit. However, this naive approach immediately fails due to the *ill-posed* nature of the problem and the solution shows *sawtooth noise* instability. We introduce the discrete analog of the Fredholm equation in Sec. 1.1. The least-squares fit approach, the nature of the *ill-posed* problems, and the features of the *sawtooth noise* instability is described in Sec. 1.2.

1.1 Discrete form of integral equation

Approximating the spectral function by its values on a finite spectral mesh of N points

$$A(\omega) = \sum_{n=1}^{N} A(\omega_n)\delta(\omega - \omega_n) \; , \tag{10}$$

the integral equation (2) can be rewritten in a matrix form

$$G(m) = \sum_{n=1}^{N} \mathcal{K}(m, \omega_n) A(\omega_n) \; , \; m = 1, \ldots, M \; , \tag{11}$$

or equivalently presented as

$$\vec{G} = \widehat{\mathcal{K}} \vec{A} \; . \tag{12}$$

Here $\vec{G} \equiv (G(1), G(2), \ldots, G(M))$ [$\vec{A} \equiv (A(\omega_1), A(\omega_2), \ldots, A(\omega_N))$] is the M-dimensional [N-dimensional] vector and $\widehat{\mathcal{K}}(m,n) \equiv \mathcal{K}(m, \omega_n)$ is $M \times N$ matrix ($M \geq N$). The matrix $\widehat{\mathcal{K}}(m,n)$ is a known matrix depending on the kernel of integral equation, $\vec{A}$ is a vector to be found, and $\vec{G}$ is a vector obtained by QMC whose components are known with some error-bars.

1.2 Sawtooth noise instability

In practice, the problem expressed by Eq. (12) is usually *ill-posed* either because of non-existence or non-uniqueness of solution $\vec{A}$ [9]. A noise, which is always present in the given vector $\vec{G}$, leads to the situation when there is no solution $\vec{A}$ which exactly satisfies the Eq. (12). On the other hand there is an infinite number of solutions which make the left hand side of the Eq. (12) approximately equal to its right hand side. Therefore, one can not search for a unique vector $\vec{A}$ but has to find some solution which is the best in some sense or find a set of solutions which are good according to some sensible criterion. The above features are the fingerprints of the *ill-posed* problems.

The most simple minded approach in such a case is to search for the least-squares minimum-norm solution $\vec{A}$ which minimizes the *deviation measure* which is chosen in the form of the Euclidean residual norm

$$\parallel \widehat{\mathcal{K}} \vec{A} - \vec{G} \parallel^2 = \sum_{m=1}^{M} \left| \sum_{n=1}^{N} \mathcal{K}(m, \omega_n) A(\omega_n) - G(m) \right|^2 \; . \tag{13}$$

Indeed, one immediately surrenders to the *ill-posed* nature of the problem and tries to minimize, although never reach zero, the difference between left- and right-hand sides of the Eq. (12). Choosing the Euclidean norm one admits the absence of a unique solution because the approximate equality in the Eq. (12) can be defined in terms of an infinite number of another norms. For Euclidean norm one can find solution of the best least-squares fit in terms of singular value decomposition of the matrix $\widehat{\mathcal{K}}$ into a weighted sum of separable matrices [10]

$$\widehat{\mathcal{K}} = \sum_{i=1}^{r} \sigma_i \, \vec{u}_i \otimes \vec{v}_i^{\dagger} \; , \tag{14}$$

where $\vec{u} \otimes \vec{v} \equiv \vec{u}(k)\vec{v}(l)$ is a matrix determined by the outer product of left and right singular vectors of $\widehat{\mathcal{K}}$, $\vec{u}$ and $\vec{v}$, respectively. Real and nonnegative numbers $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r > 0$ are singular values of $\widehat{\mathcal{K}}$. The least-squares solution $\vec{A}$ is given by the explicit expression

$$\vec{A} = \sum_{i=1}^{r} \frac{\vec{u}_i^{\dagger} \otimes \vec{v}_i}{\sigma_i} \vec{G} \; . \tag{15}$$

The problem comes from small singular values σ_i. Even very small statistical errors in $\vec{G}$ induce large perturbations to solution $\vec{A}$. This perturbations, called *sawtooth noise*, are typical for *ill-posed* problems and look like fast oscillations with amplitude much larger than the actual solution $\vec{A}$. The reason for the *sawtooth noise* is that the solution $\vec{A}$ *over-fits* the statistical errors present in the input data $\vec{G}$. The *sawtooth noise* can even lead to large negative values of the otherwise positive actual solution $\vec{A}$. However, the least-squares fit under condition of non-negativity of the spectral function $\vec{A}$ results in the sawtooth noise too (see Fig. 1).

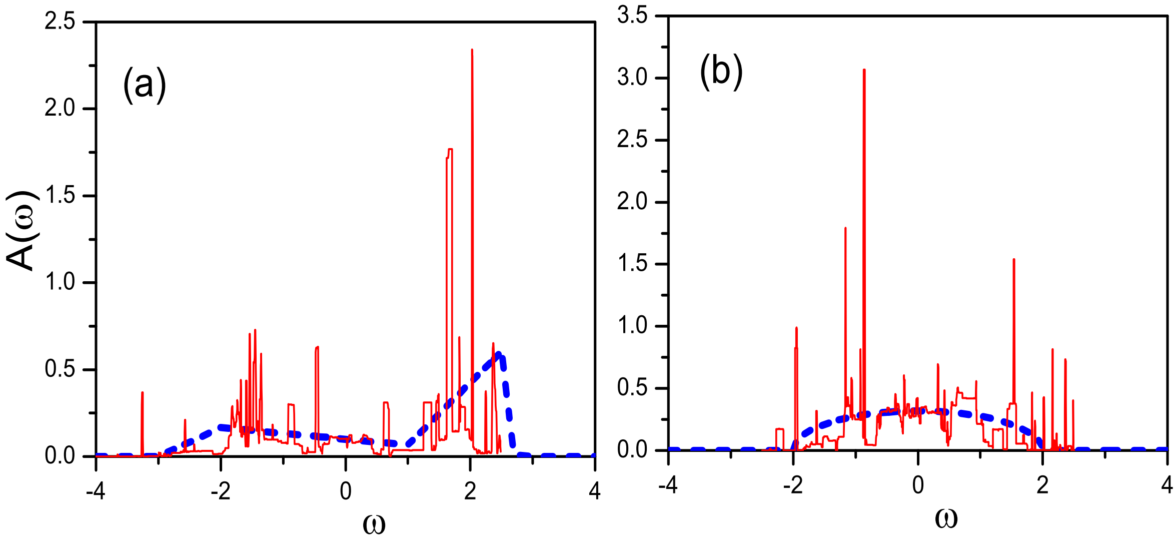

Fig. 1: *Examples of sawtooth noise in the least-squares fit with restricted positive spectral function. The amplitudes of spikes in non-regularized solutions (red solid line in panels (a) and (b)) are much larger than the actual value of the spectra (blue dashed lines in (a) and (b)).*

2 Methods to solve ill-posed problems

It was shown in the previous section that the least-squares fit approach to the matrix form (12) of the integral equation (2) leads to results with *sawtooth noise* which can be very far from the actual solution. The noise arises from the small singular values σ_i of the kernel of integral equation $\widehat{\mathcal{K}}$.

There are many methods developed to fight with this noise. The simplest and the most obvious one is truncated singular value decomposition when the terms in Eq. (15), which correspond to several smallest singular values σ_i, are neglected. However, the above trick is a particular example of a broad general class of approaches. The most of, if not all, methods to circumvent the problem of spurious noise in the solution of integral equation (2) can be united under the title *regularization methods*. As a particular example, the simplest regularization method for the matrix representation (12) of the integral equation (2) is based on the following trick. A regularization functional $\mathcal{F}(\vec{A})$, suppressing the oscillations of the solution $\vec{A}$, is added to the Euclidean norm (13)

$$\parallel \widehat{\mathcal{K}}\vec{A} - \vec{G} \parallel^2 + \gamma\mathcal{F}(\vec{A}) \tag{16}$$

and the *deviation measure* (16) is minimized instead. So, in general words the regularized solution is sought as a minimizer of the deviation measure which is a weighted combination of the residual norm $\parallel \widehat{\mathcal{K}}\vec{A} - \vec{G} \parallel^2$ and a side constraint $\gamma\mathcal{F}(\vec{A})$. Indeed, to construct the functional $\mathcal{F}(\vec{A})$ one needs some prior knowledge about solution $\vec{A}$.

The functional (16) is historically the very first approach, named Tikhonov-Phillips regularization method, which was developed to fight the sawtooth noise instability. However, to introduce a generic classifications of the regularization approaches, it is convenient to use Bayesian statistical inference. According to the Bayes's theorem [11]

$$P[A|G]\ P[G] = P[G|A]\ P[A]\ , \tag{17}$$

where $P[A|G]$ is the posterior or conditional probability that the spectral function is A provided the correlation function is G. Neglecting the normalization factor $P[G]$, which is independent of A, one gets

$$P[A|G] \sim P[G|A]\ P[A]\ , \tag{18}$$

where the ill-posed problem to find the most probable A given G is converted into much easier problem to find G given A, i.e. to maximize the *likelihood function* $P[G|A]$ taking into account simultaneously the *prior knowledge* about the spectrum $P[A]$. Note, any attempt to neglect the prior knowledge, i.e. to set $P[A] \equiv \text{const}$ and reduce the problem to minimization of just the likelihood function, leads to the sawtooth noise instability of the solution.

Notably, any method to regularize the *ill-posed* problem can be presented in the form of Bayesian approach and the distinctions between different approaches are restricted to choice of the likelihood function $P[G|A]$ and prior knowledge $P[A]$. Below we introduce different possibilities of this choice. We describe Tikhonov-Phillips regularization method in Sec 2.1, maximum entropy method in Sec. 2.2, and several variants of the stochastic sampling method in Sec. 2.3. We also consider the SOM as an effective example of stochastic sampling methods in Sec. 2.4.

2.1 Tikhonov-Phillips regularization method

Historically, the approach called *Tikhonov-Phillips regularization method* (TPRM) has been invented independently in many different contexts and became the first approach to solve the above problems. The name comes from the first applications of the ideas to the integral equations by A. N. Tikhonov [Tikhonoff] [12, 13] and D. L. Phillips [14] in early 40-es of the last century. Independently, the regularization approach was applied in different context to the discrete problem of matrix inversion [15–17] and know in the statistical literature as ridge regression. However, leaving aside the differences in terminology and interpretations, the general idea is the following.

In the sense of Bayes's inference the TPRM is a choice where the likelihood function is

$$P[G|A] \sim \exp\{- \parallel \widehat{\mathcal{K}}\vec{A} - \vec{G} \parallel^2\} \tag{19}$$

and the prior knowledge is

$$P[A] \sim \exp\{-\lambda^2 \parallel \hat{\Gamma}\vec{A} \parallel^2\} \ . \tag{20}$$

Thus, the *deviation measure* to minimize is the sum $\parallel \widehat{\mathcal{K}}\vec{A} - \vec{G} \parallel^2 + \lambda^2 \parallel \hat{\Gamma}\vec{A} \parallel^2$. Here, the likelihood function requires the least-squares fit of $\vec{G}$ while side constraint, where solution $\vec{A}$ is multiplied by a nonzero matrix $\hat{\Gamma}$, suppresses large absolute values of $A(\omega_k)$. Namely, the side constraint removes spikes and, hence, large values of derivatives in solution $[A(\omega_{k+1}) - A(\omega_k)]/[\omega_{k+1} - \omega_k]$.

The simplest modification of TPRM sets $\hat{\Gamma}$ as identity matrix $\hat{\Gamma} = \hat{I}$. In this case the expression (15) for solution $\vec{A}$ takes the form

$$\vec{A} = \sum_{i=1}^{r} \left\{ \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \right\} \frac{\vec{u}_i^{\dagger} \otimes \vec{v}_i}{\sigma_i} \vec{G} \ . \tag{21}$$

It is clear that contributions, corresponding to small singular values $\sigma_i \ll \lambda$, are automatically filtered out by the factors in the curly brackets and large sawtooth spikes of solution are suppressed. Then, the *over-fitting* of the noise in the input data is avoided by restricting the possible solution to the smooth ones. There are several approaches to find the optimal regularization parameter λ, L-curve [18, 19] and U-curve [20] methods in particular. These approaches consider relations between the Euclidean norm of solution $\parallel \hat{\Gamma}\vec{A} \parallel^2$ and residual $\parallel \widehat{\mathcal{K}}\vec{A} - \vec{G} \parallel^2$.

An interesting modification of the TPRM is given in [21,22]. The method expresses the solution $\vec{A}$ in terms of an average over a correlation matrix $< \tilde{\vec{A}}\tilde{\vec{A}}^{\dagger} >$ of possible solutions $\tilde{\vec{A}}$. The knowledge of this correlation matrix provides a prior knowledge about the solution.

There are other methods which are based on the suppression of the large derivatives of the solution. These methods are based on the form of functional (16) where the side constraint $\gamma\mathcal{F}(\vec{A})$ is explicitly taken in a form which suppresses large derivatives of the solution [23, 24]. A plenty of similar functionals can be found in earlier studies, see [9] for details of rigorous mathematical treatment of the *ill-posed* problems.

2.2 Maximum entropy method

One can criticize the first historical method to solve ill-posed problems as relying on unconditional smoothening of solution. The side constraint of the TPRM neglects solutions with large derivatives. This can be a problem when the spectral function has sharp edges or narrow peaks. One of recent approaches, *Maximum Entropy Method* (MEM) [1], provides an attractive strategy to circumvent some problems of the TPRM .

MEM searches for the most probable ”true” solution $A(\omega)$ among many possible particular solutions $\widetilde{A}(\omega)$ assuming prior knowledge that the ”true” solution $A(\omega)$ is close to a predefined function $D(\omega)$ called *default model*. The likelihood function of MEM is

$$P[G|\widetilde{A}] = \exp\{-\chi^2[\widetilde{A}]/2\}\ , \tag{22}$$

where

$$\chi^2[\widetilde{A}] = \sum_{m=1}^{M} \mathcal{E}^{-1}(m)[G(m) - \widetilde{G}(m)]^2\ , \tag{23}$$

and $\widetilde{G}(m)$ is related to particular solution $\widetilde{A}(\omega)$ through $\widetilde{G}(m) = \int_{-\infty}^{\infty} d\omega\ \mathcal{K}(m,\omega)\ \widetilde{A}(\omega)$. The matrix $\mathcal{E}(m)$ is set by noise in G data and related to covariance matrix. The prior knowledge function is defined as

$$P[G|A] = \exp\{\alpha^{-1}S[\widetilde{A}]\}\ , \tag{24}$$

where entropy

$$S[\widetilde{A}] = \int d\omega\ \widetilde{A}(\omega) \ln[\widetilde{A}(\omega)/D(\omega)] \tag{25}$$

characterizes deviation of a particular solution $\widetilde{A}(\omega)$ from the default model $D(\omega)$, a function that serves as the maximum entropy configuration. The regularization parameter α controls how much weight is given to minimization of the *deviation measure* $\chi^2[\widetilde{A}]$ and optimization of resemblance of solution $\widetilde{A}(\omega)$ to the default model $D(\omega)$.

The MEM is superior over TPRM in case when a lot of explicit information is known about $A(\omega)$. Moreover, one can avoid smoothening of large derivatives, typical for TPRM, given the knowledge about sharp parts of the ”true” solution $A(\omega)$. The nonphysical smoothening can be avoided if sharp parts of solution can be explicitly included into the default model. However, the method highly relies on the default model which can be a serious drawback if the most interesting features of the spectra are very sensitive to the form of the chosen default model [25].

2.3 Stochastic sampling methods

Any *stochastic sampling method* (SSM) uses a minimal prior knowledge about the solution, does not require any default model, and does not introduce any apparent smoothening of the solution. Characteristic feature of this class of methods is a change of the likelihood function $P[\widetilde{A}|G]$ into a likelihood functional (see e.g. [26])

$$A = \int d\widetilde{A}\ \widetilde{A}\ P[\widetilde{A}|G] \tag{26}$$

where ”true” solution A is obtained as an average of particular solutions $\widetilde{A}$ weighted by the likelihood function $P[\widetilde{A}|G]$. Optimal likelihood function has to prefer solutions $\widetilde{A}$ with small *deviation measure* $\chi^2[\widetilde{A}]$. Particular solution $\widetilde{A}$ with too small $\chi^2[\widetilde{A}]$ over-fits the data G and suffers from the sawtooth noise. However, although it is not proved formally, it is known practically that the sawtooth noise can be self-averaged in a sum of large enough number of solutions. One has to keep $\chi^2[\widetilde{A}]$ not too restrictive because the sawtooth noise persists if the most of solutions in the functional (26) over-fit the input data. The requirement to take into account solutions $\widetilde{A}$ with large enough $\chi^2[\widetilde{A}]$ sets up an *implicit regularization procedure*.

Starting from the very first practical design of SSM by Sandvik [27], the most of SSMs suggest the likelihood function in the form of Boltzmann distribution

$$\mathcal{P}[A|G] = \exp\{-\chi^2[\widetilde{A}]/\mathcal{T}\}, \tag{27}$$

where $\mathcal{T}$ is treated as a fictitious temperature and χ^2 is defined by Eq. (23) and can be considered as a fictitious energy. Then, because of the above interpretation, one can use the QMC, Metropolis algorithm [28] in particular, to sample possible functions $\widetilde{A}$. The prior knowledge function is usually defined by conditions that the spectral function $\widetilde{A}(\omega)$ is positively definite and that the first few known frequency moments are conserved.

In principle, although introducing many useful details, approaches suggested in Refs. [25, 29] belong to the same class as that introduced by Sandvik [27]. All three approaches do not use any default model for defining the prior knowledge function and, thus, are convenient in the problems where there is no much knowledge about how the resulting spectral function $A(\omega)$ has to look like.

Somehow different approach is suggested in the statistical MEM (SMEM) by Beach [30] and Jarrel [31]. The method defines a dimensionless field $n(x)$ which is related to default model $D(\omega)$. Then, the averaging is performed over the dimensionless field using the likelihood Boltzmann distribution Eq. (27). The useful feature of the method is that, depending on its parameters, it can interpolate between two limiting cases when the spectrum is completely governed by deviation measure (27) and when it is defined solely by default model.

2.4 Stochastic optimization method: relation to other stochastic sampling approaches

SOM [32], which is the main topic of this chapter, is a particular example of SSMs. SOM also does not use any default model, does not impose any apparent smoothening on solution, and restricts prior knowledge to normalization and positivity of the solution.

A particular feature of SOM, which singles it out among SSMs, is that the sampling of solutions, which optimize a deviation measure, is made without artificial interpretations of the likelihood function as a Boltzmann distribution [32, 33]. Similar idea was also suggested later in generic inversion via falsification of theories strategy [34]. Indeed, averaging of particular solutions (26), weighted by some likelihood function, has no any relation to any real partition function. To the contrary, one simply has to average over a set of particular solutions each of which fits the input data well enough. Therefore, interpretation of $\chi^2[\widetilde{A}]$ as an "energy" of some quantum state in a system with a given temperature $\mathcal{T}$ is a superfluous feature of the traditional SSMs. There is no real Hamiltonian and real temperature in the averaging procedure (26) and the net goal is to accumulate large enough number of solutions which fit, but not over-fit, the data set $G(m)$.

Hence, it does not matter how the set of averaged "good enough" solutions is found. This is why the strategy to find particular solutions in SOM is completely different from other SSMs. On every step SOM starts from an arbitrary chosen initial particular solution and minimizes its deviation measure until "good enough" fit is found. In this way SOM finds enough number of "good" particular solutions and calculates an average

$$A(\omega) = \sum_{j=1}^{L} \xi_j \widetilde{A}_j(\omega). \tag{28}$$

The simplest option is to set all coefficients equal to $\xi_j = 1/L$ for all L particular solutions whose deviation measure $\chi^2[\widetilde{A}]$ is smaller than some selected value. A detailed description of the SOM, i.e. how to organize the process and how to choose its optimal parameters, is given in Secs. 3 and 4.

3 Stochastic optimization method: general description

In comparison with other SSMs, the SOM uses slightly different measure $\chi^2[\widetilde{A}]$, considerably different way to parametrize a particular solution $\widetilde{A}_j(\omega)$, and completely different way to accumulate the particular spectral functions $\widetilde{A}_j(\omega)$ for averaging. An important feature of SOM is that it treats the energy space continuously without imposing any finite ω-mesh. We describe the deviation measure and parametrization of the spectra in Secs. 3.1 and 3.2, respectively. Sections 3.3 and 3.4 discuss the way to obtain particular solution and explain general features of elementary updates which decrease the deviation measure of particular solutions. Global updates and refinement of solution is considered in Secs. 3.5, and 3.6, respectively. Finally, elementary updates of classes I and II are described in Secs. 3.7 and 3.8.

3.1 Deviation measure

The first step is to define the *deviation measure* determining the criterion which solution is a good approximation of the input data set G. The set G corresponds to some QMC data on imaginary times $G(m) = G(\tau_m)$ or at some Matsubara frequencies $G(m) = \mathcal{G}(i\omega_m)$, $m = 1, \ldots, M$. Then the *deviation measure* of SOM is given by expression

$$D[\widetilde{A}] = \sum_{m=1}^{M} |\varDelta(m)| \, . \tag{29}$$

Here $\varDelta(m)$ is the *deviation function*

$$\varDelta(m) = \frac{G(m) - \tilde{G}(m)}{\mathcal{S}(m)}, \tag{30}$$

which characterizes individual deviations of specific data points $G(m)$ from the values of the simulated function $\widetilde{G}(m)$ defined by the particular spectral function $\widetilde{A}$ in terms of relation

$$\widetilde{G}(m) = \int_{-\infty}^{\infty} d\omega \; \mathcal{K}(m,\omega) \; \widetilde{A}(\omega) \; . \tag{31}$$

The factors $\mathcal{S}(m)$ can be chosen as error-bars of the QMC data $G(m)$, if the error-bars are known.

In many of standard approaches the deviation measure is called ”*objective*” and has the form

$$O[\widetilde{A}] = \frac{1}{M} \sum_{m=1}^{M} [\mathcal{D}(m)]^2 \; . \tag{32}$$

Here $\mathcal{D}(m)$ is the *deviation function*

$$\mathcal{D}(m) = \frac{G(m) - \tilde{G}(m)}{G(m)\sigma(m)}, \tag{33}$$

where $\sigma(m)$ are the relative errorbars.

However, there are plenty of methods, using ”guiding function” approach [32, 35–37], which can provide an almost uniform, m-independent error-bars of the QMC data. These methods are usually used when $G(m)$ changes several orders of magnitude in the range $1 \leq m \leq M$. If the m-indepenent factor $\mathcal{S}(m) \equiv \mathcal{S}$ is put outside of the sum (29), the contribution of the data points with small $\mid G(m) \mid$ to the deviation measure $D[\widetilde{A}]$ is evidently underestimated. In this case a reasonable choice for $\mathcal{S}(m)$ is to take $\mathcal{S}(m) = \mid G(m) \mid^d$, where $0 \leq d \leq 1$. Then, the contributions from the points with small and large values of $\mid G(m) \mid$ are equally represented in the sum when $d \to 1$.

3.2 Parametrization of particular spectra

We parameterize the spectral function $\widetilde{A}$ as a sum

$$\widetilde{A}(\omega) = \sum_{t=1}^{K} \eta_{\{P_t\}}(\omega) \tag{34}$$

of rectangles $\{P_t\} = \{h_t, w_t, c_t\}$

$$\eta_{\{P_t\}}(\omega) = \begin{cases} h_t \quad , & \omega \in [c_t - w_t/2, c_t + w_t/2] \; , \\ 0 \quad , & \text{otherwise} \; , \end{cases} \tag{35}$$

determined by height $h_t > 0$, width $w_t > 0$, and center c_t.

A configuration

$$\mathcal{C} = \{\{P_t\}, \, t = 1, ..., K\} \tag{36}$$

with the normalization constraint

$$\sum_{t=1}^{K} h_t w_t = I, \tag{37}$$

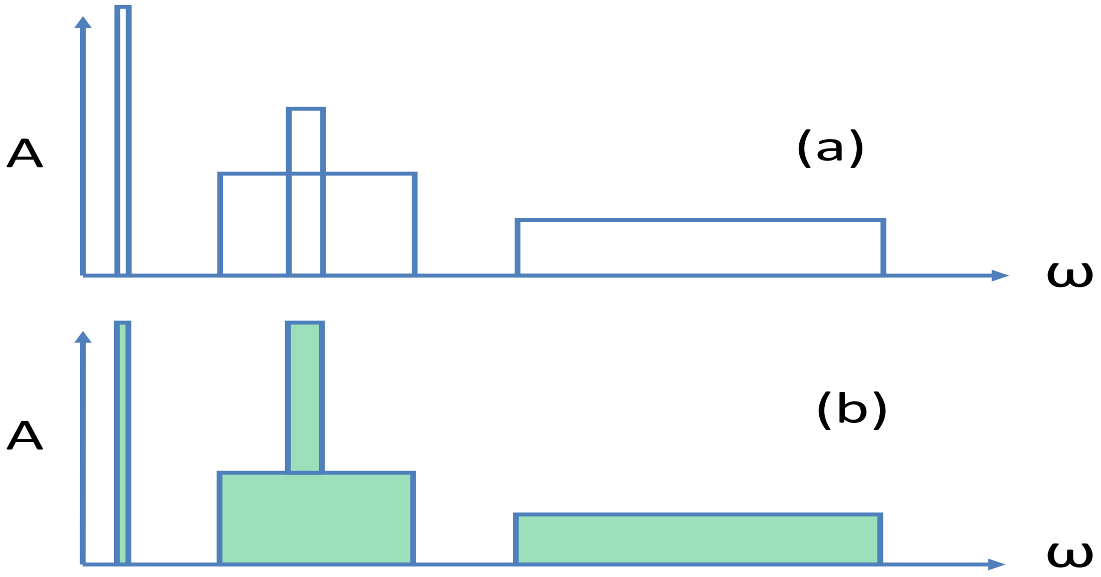

Fig. 2: *An example of configuration with $K = 4$. Panel (b) shows how the intersection of rectangles in panel (a) is treated.*

defines, according to Eqs. (31,34,35), the function $\tilde{G}(m)$ at any point m. Figure 2 shows how the intersection of rectangles is understood in SOM.

Note that the specific type of the functions (35) is not crucial for the general features of the method although simple form of analytic expressions (31,34,35) is of considerable importance for fast performance of the method. If analytic expression for $\tilde{G}(m)$ is not available for a given kernel $\mathcal{K}$, one tabulates the quantities

$$\Lambda(m,\Omega) = \int_{-\infty}^{\Omega} \mathcal{K}(m,x)\, dx \;, m = 1,\dots,M \tag{38}$$

and finds the value of $\tilde{G}(m)$ using the following straightforward relation

$$\tilde{G}(m) = \sum_{t=1}^{K} h_t \left[\Lambda(m, c_t + w_t/2) - \Lambda(m, c_t - w_t/2)\right] \;. \tag{39}$$

However, one can find analytic expressions in some cases. For example, let us consider the case when $G(m) = G(\tau_m)$ is a GF of fermion given by QMC at imaginary time (3) at zero temperature. In this case Eq. (8) reduces to $\mathcal{K}(\tau_\omega) = e^{-\tau_m \omega}$ and the spectral function is defined only at $\omega > 0$. It implies that in configuration (36) all $c_t - w_t/2 \geq 0$. Then, the explicit relation for the GF $\tilde{G}(\tau_m)$ in terms of the configuration $\mathcal{C}$ is

$$\tilde{G}_{\mathcal{C}}(\tau_m) = \begin{cases} I & , \quad \tau_m = 0 \;, \\ 2\tau_m^{-1} \sum\limits_{t=1}^{K} h_t e^{-c_t \tau_m} \sinh(w_t \tau_m/2) & , \quad \tau_m \neq 0 \;. \end{cases} \tag{40}$$

Another example of analytic expression is when one has Matsubara GF (5) with kernel (7). In this case the analytic expression for $\tilde{\mathcal{G}}(i\omega_m)$ is

$$\tilde{\mathcal{G}}_{\mathcal{C}}(i\omega_m) = \pm \sum_{t=1}^{K} h_t \ln \left[\frac{c_t - w_t/2 - i\omega_m}{c_t + w_t/2 - i\omega_m} \right] \ , \tag{41}$$

where plus (minus) sign is for boson (fermion) operators.

3.3 General overview: obtaining particular solution and its sum

Here we generally describe the whole procedure while the following sections add necessary details.

First, $L \geq 10$ attempts to find particular solutions are performed. An attempt to obtain each particular solution $\widetilde{A}_j$ consists of two stages. The first one is a random generation of initial configuration of rectangles and the second one is a fixed number F of *global updates* decreasing the deviation measure D.

At first stage of the attempt j some initial configuration $\mathcal{C}_j^{\text{init}}$ (36) is randomly generated. Random generation of configuration means that a number of rectangles K is randomly chosen, with K in some range $1 < K < K_{\text{max}}$. Parameters $\{P_t\}$ of all rectangles are randomly generated under the constraint of normalization (37). Indeed one can impose further constraints if some additional information is available.

Then, F global updates are performed. The *global update* consists of a randomly chosen sequence of *elementary updates* which are described in next sections. A global update, which modifies the configuration $\mathcal{C}_j(r) \to \mathcal{C}_j(r+1)$, is accepted when $D[\widetilde{A}_{r+1}] < D[\widetilde{A}_r]$. For the particular solution $\widetilde{A}_j$, obtained at each attempt after F global updates, one can perform control of the quality of fit of the input data using the deviation function $\Delta(m)$ (30). Number F is considered to be satisfactory if the input data $G(m)$ are fit down to the noise level in a more than half of attempts L. If not, the number of global updates F is increased and the procedure with $L \geq 10$ attempts is repeated.

Finally, when satisfactory number of global updates F is found, an accumulation of $L \gg 10$ particular solutions $\widetilde{A}_j$ and its deviation measures $D[\widetilde{A}_j]$ is performed. After L attempts there is a minimal deviation measure MIN$\{D[\widetilde{A}_j]\}$, limited by the noise of the input data $G(m)$, and the rest of measures are larger. The tests show that to avoid over-fitting, i.e. to regularize the final solution, one has to include into the sum (28) all particular solutions whose deviation measures $D[\widetilde{A}_j]$ are smaller than the double of the minimal deviation measure

$$D[\widetilde{A}_j] \leq 2\text{MIN}\{D[\widetilde{A}_j]\} \ . \tag{42}$$

Such choice of the regularization parameter is very similar to the strategy adopted in many other methods, e.g. SSM [29] or MEM [31]. Both in SOM and in many another methods the strategy is to keep differences between the fit $\tilde{G}(m)$ and data $G(m)$ of the order of error-bars to avoid over-fitting. The inequality (42) is the way to introduce the regularization parameter in the most explicit manner.

3.4 General features of elementary updates

By elementary update we mean a random change of the configuration, which is either accepted or rejected in accordance with certain rules. There are two classes of elementary updates. The updates of the class I do not alter the number of rectangles, K, changing only the values of the parameters from a randomly chosen set $\{P_t\}$. The updates of the class II either add a new rectangle with randomly chosen parameters $\{h_{K+1}, w_{K+1}, c_{K+1}\}$, or remove stochastically chosen rectangle t from the configuration. If a proposed change violates constraint (37) (e.g., a change of w_t or h_t, or any update of the class II), then the necessary change of some other parameter set $\{P_{t'}\}$ is simultaneously proposed, to satisfy the requirement of the constraint.

The updates should keep parameters of a new configuration within domain of definition of the configuration $\mathcal{C}$. Formally, the domains of definition of the configuration (36) are $\Xi_{h_t} = [0, \infty]$, $\Xi_{c_t} = [-\infty, \infty]$, $\Xi_{w_t} = [0, \infty]$, and $\Xi_K \in [1, \infty]$. However, for the sake of faster convergence, one can reduce the domains of definition.

As there is no general *a priori* prescription for choosing reduced domains of definition, the rule of thumb is to start with maximal domains and then, after some rough solution is found, reduce the domains to reasonable values suggested by this solution. In particular, since the probability to propose a change of any parameter of configuration is proportional to K^{-1}, it is natural to restrict maximal number of rectangles $\Xi_K \in [1, K_{\text{max}}]$ by some large number K_{max}. To forbid rectangles with extremely small weight, which contribution to $\tilde{G}(\tau)$ is less than statistic errors of $G(\tau)$, one can impose the constraint $h_t w_t \in [S_{\text{min}}, 1]$, with $S_{\text{min}} \ll I K_{\text{max}}^{-1}$. When there is some preliminary knowledge that overwhelming majority of integral weight of the spectral function $\widetilde{A}(\omega)$ is in a range $[\omega_{\text{min}}, \omega_{\text{max}}]$, one can restrict the domain of definition of the parameter c_t by $\Xi_{c_t} = [\omega_{\text{min}}, \omega_{\text{max}}]$. Then, to reduce the phase space one can choose $\Xi_{h_t} = [h_{\text{min}}, \infty]$ and $\Xi_{w_t} = [w_{\text{min}}, \min\{2(c_t - \omega_{\text{min}}), 2(\omega_{\text{max}} - c_t)\}]$.

While the initial configuration, the update type, and the parameter to be altered are chosen stochastically, the variation of the values of the parameters relevant to the update is optimized to maximize the decrease of D. Each elementary update of our optimization procedure (even that of the class II) is organized as a proposal to change some continuous parameter ξ by randomly generated $\delta\xi$ in a way that the new value belongs to Ξ_ξ. Although proposals with smaller values of $\delta\xi$ are accepted with higher probability it is important, for the sake of better convergence, to propose sometimes changes $\delta\xi$ that probe the whole domain of definition Ξ_ξ. To probe all scales of $\delta\xi \in [\delta\xi_{\text{min}}, \delta\xi_{\text{max}}]$ we generate $\delta\xi$ with the probability density function $\mathcal{P} \sim (\max(|\delta\xi_{\text{min}}|, |\delta\xi_{\text{max}}|)/|\delta\xi|)^\gamma$, where $\gamma \gg 1$.

Calculating the deviation measures $D(\xi)$, $D(\xi + \delta\xi)$, $D(\xi + \delta\xi/2)$, and searching for the minimum of the parabolic interpolation, we find an optimal value of the parameter change

$$\delta\xi_{\text{opt}} = -b/2a, \tag{43}$$

where

$$a = 2(D(\xi + \delta\xi) - 2D(\xi + \delta\xi/2) + D(\xi))(\delta\xi)^{-2}, \tag{44}$$

and

$$b = (4D(\xi + \delta\xi/2) - D(\xi + \delta\xi) - 3D(\xi))\delta\xi. \tag{45}$$

In the case $a > 0$ and $\xi_{\text{opt}} \in \Xi_\xi$ we adopt as the update proposal $\tilde{\delta\xi}$ one of the values $\delta\xi$, $\delta\xi/2$, or $\delta\xi_{\text{opt}}$ for which the deviation measure $D(\xi + \tilde{\delta\xi})$ is the smallest. Otherwise, if the parabola minimum is outside Ξ_ξ, one has to compare only deviations for $\delta\xi$ and $\delta\xi/2$.

3.5 Global updates

The updating strategy has to provide efficient minimization of the deviation measure. It is highly inefficient to accept only those proposals that lead to the decrease of deviation, since there is an enormous number of local minima of deviation measure with values $D_{\text{loc}}[\mathcal{C}]$ much larger than that obtained as minimal deviation measure MIN$\{D[\widetilde{A}_j]\}$. As we observed it in practice, these multiple minima drastically slow down (or even freeze) the process.

To optimize escape from a local minimum, one has to provide a possibility of reaching a new local minimum with lower deviation measure through a sequence of less optimal configurations. It might seem that the most natural way of doing this would be to accept sometimes (with low enough probability) the updates leading to the increase of the deviation measure. However, this simple strategy turns out to be impractical. The reason is that the density of configurations per interval of deviation sharply increases with D. So that the acceptance probability for a deviation-increasing update should be fine-tuned to the value of D. Otherwise, the optimization process will be either non-convergent, or ineffective [if the acceptance probability is, correspondingly, either too large, or too small in some region of D].

A way out of the situation is to perform some sequence of T *temporary* elementary updates of a configuration $\mathcal{C}(0)$

$$\mathcal{C}(0) \to \mathcal{C}(1) \to ... \to \mathcal{C}(r) \to \mathcal{C}(r+1) \to ... \to \mathcal{C}(T) \ , \tag{46}$$

where the proposal to update the configuration $\mathcal{C}(r) \to \mathcal{C}(r+1)$ is (temporary) accepted with the probability

$$\mathcal{P}_{r \to r+1} = \left\{ \begin{array}{lll} 1 & , & D[\mathcal{C}(r+1)] < D[\mathcal{C}(r)] \ , \\ Z\left(D[\mathcal{C}(r)]/D[\mathcal{C}(r+1)]\right) & , & D[\mathcal{C}(r+1)] > D[\mathcal{C}(r)] \ . \end{array} \right. \tag{47}$$

(Function Z satisfies boundary conditions $Z(0) = 0$ and $Z(1) = 1$.) Then we choose out of the configurations $\{\mathcal{C}(r)\}$ (46) the one with minimal deviation measure and, if it is different from $\mathcal{C}(0)$, declare it to be the result of the global update, or, if this configuration turns out to be just $\mathcal{C}(0)$, reject the update.

We choose the function Z in the form

$$Z(x) = x^{1+d} \qquad (d > 0) \ , \tag{48}$$

which leads to comparatively high probabilities to accept small increases of deviation measures and hampers significant enlargements of deviation measure. Empirically, we found out that the

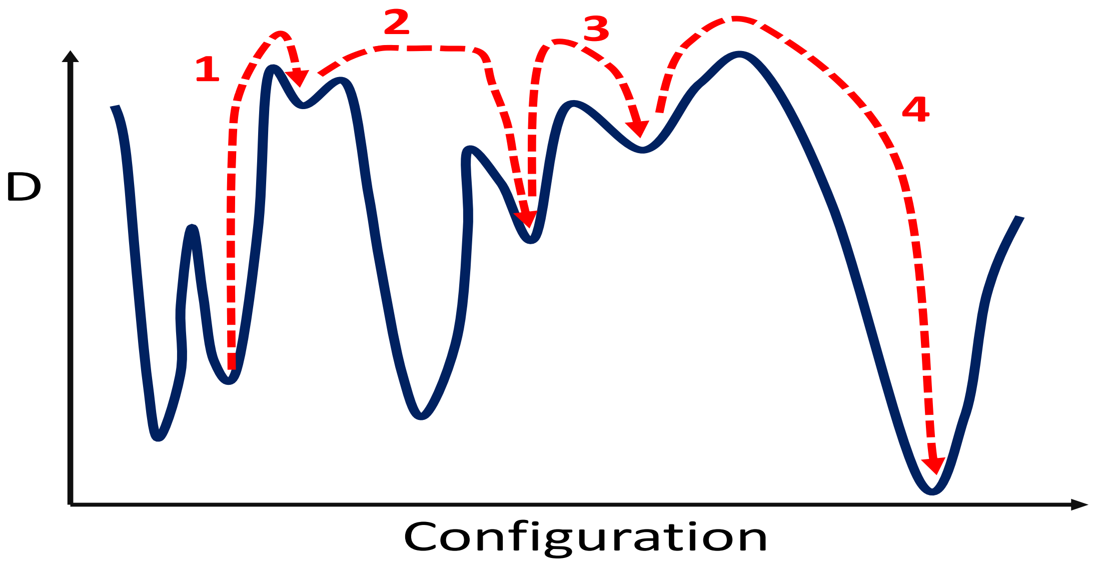

Fig. 3: *Example of a* global update *with 4 elementary updates. The process transfers initial configuration by 4 (dashed red arrows)* elementary updates *from the initial minimum of D-relief (solid blue line) to the lower minimum through the minima whose deviation measures are larger than that in the initial configuration.*

global update procedure is most effective if one keeps parameter $d = d_1 \ll 1$ at the first T_1 steps of sequence (46) (to leave local minimum) and then changes this parameter to a value $d = d_2 \gg 1$ for the last $T - T_1$ elementary updates (to decrease the deviation measure). In our algorithm the values $T \in [1, T_{\text{max}}]$, $T_1 \in [1, T]$, $d_1 \in [0, 1]$, and $d_2 \in [1, d_{\text{max}}]$ were stochastically chosen for each global update run.

The two-step procedure of the global update is a method to reach the same goal as the tempering and annealing procedures in SSMs methods [27,25,29]. A temporary rise and consequent drop of the fictitious temperature $\mathcal{T}$ is used in the standard SSMs. Similarly, temporary permission to grow up the deviation measure with the following directive to drop it down is introduced into SOM. An exchange between deep local minima of the deviation measures $D_{\text{loc}}[\mathcal{C}]$ has low probability. Therefore, the procedure which first rises and then drops the deviation measure arranges a path between some deep local minima through some shallow ones.

3.6 Final solution and refinement

After a set of L configurations

$$\{C_j^{\text{fin}}, j = 1, ..., L\} \tag{49}$$

that satisfy the criterion (42) is produced, the solution (28) is obtained by summing up the rectangles (35, 49).

We, however, employ a more elaborated procedure, which we call refinement. Namely, we use the set (49) as a source of L_{ref} new independent starting configurations for further optimization. These starting configurations are generated as a linear combinations of randomly chosen

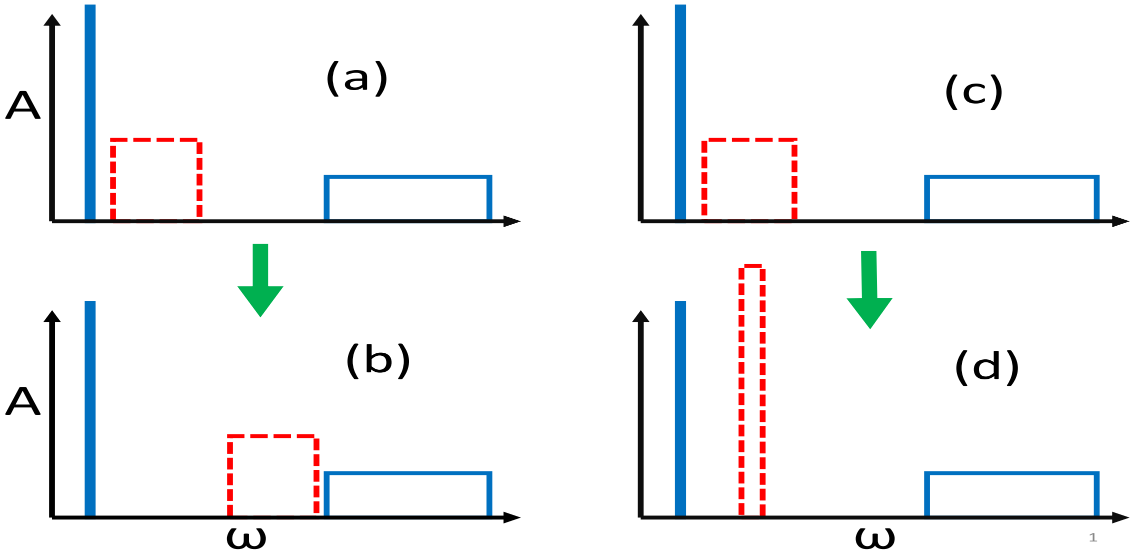

Fig. 4: *Some of elementary updates of the class I: (a)→(b) is shift of rectangle (dashed red line); (c)→(d) is change of the height of rectangle (dashed red line) without changing its weight* $h_t w_t$ *and center* c_t*.*

members of the set (49) with stochastic weight coefficients. Then, the refined final solution is represented as the average (28) of L_{ref} particular solutions resulting from the optimization procedure.

The main advantage of such a trick is that initial configurations for optimization procedure now satisfy the criterion (42) from the very beginning. Moreover, as any linear combination of sufficiently large number R of randomly chosen parent configurations $\{C_\eta^{\text{fin}}, \eta = 1, ..., R\}$ smooths the sawtooth noise, the deviation of a summary configuration $C_{\text{ref}}^{\text{fin}}$ is normally lower than that of each additive one.

3.7 Elementary updates of class I

(A) Shift of rectangle. Change the center c_t of a randomly chosen rectangle t (Fig. 4a and 4b). The continuous parameter for optimization (43-45) is $\xi = c_t$ which is restricted by domain of definition $\Xi_{c_t} = [\omega_{\text{min}} + w_t/2, \omega_{\text{max}} - w_t/2]$.

(B) Change of width without change of weight. Alter the width w_t of a randomly chosen rectangle t without change of the rectangle weight $h_t w_t = const$ and center c_t (Fig. 4c and 4d). The continuous parameter for optimization is $\xi = w_t$ which is restricted by $\Xi_{w_t} = [w_{\text{min}}, \min\{2(c_t - \omega_{\text{min}}), 2(\omega_{\text{max}} - c_t)\}]$.

(C) Change of weight of two rectangles. Change the heights of two rectangles t and t' (where t is a randomly chosen and t' is either randomly chosen or closest to t rectangle) without change of widths of both rectangles. Continuous parameter for optimization is the variation of the rectangle t height $\xi = h_t$. To restrict the weights of chosen rectangles to $[S_{\text{min}}, 1]$ and preserve the total normalization (37) this update suggests to change $h_t \rightarrow h_t + \delta\xi$ and $h_{t'} \rightarrow h_{t'} -$

$\delta\xi w_{t'}/w_t$ with $\delta\xi$ confined to the interval

$$S_{\text{min}}/w_t - h_t < \delta\xi < (h_{t'} - S_{\text{min}}/w_{t'})w_t/w_{t'} \ . \tag{50}$$

3.8 Elementary updates of class II

(D) Adding a new rectangle. To add a new rectangle one has to generate some new set $\{P_{\text{new}}\} = \{h_{\text{new}}, w_{\text{new}}, c_{\text{new}}\}$ and reduce the weight of some other rectangle t (either randomly chosen or closest) in order to keep the normalization condition (37). The reduction of the rectangle weight t is obtained by decreasing its height h_t.
The center of the new rectangle is selected at random according to

$$c_{\text{new}} = (\omega_{\text{min}} + w_{\text{min}}/2) + (\omega_{\text{max}} - \omega_{\text{min}} - w_{\text{min}})r \ . \tag{51}$$

As soon as the value c_{new} is generated, the maximal possible width of a new rectangle is given by

$$w_{\text{new}}^{\text{max}} = 2\min(\omega_{\text{max}} - c_{\text{new}}, c_{\text{new}} - \omega_{\text{min}}) \ . \tag{52}$$

Continuous parameter for optimization $\delta\xi = h_{\text{new}}w_{\text{new}}$ is generated to keep weights of both new rectangle and rectangle t larger than S_{min}

$$\delta\xi = S_{\text{min}} + r(h_t w_t - S_{\text{min}}) \ . \tag{53}$$

Then, the value of the new rectangle height h_{new} for given $\delta\xi$ is generated to keep the width of new rectangles within the limits $[w_{\text{min}}, w_{\text{new}}^{\text{max}}]$

$$h_{\text{new}} = \delta\xi/w_{\text{new}}^{\text{max}} + r(\delta\xi/w_{\text{min}} - \delta\xi/w_{\text{new}}^{\text{max}}) \ . \tag{54}$$

(E) Removing a rectangle. To remove some randomly chosen rectangle t, we enlarge the height $h_{t'}$ of some another (either randomly chosen or closest) rectangle t' according to condition (37). Since such procedure does not involve continuous parameter for optimization, we unite removing of rectangle t with the shift procedure *(A)* of the rectangle t'. Then, the proposal is the configuration with the smallest deviation measure.
(F) Splitting a rectangle. This update cuts some rectangle t into two rectangles with the same heights h_t and widths $w_{\text{new}_1} = w_{\text{min}} + r(w_t - w_{\text{min}})$ and $w_{\text{new}_2} = w_t - w_{\text{new}_1}$ (Fig. 5). Since removing a rectangle t and adding of two new glued rectangles does not change the spectral function we introduce the continuous parameter $\delta\xi$ which describes the shift of the center of a new rectangle with the smallest weight. Second rectangle is shifted into opposite direction to keep the center of gravity of two rectangles unaltered. The domain of definition Ξ_ξ obviously follows from the parameters of the new rectangles.
(G) Gluing rectangles. This update glues two (either randomly chosen or closest) rectangles t and t' into single new rectangle with the weight $h_{\text{new}}w_{\text{new}} = h_t w_t + w_{t'}h_{t'}$ and width $w_{\text{new}} = (w_t + w_{t'})/2$. The initial center of the new rectangle c_{new} corresponds to the center of gravity of rectangles t and t'. We introduce a continuous parameter by simultaneously shifting the new rectangle.

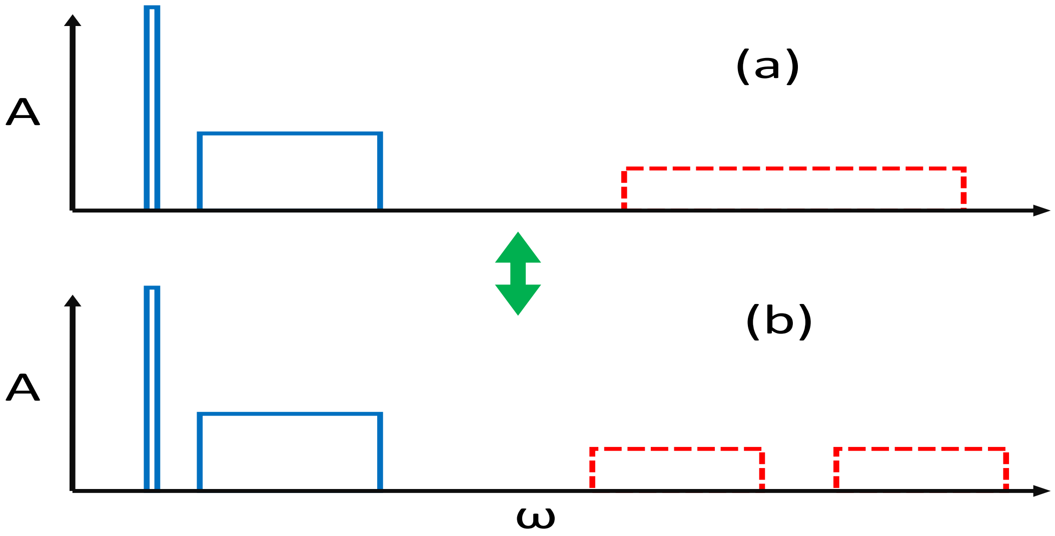

Fig. 5: *Elementary updates of the class II: (a)→(b) splitting of rectangle (dashed red line) and (b)→(a) gluing of rectangles (dashed red line) without changing its total weight $h_t w_t$ and center of gravity c_t.*

4 Practical aspects of the method

First, we summarize what is done by SOM algorithm automatically and what are the numbers we need to determine in each particular case. The algorithm described in the previous section is able to search for as many L particular solutions $\widetilde{A}_j$ as requested. For every attempt j to find a particular solution $\widetilde{A}$ it does the following steps:

- [i] Generates an initial configuration $\mathcal{C}_j^{\text{init}}$ with $K < K_{\text{max}}$ rectangles. Each initial configuration $\mathcal{C}_j^{\text{init}}$ is statistically independent from the previous one $\mathcal{C}_{j-1}^{\text{init}}$.
- [ii] Searches for a particular solution $\widetilde{A}_j$ performing F global updates.
- [iii] Stores the final configuration C_j^{fin} of the particular solution $\widetilde{A}_j$ and its deviation measure $D[\widetilde{A}_j]$.

Then, after L attempts one obtains the final regularized solution as the sum

$$A(\omega) = \frac{1}{L_{\text{good}}} \sum_{j=1}^{L} \theta \left\{ 2\text{MIN}\{D[\widetilde{A}_j]\} - D[\widetilde{A}_j] \right\} \widetilde{A}_j(\omega) \; . \tag{55}$$

Here $\theta(x)$ is the θ-function: $\theta(x \geq 0)$ equals to unity and zero otherwise. L_{good} is a number of ”good” fits

$$L_{\text{good}} = \sum_{j=1}^{L} \theta \left\{ 2\text{MIN}\{D[\widetilde{A}_j]\} - D[\widetilde{A}_j] \right\} , \tag{56}$$

restricted by the regularization condition that the deviation measure is less than twice of the minimal deviation measure MIN$\{D[\widetilde{A}_j]\}$ found during L attempts.

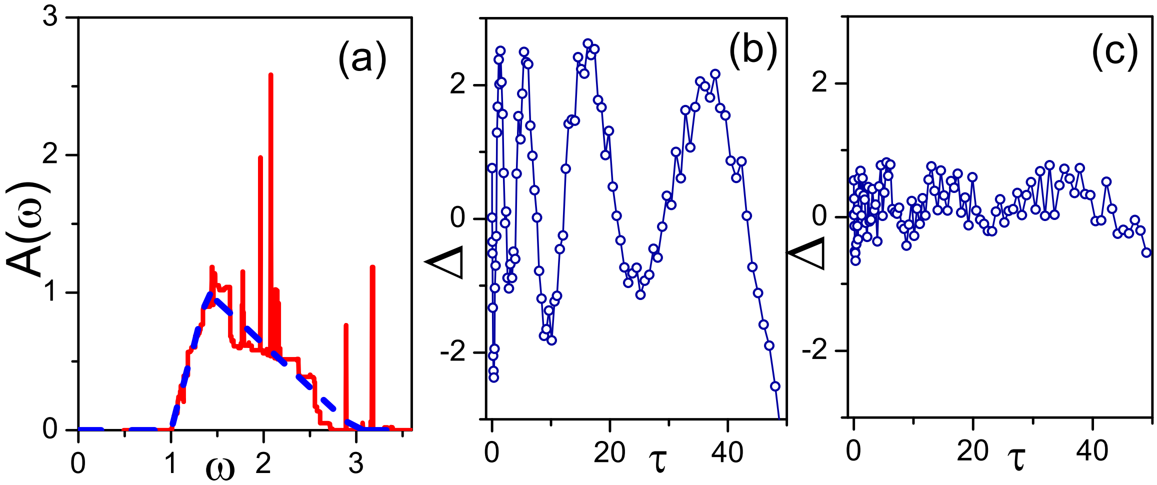

Fig. 6: *(a) Typical spectrum* $\widetilde{A}_j(\omega)$ *(red solid line), corresponding to particular configuration* $\mathcal{C}_j$*, compared to actual spectrum (blue dashed line). Typical dependence of the deviation function* $\varDelta(m)$ *(30) on imaginary times* τ_m *corresponding to a spectrum* $\widetilde{A}_j(\omega)$ *which (a) under-fits and (b) over-fits the uncorrelated noise of imaginary time data.*

To finalize the preparation of the method for solving different problems it is necessary to give recipes how to choose the numbers F (Sec. 4.1) and L (Sec. 4.2) in every particular case.

4.1 Choosing the number of global updates F

To check whether particular number F is large enough to reproduce the given data set (1) it is enough to perform about $L \approx 10$ attempts to find particular solutions $\widetilde{A}_j$ and consider the deviation functions $\varDelta(m)$ (30) which correspond to each particular solution.

Note, a particular solution itself does not bear any important information on the quality of the fit of the data. Indeed, every particular solution contains a sawtooth noise and typically looks like solid red line in Fig. 6(a). One can claim that comparison with exact answer (dashed blue line) can give some key to tell about quality, but, in practice, the exact answer is not known.

To the contrary, the deviation function $\varDelta(m)$ gives direct answer on the question on the quality of the fit. Such test of the quality of the fit requires uncorrelated noise in the QMC data, i.e. when deviation from exact solution $\delta G(m)$ for any m-point is independent from that in neighboring m-point. Indeed, we take uncorrelated nature of noise for granted because analytic continuation from correlated QMC data is a way to a wrong answer from the onset.

For the sake of definiteness we consider here an example with imaginary time data set. However, the case of the Matsubara representation is identical. Figure 6(b) shows an example when the input data are under-fitted by a particular solution. One can see that the typical period of oscillations of the deviation function $\varDelta(m)$ around zero is much larger than the typical distance between the input data points τ_m. To the contrary, the fit shown in the Fig. 6(c) is noise-limited because the typical period of oscillations is comparable with a typical distance between data points. One can introduce a numeric criterion of the fit-quality

$$\kappa = \frac{1}{M-1}\sum_{m=2}^{M}\theta\left\{-\mathcal{D}(m)\mathcal{D}(m-1)\right\} \tag{57}$$

equals to the ratio of number of intersections of zero by the function $\mathcal{D}(m)$ and number of the intervals between M input data points. Ideally, one has to pursue $\kappa \to 1$ though it happens very rare that $\kappa > 1/3$. Practically, a solution with fit-quality $\kappa > 1/4$ can be considered as a good one.

Then, after $L \approx 10$ attempts to find particular solutions $\widetilde{A}_j$, each by F global updates, the fit-qualities κ_j are considered. If $\kappa_j > 1/4$ is for more than $L/2$ attempts, the number F is large enough. If not, it has to be increased.

4.2 Choosing the number of particular solutions L

SOM performs L attempts to find particular solutions $\tilde{A}_j$. The sum of particular solutions (28) becomes smoother as the number L of attempts increases. Figure 7(a) shows how the *sawtooth noise* is self-averaged when L increases. One can collect a distribution of the deviation measures and the typical picture is presented in Fig. 7(b). There is the minimal deviation measure D_{min} which corresponds to the best fit of the noisy data $G(m)$ and there is a probability distribution which increases when D is slightly larger than the minimal deviation measure D_{min}. Indeed, the distribution has some maximum and decreases for larger deviation measures D because there is the most probable deviation measure which is reached after F global updates.

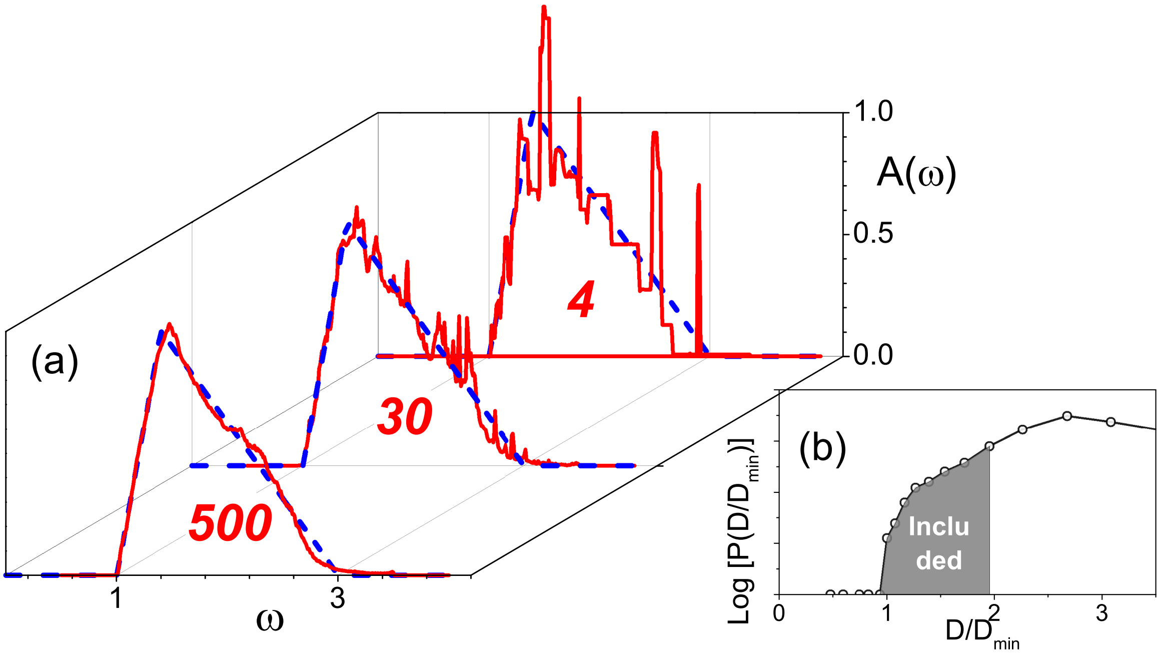

Fig. 7: *(a) Self-averaging of the sawtooth noise after summation of 4, 30, and 500 particular solutions. (b) Typical probability distribution $P(D/D_{min})$ of solutions with different deviation measures D.*

The shaded area in Fig. 7(b) shows which part of the distribution is included into the final solution (55) in order to regularize the sawtooth noise. We can not formulate any rigorous criterion when one can stop accumulation of particular solutions. However, it looks reasonable to stop accumulation when there is no significant difference between final spectra with $L_{\rm good}$ and $(1-1/3)L_{\rm good}$ particular spectra included. We found that the above criterion is similar to that when one compares the sum (55) with $\theta\left\{2\text{MIN}\{D[\widetilde{A}_j]\}-D[\widetilde{A}_j]\right\}$ and with $\theta\left\{2(1-1/3)\text{MIN}\{D[\widetilde{A}_j]\}-D[\widetilde{A}_j]\right\}$.

5 Tests of SOM

The procedure to check the SOM for different cases is the following. A spectral function $A(\omega)$, which is called *test spectrum*, is selected. Then, a set of input data with superimposed noise

$$\left\{\tilde{G}(m)\left[1+\frac{\mathcal{B}}{2}\mathcal{R}\right]\;,\; m=1,M\right\} \tag{58}$$

is generated. Finally, the SOM procedure is performed to restore the test spectrum.
Generation procedure of $\tilde{G}(m)$ uses particular kernel $\mathcal{K}$, relation (31), and test spectrum. A statistical noise is added with amplitude $\mathcal{B}$ using a random number $\mathcal{R}$ in the range

$$\mathcal{R}\in[-1,1]\;. \tag{59}$$

We present tests for imaginary time representation in Sec. 5.1. In particular, we test the case of zero temperature GF in Sec. 5.1.1, finite temperature GF for fermions in Sec. 5.1.2, and finite temperature optical conductivity in Sec. 5.1.3. The test for GF in the Matsubara representation is presented in Sec. 5.2.

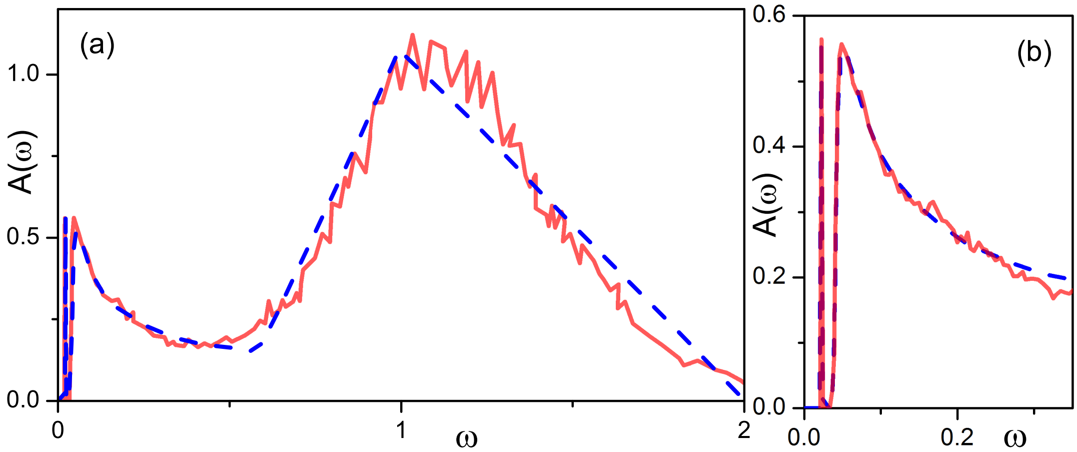

Fig. 8: *The test spectrum (dashed blue line) and the spectrum obtained by SOM (solid red line). Panels (a) and (b) show the whole spectrum and its low energy part, respectively.*

5.1 Test of SOM for imaginary time representation

5.1.1 Zero temperature Green function for a quasiparticle

For zero temperature GF the kernel (8) for fermions reduces to exponent $\mathcal{K}(\tau_\omega) = e^{-\tau_m \omega}$ and the spectral function $A(\omega)$ is defined only at $\omega > 0$. To check the accuracy of SOM, we tested it for the spectral density distribution that spreads over large range of frequencies and simultaneously possesses fine structure in low-frequency region [32].

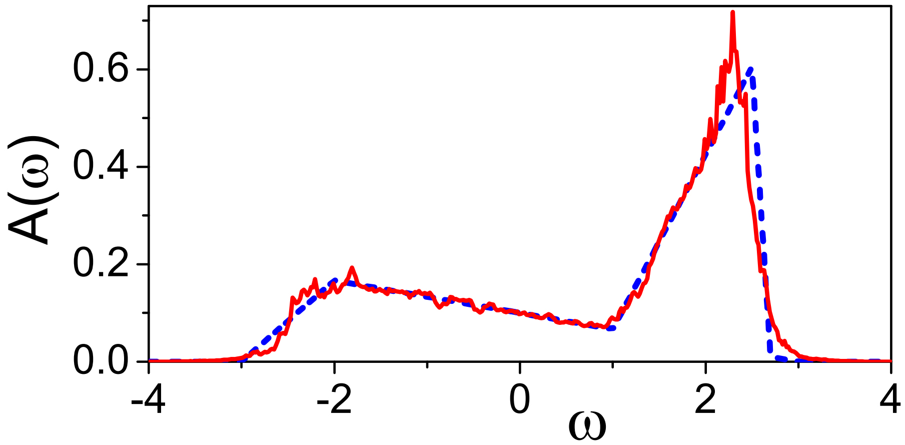

Fig. 9: *The test spectrum (dashed blue line) and the spectrum obtained by SOM (solid red line). for Lehmann spectral function of fermions at finite temperature.*

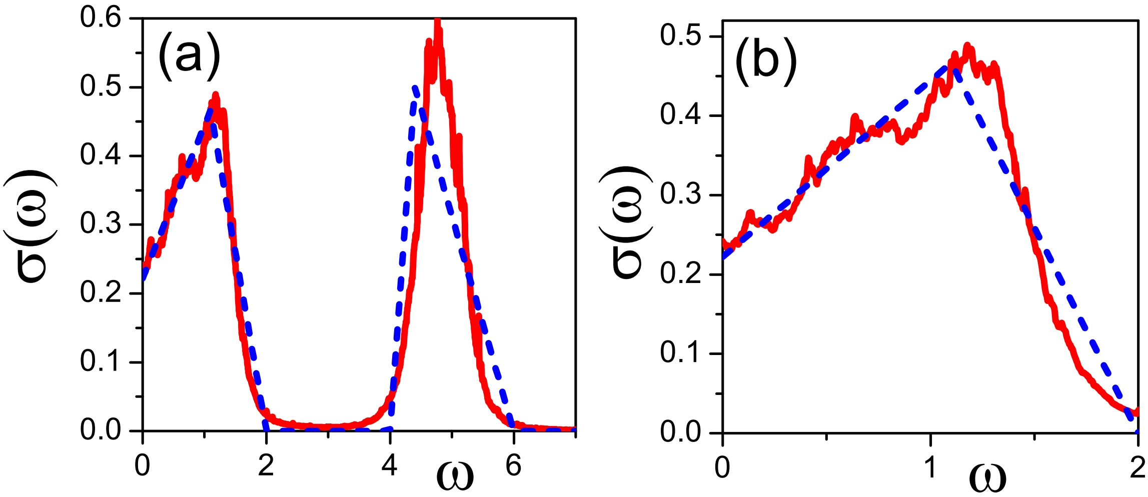

Fig. 10: *The test spectrum (dashed blue line) and the spectrum obtained by SOM (solid red line). for optical conductivity at finite temperature. Panels (a) and (b) show the whole range and low energy part, respectively.*

The test spectrum was modeled as the sum of the delta-function with the energy $\varepsilon_\delta = 0.03$ and the weight $Z_\delta = 0.07$, and continuous high-frequency spectral density which starts at the

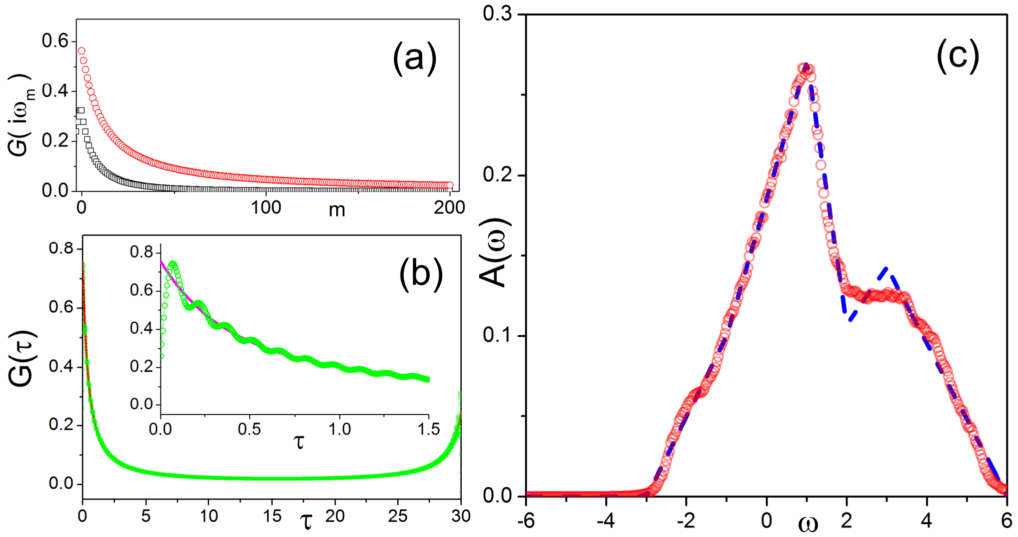

Fig. 11: *(a) First 200 Fourier components of real (red circles) and imaginary (black squares) parts of GF in the Matsubara representation obtained from GF on imaginary time in panel (b). (b) Imaginary time GF (solid line) and imaginary time GF obtained from first $M = 200$ GFs in Matsubara representation. Inset shows data at low imaginary times. (c) Actual spectrum (dashed blue line) and that restored from 200 Matsubara components (red solid line).*

threshold $\varepsilon_{\text{th}} = 0.04$. The continuous part of the spectral function A_{con} was modeled by the function

$$A_{\text{con}}(\omega) = \frac{Z_\delta \sqrt{\omega - \varepsilon_{\text{th}}}}{2\pi\sqrt{\varepsilon_{\text{gap}}}[(\omega - \varepsilon_{\text{th}}) + \varepsilon_{\text{gap}}]} \theta\left(\omega - \varepsilon_{\text{th}}\right) \theta\left(0.566 - \omega\right) \tag{60}$$

(here $\varepsilon_{\text{gap}} = \varepsilon_{\text{th}} - \varepsilon_\delta$ is a microgap) in the range $\omega \in [\varepsilon_{\text{th}}, 0.566]$ and by a triangle at higher frequencies (see the blue dashed line in the Fig. 8).

The GF $G(\tau)$ was calculated from the test spectrum in the $M = 300$ points $\tau_m = \tau_{\text{max}} m^2/M^2$ in the time range from zero to $\tau_{\text{max}} = 1000$. The noise amplitude was chosen rather small $\mathcal{B} = 10^{-4}$. The restored spectral function reproduces both gross features of high-frequency part (Fig. 8(a)) and the fine structure at small frequencies (Fig. 8(b)). The energy and the weight of the delta-function was restored with the accuracy 10^{-4}. The final solution was obtained by the averaging (55) of $L_{\text{good}} = 1100$ particular solutions.

5.1.2 Finite temperature Green function for fermions

In this test the kernel is (8) for fermions and the spectral function $A(\omega)$ is defined in the whole range $\infty < \omega < \infty$. The test spectrum was modeled by two triangles (blue dashed line in Fig. 9).

The GF $G(\tau)$ was calculated at finite temperature $\beta = 50$ in the uniformly separated $M = 600$ points in the range $[0, \beta]$. The noise amplitude was chosen rather small $\mathcal{B} = 10^{-4}$. The restored spectral function (red solid line in Fig. 9) reproduces the main features of the test spectrum. The final solution was obtained by the averaging (55) of $L_{\text{good}} = 150$ particular solutions.

5.1.3 Finite temperature optical conductivity

In this test the kernel is (9) and the spectral function $\sigma(\omega)$ is symmetric $\sigma(\omega) = \sigma(-\omega)$. The test spectrum was modeled by two triangles (blue dashed line in Fig. 10).

The current-current correlation function $J(\tau)$ was calculated at finite temperature $\beta = 20$ in the uniformly separated $M = 200$ points in the range $[0, \beta]$. The noise amplitude was chosen rather small $\mathcal{B} = 10^{-4}$. The restored optical conductivity (red solid line in Fig. 10) reproduces the main features of the test spectrum (Fig. 10(a)) and its low energy part (Fig. 10(b)). The final solution was obtained by the averaging (55) of $L_{\text{good}} = 200$ particular solutions.

5.2 Test of SOM for Matsubara representation

In this test the kernel is (7) and the spectral function $A(\omega)$ is defined in the whole range $\infty < \omega < \infty$. The test spectrum was modeled by two triangles (blue dashed line in Fig. 11(c)).

The GF $\mathcal{G}(i\omega_n)$ was calculated at finite temperature $\beta = 30$ for the first $M = 200$ positive Matsubara frequencies $i\omega_n$ and the analytic continuation was done directly from the set of GFs in Matsubara representation. The noise amplitude was $\mathcal{B} = 10^{-4}$. In Fig. 11 one can see rather good overall agreement between the test and restored spectra. The final solution was obtained as average (55) of $L_{\text{good}} = 200$ particular solutions.

The real and imaginary parts of GF in the Matsubara representation are shown in Fig. 11(a). Imaginary time GF, calculated from the test spectrum by Eq. (31), is shown in Fig. 11(b) by solid black line. Imaginary time GF, calculated from the first 200 Matsubara components in the inverse Fourier transform (6), is shown in Fig. 11(b) by green circles. It is seen that the first 200 components of the GF in the Matsubara representation are not enough to describe the imaginary time GF at small values of τ. This discrepancy is a direct indication that the transformation of the QMC data from one representation into another one is a step which can miss information. Namely, it is dangerous to transform the Matsubara representation into the imaginary time one because even large number of Matsubara points still can lead to spurious oscillations of the imaginary time GF (Fig. 11(b)). Hence, it is preferable to make the analytic continuation from the same representation as that in which the QMC data are obtained.

Conclusion

We presented stochastic optimization method for analytic continuation. The method was considered in relation with numerous other methods handling ill-posed problems. It was concluded that the method is the best for the problems when one has to avoid any artificial smoothening of the spectral function and when there is no *a priory* knowledge about the expected solution.

The method was successfully applied to many problems. The exponential kernel $\mathcal{K}(m, \omega) = \exp[-\tau_m \omega]$ for zero temperature was considered in Refs. [32, 38–57] and various kernels ranging from Fermi distribution to the Matsubara frequency representation are considered in Refs. [58–62]. Also, the method was used for Gaussian kernel in Refs. [6, 7]. Indeed, broad area of the solved problems and successful tests for problems which were not considered before give confidence that the method still has considerable potential in application to problems where initial *a priory* knowledge is not available.

Author acknowledges fruitful discussions with B.V. Svistunov and N.V. Prokof'ev.

6 General description

The code is for making analytic continuation (or generally solving Fredhold integral equation of the first kind) for 6 types of kernels. The code is made in blocks which allows trivial insertion of any new kernel. The code works as a combination of the stochastic optimization method and consistent constraints method [N. Prokof'ev and B. Svistunov arXiv:1304.5198]. It is **THREE** orders of magnitude faster that previous versions. It **determines the errorbars** of the obtained analytic continuation by statistical analysis of obtained solutions in selected bins.

6.1 Objective function

Here is a definition of the *objective function* to minimize. The set G corresponds to some QMC data on imaginary times $G(m) = G(\tau_m)$ or at some Matsubara frequencies $G(m) = \mathcal{G}(i\omega_m)$, $m = 1, \dots, M$. Then the *objective function* is given by expression

$$D[\widetilde{A}] = \sum_{m=1}^{M} |\varDelta(m)|^2 \; . \tag{61}$$

Here $\varDelta(m)$ is the *deviation function*

$$\varDelta(m) = \frac{G(m) - \tilde{G}(m)}{\sigma(m)}, \tag{62}$$

which characterizes individual deviations of specific data points $G(m)$ from the values of the simulated function $\widetilde{G}(m)$ defined by the particular spectral function $\widetilde{A}$ in terms of relation

$$\widetilde{G}(m) = \int_{-\infty}^{\infty} d\omega \; \mathcal{K}(m, \omega) \; \widetilde{A}(\omega) \; . \tag{63}$$

The factors $\sigma(m)$ are the error-bars of the QMC data $G(m)$ and given in the input files.

7 Available kernels and their properties

In this section we give description of different problems of analytic continuation with different kernels and describe their properties which are important for making effective algorithm.

7.1 One particle at zero temperature

Low density limit assumes $\mu \to -\infty$ and, hence, the ω-dependent factor of the kernel (8) for any temperature transforms to

$$\mathcal{K}(\tau, \omega) = \exp(-\tau\omega) \; . \tag{64}$$

For zero temperature $\beta \to \infty$ the kernel is nonzero only for positive $\omega > 0$ and the final equation to solve is

$$\boxed{G(\tau) = \int_{0}^{\infty} \exp(-\tau\omega) A(\omega) d\omega} \; . \tag{65}$$

It follows from the above equation that the normalization condition is

$$\boxed{\int_0^\infty A(\omega)d\omega = G(0)} \; . \tag{66}$$

7.2 Optical conductivity

In this case the kernel (9) relates current-current correlation function $J(\tau)$ to the optical conductivity $\sigma(\omega)$.

7.2.1 $T \neq 0$

In this case [1,25,63]

$$\boxed{J(\tau) = \int_{-\infty}^\infty \mathcal{K}_{\text{OC}}(\tau,\omega)\sigma(\omega)d\omega} \; . \tag{67}$$

Dividing the integral into two parts $\omega > 0$ and $\omega < 0$, changing $\omega \to -\omega$, and using property $\sigma(\omega) = \sigma(-\omega)$, one gets

$$J(\tau) = \int_0^\infty \widetilde{\mathcal{K}}_{\text{OC}}(\tau,\omega)\sigma(\omega)d\omega \; , \tag{68}$$

where

$$\widetilde{\mathcal{K}}_{\text{OC}}(\tau,\omega) = \mathcal{K}_{\text{OC}}(\tau,\omega) + \mathcal{K}_{\text{OC}}(\tau,-\omega) \; . \tag{69}$$

After some algebra one can get several representations for $K_{\text{OC}}(\tau,\omega)$, like

$$\widetilde{\mathcal{K}}_{\text{OC}}(\tau,\omega) = \frac{\omega}{\pi}\frac{\exp(-\tau\omega) + \exp(-(\beta-\tau)\omega)}{1-\exp(-\beta\omega)} \tag{70}$$

or

$$\widetilde{\mathcal{K}}_{\text{OC}}(\tau,\omega) = \frac{\omega}{\pi}\frac{\cosh\left[(\tau-\beta/2)\omega\right]}{\sinh[\beta\omega/2]} \; . \tag{71}$$

Easily derived property is the normalization condition

$$\boxed{\int_0^\infty \sigma(\omega)d\omega = \frac{\pi}{2}\int_0^\beta J(\tau)d\tau} \; . \tag{72}$$

7.2.2 $T = 0$

For $\beta \to \infty$ the kernel reduces to $\widetilde{\mathcal{K}} = (\omega/\pi)\exp(-\tau\omega)$ and one can assume $A(\omega) = \omega\sigma(\omega)/\pi$, use Eq. (66) to find $A(\omega)$, and find optical conductivity as

$$\boxed{\sigma(\omega) = \frac{\pi}{\omega}A(\omega)} \; . \tag{73}$$

Obvious normalization condition is

$$\boxed{\int_0^\infty A(\omega)d\omega = J(0)} \; . \tag{74}$$

7.3 Fermi statistics at finite T

In this case kernel is defined

$$\mathcal{K}_{\mathrm{F}}(\tau,\omega) = \frac{\exp(-\tau\omega)}{1+\exp(-\beta\omega)} \; , \tag{75}$$

and the integration in integral equation is from minus to plus infinity

$$\boxed{G(\tau) = \int_{-\infty}^{\infty} \mathcal{K}_{\mathrm{F}}(\tau,\omega) A(\omega) d\omega} \; . \tag{76}$$

In this case there is no additional symmetries to reduce the equation.

7.4 Bose statistics at finite T

In this case kernel is defined

$$\mathcal{K}_{\mathrm{B}}(\tau,\omega) = \frac{\exp(-\tau\omega)}{1-\exp(-\beta\omega)} \; , \tag{77}$$

and the integration in integral equation is from minus to plus infinity

$$\boxed{D(\tau) = \int_{-\infty}^{\infty} \mathcal{K}_{\mathrm{B}}(\tau,\omega) A(\omega) d\omega} \; . \tag{78}$$

In this case there is no additional symmetries to reduce the equation.

7.4.1 Phonons

Phonon green function $D(\omega)$ is the displacement-displacement correlation function which assumes property $A(-\omega) = -A(\omega)$. This property follows from general properties of correlation functions $< \hat{A}(\tau)\hat{B} >$ when $\hat{A} = \hat{B}$. In particular, this property is required to get properly unperturbed phonon Green function on imaginary time

$$D^{(0)}(\tau) = \frac{e^{\beta\omega_0} e^{-\tau\omega_0} + e^{\tau\omega_0}}{e^{\beta\omega_0} - 1} = \frac{\cosh\left[\left(\frac{\beta}{2} - \tau\right)\omega_0\right]}{\sinh\left[\frac{\beta\omega_0}{2}\right]} \tag{79}$$

Note, phonon green function (79) can be obtained from equation (77) only if $A(\omega) = \delta(\omega - \omega_0) - \delta(\omega + \omega_0)$. Then, one can define the equation for positively definite ω

$$D(\tau) = \int_{0}^{\infty} \frac{\cosh\left[\left(\frac{\beta}{2} - \tau\right)\omega\right]}{\sinh\left[\frac{\beta\omega}{2}\right]} A(\omega) d\omega \; . \tag{80}$$

It is convenient to introduce function $\widetilde{A}(\omega)$ definet through

$$\boxed{A(\omega) = \tanh\frac{\beta\omega}{2}\widetilde{A}(\omega)} \tag{81}$$

which gives possibility to rewrite Eq. (80) as

$$\boxed{D(\tau) = \int_0^\infty \frac{\cosh\left[\left(\frac{\beta}{2} - \tau\right)\omega\right]}{\cosh\left[\frac{\beta\omega}{2}\right]} \widetilde{A}(\omega) d\omega} \tag{82}$$

with obvious normalization condition

$$\boxed{\int_0^\infty \widetilde{A}(\omega) d\omega = D(0)} \tag{83}$$

So, to get $A(\omega)$ one obtains $\widetilde{A}(\omega)$ from Eq. (82) with normalization (83) and uses relation (81) .

7.5 Matsubara representation

The general spectral representation for the Green function of complex frequency z is

$$G(z) = \int_{-\infty}^{\infty} \frac{A(\omega')}{\omega' - z} d\omega' \ . \tag{84}$$

For real $z \to \omega$

$$Im(G(\omega)) = -\pi A(\omega) \ . \tag{85}$$

If the function is given on Matsubara frequencies, $\omega_n = (2\pi/\beta)n$ for Bose and $\omega_n = (2\pi/\beta)(n+1/2)$ for Fermi statistics, the complex value of $G(i\omega_n)$ is given by

$$G(i\omega_n) = \int_{-\infty}^{\infty} \frac{A(\omega')}{\omega' - i\omega_n} d\omega' \ . \tag{86}$$

The value of Green function at imaginary time $G(\tau)$ can be obtained from Matsubara function

$$G(\tau) = \sum_0^N e^{i\omega_n \tau} G(i\omega_n) \ . \tag{87}$$

8 Manual for operating of the set of codes

8.1 Description of Fortran-90 and input files in the set

Main code Fortran-90 files

```
xenon02.f90           code to compile with Fortran-90
xenon02upd.f90        code to include into xenon02.f90
xenon02aux.f90        code to include into xenon02.f90
xenon02cc.f90             code to include into xenon02.f90
```

Main code input files

```
control.in       control parameters file

infloa.in        input correlation function G(tau) in format:

          Kernels 0, 1, 3:
                 First line is number of points.
                 From second line there are 3 columns
                 t , G, relative error.

          Kernels 2:
                 First line is 3 numbers: number of all points, number
                 of points used in fit from the beginning, and
                 number of points used in fit from the end of file,.
                 From second line there are 3 columns
                 t , G, relative error

matsubara.in     correlator G(n) in Matsubara
          Kernels 4, 5:
                 Representation in format:
                 first line is number of points.
                 From second line there are 4 columns
                 n , Re[G(n)], Im[G(n)], relative error

anzac_1.in       delta-functional anzac parameters

manupre.in       increase precision manually by number
```

Data processing code Fortran-90 file

```
obrab_x01.f90
```

Data processing control files

```
precision.in   file selecting how to handle normalization and precision

brek.in        contains integer"n" enlarging the histogram cell
               width "n" times. Smoothened spectral files
               contain "_tr" symbol.
```

8.2 Output files after processing

Output from `"obrab_01.f"` creates following data files

final.dat	Final spectral function for selected range 1 and grid 1, file is used to make a coarse grid and determine the errorbars
final_tr.dat	Smoothened final spectral function for selected range 1 and grid 1
final_big.dat	Final spectral function for selected range 2 and grid 2, file is used to make the smoothened function
final_tr_big.dat	Smoothened final spectral function for selected range 2 and grid 2
measur.dat	Distribution of objective function values in the set of obtained solutions.
deviat.dat	Distributuion of maximal value of deviation function in the set of obtained solutions
normalic.dat	For all successful solutions there are three columns: normalization, objective, kappa
errobars.dat	For "l" bins from 4-th line of "precision.in" uses the following spectral characteristics of the coarse grained spectrum: center energy of bin (c), average of all independent solutions (a), standard deviation(s), half width of the bin (h). The columns are: c, a-s, a, a+s
errobars_GNU.dat	The columns are: c, a, c-h, c+h, a-s, a+s Good for gnuplot.
errobars_GNU2.dat	The columns are: c, a, a-s, a+s Good for gnuplot.
errobars_ORI.dat	The columns are: c, a, s, h Good for ORIGIN.
smooth.dat	Smoothened spectrum (can be over-smoothed)

In the above files ”objective” means Eq. (61) for $\mathcal{D}$ Eq. (62) and ”kappa” means Eq. (57).

8.3 Description of the main control file ”control.in”

The file ”control.in” looks as follovs

```
5000                !Max number of attempts to find spectrum
0                   !Use previous statistics (if /=0)
0                   !Start refinement procedure (if /=0)
0                   !SuperRefinement (if /=0)
0                   !Use refined statistic (if /=0)
0                   !Write Intermediate Data, (if /=0)
0.9999 1.0001       !Min/Max normalization at initialization
0.0 5.0 50          !Low/Upp borders for free spectrum,HeReGrid
0.0 5.0  1000        !Low/UppHistRange,HistGrid - 1
0.0 5.0  50        !Low/UppHistRange,HistGrid - 2
60                  !MaxNumberOfRectangles
9                   !Number of attempts to find precision
100                 !Number of global updates for one attempt
0                   !Kernel Type: 0, 1, 2, 3, 4, 5
50.0  1.1              !InverseTemperature, phi
10                  !start CC only number of frequencies >
3                   !How many times to start CC
1.0d5 1.d3 1.d1 1.d10 1.d-1 1.0d-2 1.0d-3 1.0d-4 ! Below which O’s
100 170       !Complex and maginary included
697911              !Initial RNGP, if no statistics (odd)
```

The meaning of the parameters in the following

```
5000                !Max number of attempts to find spectrum
```

Maximal number of attempts L to get independent solutions. Number L_{good} in Eq. (56) is a number of successful attempts which is usually smaller than the number of attempts $L_{\text{good}} < L$.

```
0                   !Use previous statistics (if /=0)
```

Continues calculations in normal procedure with previously saved statistics if =1, starts new one if =0.

```
0                   !Start refinement procedure (if /=0)
```

Start refinement procedure using the independent solutions, obtained in normal procedure, if ($\neq 0$).

```
0                   !SuperRefinement (if /=0)
```

Make superrefinement procedure (previous parameter “refinement” must be $\neq 0$). Namely, independent solutions, obtained in refinement procedure are taken as the input for for new refinement process. Warning (!!): if this parameter is ($\neq 0$), the result-files for of independent solutions, which are found in normal procedure, are rewritten. Namely, files with results obtained in refinement procedure are put into files for normal procedure. Therefore, put this parameter ($\neq 0$) only once when start superrefinement. Then, if restart with continuation of refinement, set it to zero again to prevent the files to be rewritten again.

```
0                  !Use refined statistic (if /=0)
```

Continues refinement procedure with previously saved statistics if =1, starts new refinement if =0.

```
0                  !Write Intermediate Data, (if /=0)
```

If ($\neq$ 0), code write files for graphical control of data quality after every global update. This files are good to control what is going on at current time with a fitting process. The first file is 'dev_flo.dat' which contains written in columns deviation functions (33) $\mathcal{D}(m)$

$$\begin{array}{ll} \tau_1 & \mathcal{D}(1) \\ & \ldots\ldots\ldots\ldots\ldots\ldots \\ \tau_M & \mathcal{D}(M) \end{array}$$

This file is good to test whether convergence is good. If convergence can be reached, the curve fluctuates around zero. If convergence is perfect, curve changed sign almost at every point, i.e. large κ in Eq. (57). If not, the oscillations are much larger than the distance between neighboring τ_i points. If after long time the curve still has more deviation into one of (positive/negative) directions, convergence can not be reached. If so, then, for example, the range of spectral function has to be increased or normalization is incorrect. Two other files are 'his_flo.dat' and 'his_flo_big.dat' which is immediate configuration of spectral function after the last global update. Actually, after control is made, it is better to set this parameter to zero because the writing procedures slow down the code! The files can be monitored by Gnuplot using file "p.dat".

```
0.9999 1.0001   !Min/Max normalization at initialization
```

Ths is the range in which normalization N of the spectral function $A(\omega)$ $N = \int A(\omega)d\omega$ is generated for every attempt to find independent solution. This normalization does not change during the process. This is used when the normalization N is not known exactly and one has to search for proper normalization. An explanation how to establish the proper normalization in case of unknown one is given below.

```
0.0 5.0 50       !Low/Upp borders for free spectrum,HeReGrid
```

Two first numbers show where the actual spectral function (excluding anzac) is confined. Last number is a number of bins for procedure to restrict heights of rectangles in refinement procedure. Note, if the parameter $WriteIntermediateData \neq 0$, one can look at current spectral function $A(\omega)$ in the file "his_flo.dat". If a substantial part of spectral function is glued to one of the limits, the range has to be enlarged in that direction.

```
0.0 5.0  1000    !Low/UppHistRange,HistGrid - 1
                      Third number should be in range [1000,4000]
0.0 5.0  50    !Low/UppHistRange,HistGrid - 2
                      Third number <=50
```

a b c = Range [a,b] with c-cells for collecting spectral function statistics, $b > a$ for two different histograms.

```
60                !MaxNumberOfRectangles
```

Max number of rectangles used for spectral function during the process of obtaining one independent solution. Since the current version of the code contains consistent constraint procedures, it is recommended to keep this number < 70.

```
9                 !Number of attempts to find precision
```

Number of attempts ($<= 19$) to find solution after which the reasonable precision of fit to specific quality of input data is decided automatically.

```
100               !Number of global updates for one attempt
```

Number of global updates to get one independent solution in one attempt.

```
0                 !Kernel Type: 0, 1, 2, 3, 4, 5
```

Kernel types:

=0 Subsections 7.1 (one particle at zero temperature) and 7.2.2 (optical conductivity at zero temperature).

=1 Subsection 7.2.1 (optical conductivity at nonzero temperature).

=2 Subsection 7.3 (fermions at finite temperature).

=3 Subsection 7.4.1 (phonons at nonzero temperature).

=4 Subsection 7.5 (Fermions in Matsubara representation).

=5 Subsection 7.5 (Bosons in Matsubara representation).

=6 Under development...

=7 Under development...

```
50.0     1.1        !InverseTemperature, phi
```

Inverse temperature β. This parameter has no effect for zero temperature kernels. Angle ϕ is for kernels 6 and 7, no effect for kernels ≤ 5.

```
10                !start CC only number of frequencies >
```

Code to start consistent constraints procedures if number of rectangles is larger than the value of this parameter.

```
4                 !How many times to start CC
```

How many times to start consistent constraints procedures during finding one independent solutions.

```
1.0d5 1.d3 1.d1 1.d10 1.d-1 1.0d-2 1.0d-3 1.0d-4 ! Below which O’s
```

Start next consistent constraint when the objective is smaller that the given number in array. Note, number of values in the row must be not less than integer number in the previous line. Note, number must be in descending order.

```
100 170   ! Complex and imaginary included
```

Data required for kernels 6 nd 7, underdevelopment, no effect for kernels ≤ 5.

```
697911          !Initial RNGP, if no statistics (odd)
```

Seed number for random number generator, if code starts first time. It must be odd.

8.4 Description of the file ”anzac 1.in”

The file looks as follows.

```
0.09 0.11    om_anz_min,om_anz_max
0.19d0 0.21d0     z_anz_min,z_anz_max
```

Two numbers in the first line are minimal and maximal energy of the δ-functional part of the spectrum. Two numbers in the second line are minimal and maximal Z-factor of the δ-functional part of the spectrum. All anzac operation are excluded in code if the mimimum of anzac Z-factor is less that 10^{-20}.

8.5 Description of the file ”manupre.in”

This file contains one number r with the following meaning. The initial procedure of code after *”number of attempts to find precision”* find the maximal objective O_{max} below which found solution is considered to be accepted. If the code is resumed from the place where it was stopped, i.e. uses previous statistics, then $O_{\text{max}} \to O_{\text{max}}/r$. I.e. the better precision is requested for $r > 1$. Has no effect if the code is started from the onset.

8.6 Description of the file ”precision.in”

The file looks as below.

```
.true.
.false. 0.0 1.0
.false. 0.0 1.0
20
50
1.0
1.0d2
```

line 1 Processes data from normal procedure if .false. and data from refinement procedure if .true. .

line 2 Imposing constraint to solutions (if .true.) by taking into account only ones with objectives between second and third numbers in the line. If .false., no constraints imposed.

line 3 Imposing constraint to solutions (if .true.) by taking into account only ones with normalizations between second and third numbers in the line. If .false., no constraints imposed.

line 4 Makes a coarse grained grid with number of bins given as integer number above and at every center of bin the average value of spectrum and errorbars are determined. The results are in files ”errorbars.dat”, ”errorbars_GNU.dat”, and ”errorbars_GNU2.dat”.

line 5 Number of solutions (the best ones) for attempt to make smoothing.

line 6 Reduce to 0.5. 0.25, 0.1 etc if over-smoothed in ”smooth.dat”

line 7 Increase to 1.0d6 etc if over-smoothed in ”smooth.dat”.

8.7 Description of the file ”brek.in”

Contains one integer number ”n” to create smoothened spectral files with enlarged bin width by ”n”. All spectral files of such kind have symbol ”_tr” in names.

9 Manual for preparing tests for various kernels

The tests for different kernels can be prepared with the code ”test_maker_x01.f90” which uses input file ”inpu_test.in”. An example of the file is given below

```
0                     ! type of kernel
4.0d0 10000      ! range of spectra & number of omega points
0.1d0  0.2d0     ! center & weight of anzac
1.0 1.7 3.0 0.5  ! beginning, middle, end, and weight of triangle 1
1.0 2.4 3.0 0.3 ! beginning, middle, end, and weight of triangle 2
0.5 0.75 1.0 0.0 ! beginning, middle, end, and weight of triangle 3
40.0 170          ! max imaginary time & number of points
1.0d-4 1.0d-4      ! relative errorbar complex and imaginary
2.0                   ! inverse temperature
1.0d0      ! intended norma
20.0 100    ! max complex time & number of points
1.1d0      ! Phi
```

An explanation of the parameters meaning is given below.

```
0                     ! type of kernel
```

Type of kernel in the range 1-5.

```
4.0d0 10000      ! range of spectra & number of omega points
```

Range of spectrum and number of omega points. In case of spectra with kernels 0, 1, and 3 the range is $[0, 4]$ and for kernels 2, 4, and 5 the range is $[-4, 4]$.

```
0.1d0  0.2d0     ! center & weight of anzac
```

This is the central frequency and Z-factor of δ-functional component of the spectral function, Note, δ-functional component can be present only for kernel type 0. Note, that center must be within the range of spectrum.

```
1.0 1.7 3.0 0.5  ! beginning, middle, end, and weight of triangle 1
1.0 2.4 3.0 0.3 ! beginning, middle, end, and weight of triangle 2
0.5 0.75 1.0 0.0 ! beginning, middle, end, and weight of triangle 3
```

Here one has to include to the spectral function up to 3 triangles into the spectral function, each presented by the lowest, middle peak, highest frequency and weight. Note, lowest and highest frequencies must be within the range.

```
40.0 170           ! max imaginary time & number of points
```

The maximal imaginary time 40.0 and number of imaginary time points 170 to create the imaginary time input file ”infloa.in” for kernels 0, 1, 2, 3. For the kernels 4 and 5 the number 170 is the number of Matsubara frequencies in file ”matsubara.in”. The input files can be used for calculations as prepared except for the kernel 2 where one has to add into the first line two more numbers. First line already contains the total number of imaginary time points. One has to add number of points used in fit from the beginning and number of points used in fit from the end of file.

```
1.0d-4 1.0d-4      ! relative errorbar complex and imaginary
```

These are values of relative errorbars to input into data files because input file structures assume errorbar for every data point.

```
2.0                     ! inverse temperature
```

Inverse temperature β. Ignored for $T = 0$ kernels.

```
1.0d0       ! intended norma
```

The final spectral function is normalized to the number specified in this line. The initial real frequency spectral function is in the file ”spe.dat” to control how the code of spectral analysis can restore the spectrum from imaginary of Matsubara representation.

```
20.0 100    ! max complex time & number of points
```

Ignored for all kernels ≤ 5.

```
1.1d0       ! Phi
```

Ignored for all kernels ≤ 5.

9.1 How to test

The general procedure is that one edit the file ”inpu_test.in” and runs the routine ”test_maker_x01.f90”. The routine creates files ”spe.dat” with a file of real-frequency spectra and file to be analytically continued (”infloa.in” for kernels 0, 1, 2, 3 and ”matsubara.in” for kernels 4, 5. E.g. see ”se-tau.gp” to check files in Gnuplot.

10 Tutorials

To get the fast knowledge how to work with code it is better to pass through the most detailed tutorials for kernels 0 and 1 even if you need some other kernel. The operation for the rest of kernels follows form from the above two tutorials.

10.1 Kernel 0

1 Copy the file ”inpu_test.in_Ker_0_set_0” into file ”inpu_test.in” and execute the code ”test_maker_x01.f90”. The code creates real-frequenct spectrum ”spe.dat” and imaginary time file ”infloa.in” which can be seen by Gnuplot using ”setau.gp”. Copy ”p.gp_Ker_0” to ”p.gp” and ”fp.gp_Ker_0” to ”fp.gp”.

2 Copy the file ”control.in_Ker_0_att_1” to the file ”control.in”. Note, parameter ”!Write intermediate data” is set to 1, and, hence, the files ’dev_flo.dat’ and ’his_flo.dat’ are written at every step. These files can be seen usung Gnuplt and file ”p.gp”.

3 Copy the file ”anzac_1.in_Ker_0_att_1” to the file ”anzac_1.in”.

4 Copy the file ”brek.in_Ker_0_att_1” to the file ”brek.in”.

5 Copy the file ”precision.in_Ker_0_att_1” to the file ”precision.in”.

6 Start code ”xenon02.f90”. While code is running, check the files ’dev_flo.dat’ and ’his_flo.dat’. One can see that the curve in ’dev_flo.dat’ is far from zero and almost do not crosses zero. At the same time ’his_flo.dat’ is not similar to ”spe.dat”. Instead, there is a peak in ’his_flo.dat’ at the frequency ≈ 2. This patalogical behavior is because the boders of free spectrum

```
0.0 2.0 50  !Low/Upp borders for free spectrum,HeReGrid
```

is itensionally taken too small in the range [0.0, 2.0]. It has to be increased to [0.0, 4.0].

7 Increase the range or, alternatively, copy ”control.in_Ker_0_att_2” to the file ”control.in”. The range is now set correctly.

8 Start executing ”xenon02.f90”. There is a line in the printout

```
 RUN =    -6    SUCCES=     0    MAXRUN=8000
```

where the index ”RUN” is negative during first few minutes. Wait few minutes till it becomes positive like

```
  RUN =    32    SUCCES=    27    MAXRUN=8000
```

Here the index ”SUCCESS” shows how many solutions are already found.

9 While the code ”xenon02.f90” is running (It will be finished when RUN=MAXRUN but one can stop it any time after satisfied with the collected statistics) one can check collected statitics executing file ”obrab_x01.f90” and looking at the results using file ”fp.gp” with Gnuplot. The output files shown by ”fp.gp” are described in section 8.2. Note, ”fp.gp” review summary statistics in contrast with ”p.gp” which shows the current configuration.

10 Now it is time to start industrial calculation. One stops ”xenon02.f90” and copy ”control.in_Ker_0_att_3” to the file ”control.in”. Review changes in file ”control.in” in the lines 2 and 6. we stopped write intermediate data about current configuration to accelerate the code and we will start from already collected statistics after we restart ”xenon02.f90”.

11 Restart ”xenon02.f90”. Note, accumulation of statistics continues which one can verify running ”obrab_x01.f90” and looking at the results using file ”fp.gp” with Gnuplot.

12 After SUCCESS larger than, say, 500 or 1000, one can strart refinement, see section 3.6. To do so, stop the code ”xenon02.f90” (If it is not has stopped already because RUN=MAXRUN). Then, copy ”control.in_Ker_0_att_4” to the file ”control.in” and review changes in lines 2 and 3. Then, start again ”xenon02.f90”. Note, statistics accumulation for refinement, i.e. increase of the index ”SUCCES=” is much faster than in the previous stage.

13 To get access to the statistics of the refinement procedure copy the file ”precision.in_Ker_0_att_2” to the file ”precision.in”. Alternatively, change parameter ”.false.” to ”.true.” in the file ”precision.in”. Then, refinement statistics is analyzed when one runs ”obrab_x01.f90”. The output files are similarly can be seen by ”fp.gp”.

14 One can even get super-refinement of the statistics, which is not always necessary. In this case the set of the refined configurations is rewritten to a database of the solutions found at the first stage and refinement procedure repeated. Note, in this case the statistics of the first stage is lost because it is overwritten by the refined one. To make super-refinement, stop the code ”xenon02.f90” (If it is not has stopped already because RUN=MAXRUN). Then, copy ”control.in_Ker_0_att_5” to the file ”control.in” and review changes in lines 3 and 4. Start again ”xenon02.f90”.

10.2 Kernel 1

The main new feature of this kernel in comparison with previous one is that the exact normalization of the spectrum $A(\omega)$ is usually not known. Usually, one can get an information using sum rules and additional calculation (e.g. the integral weight of the optical conductivity is related to the avetage kinetic energy [64]) though precision of sum rule integral weight calculations can be not enough to fix the normalization. It means, that although for each of kernels it can be an additional information about the normalization, e.g. equations (66, 72, 74, 83) though the error-bars are still too large to find an optimal normalization. In this case one has to find the

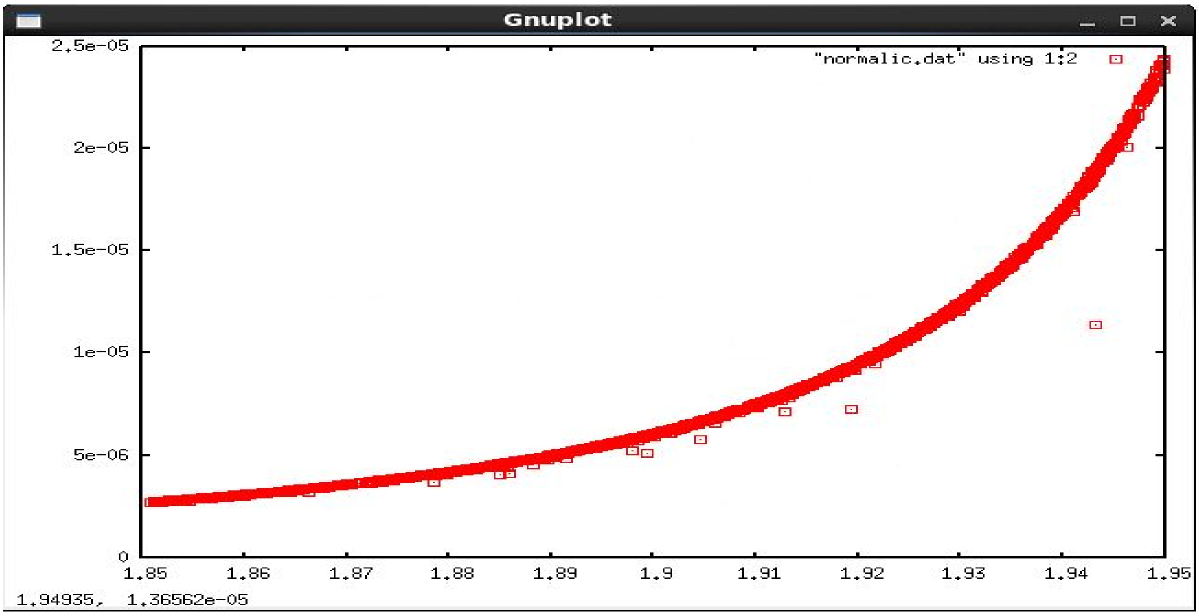

Fig. 12: *Ddistribution of the function $D^{-1}(N)$ which can be found in file "normalic.dat", N is the first column and D^{-1} is the second. This picture is for file "control.in_Ker_1_side".*

normalization minimizing the objective function. The code has necessary features to perform such job.

In current tutorial we assume the worst case when the normalization is not known at all and try to find it starting from any arbitrary range of parameters. The procedure is based on the changing parameters in line 7 of the file "control.in".

```
1.7 1.95    !Min/Max normalization at initialization
```

The normalization is initialized with uniform probability in the range [1.7, 1.95] in the beginning of every attempt to find a a solution and it is not changed during the minimization of the objective measure. So, after obtaining some M solution one can review the properties of distribution of inverse objective functions D^{-1} (61) versus normalization N. Indeed, inverse objective function has maximum at correct normalization N_0. To see this distribution one has to look at the function $D^{-1}(N)$ which can be found in file "normalic.dat", N is the first column and D^{-1} is the second. Note, one should plot the graph with "symbols", not "line plus symbols", because the data in file are not monotonic in normalization N.

1 Copy the file "inpu_test.in_Ker_1_set_0" into file "inpu_test.in" and execute the code test_maker_x01.f90". The code creates real-frequenct spectrum "spe.dat" and imaginary time file "infloa.in" which can be seen by Gnuplot using "setau.gp". Copy "p.gp_Ker_1" to "p.gp" and "fp.gp_Ker_1" to "fp.gp".

2 Copy the file "control.in_Ker_1_side" to the file "control.in". Note, parameter "!Write intermediate data" is set to 0, and, hence, the files 'dev_flo.dat' and 'his_flo.dat' are not written at every step. You may chage this parameter to 1 of you want to see the files

3 Copy the file ”anzac_1.in_Ker_1_att_1” to the file ”anzac_1.in”. Note, z-factors limits of anzac weight re set both to zero. Indeed, δ-functional anzac applies only to kernel 0.

4 Copy the file ”brek.in_Ker_1_att_1” to the file ”brek.in”.

5 Copy the file ”precision.in_Ker_1_att_1” to the file ”precision.in”.

6 Start code ”xenon02.f90”. Now, our file ”control.in” contains normalization range [1.7, 1.95]. Wait for a while untill SUCCESS>100. Looking on the distribution in file ”normalic.in” one can readily diagnose that the normalization $N_0 > 1.95$ (see Fig. 12). Note, the logarithmic scale is preferable for D^{-1}.

7 Copy the file ”control.in_Ker_1_wide” to the file ”control.in” where we take wide range to the right from 1.9, i.e. [1.9, 2.1]. Then, start code ”xenon02.f90”. One can see after 10 minutes that the maximum is around $N_0 \approx 2$ (see Fig. 12).

8 You can copy the file ”control.in_Ker_1_narrow” to the file ”control.in” with more narrow range about $N_0 \approx 2$ and see that $N_0 = 2$ after few minutes of running of ”xenon02.f90”.

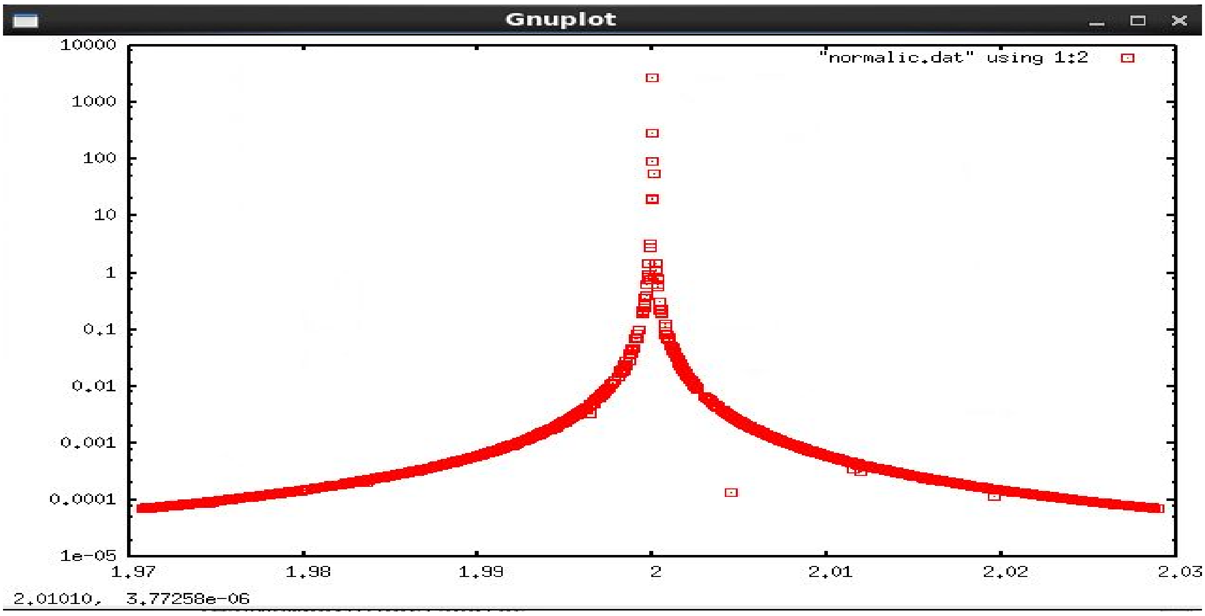

Fig. 13: *Distribution of the function* $D^{-1}(N)$ *in logarithmic scale which can be found in file ”normalic.dat” for file ”control.in_Ker_1_wide”.*

9 Now we can start industrial calculations copying ”control.in_Ker_1_exa” to the file ”control.in” where we take extremely narrow range

```
1.99999 2.000001   !Min/Max normalization at initialization
```

Note, difference between maximal and minimal normalizations can be small but must be nonzero because of construction of code. From now on one has industrial accumulation of data.

10 Refinement and super-refinement of the data, if necessary, is made in the same way as for kernel 0.

11 We remind that for general understanding of the code it is useful to pass first the tutorial for kernel 0.

10.3 Kernel 2

Peculiarity of the kernel 3 is that one is often in the situation that the statistic of Green function in imaginary time is good near $\tau = 0$ and near $\tau = \beta$. On the other hand the quality of the data in the middle between 0 and β can be very bad because Green function has very small value in that region. Hence, to handle this situation code requests three digits in the first line of the file ”infloa.in”. For example, if the first line looks like

```
250 100 50
```

then, the file contains 250 data points in imaginary time and the points used for analysis are first 100 points and last 50 points. One can include all points to analysis setting sum of two last digits equal to the first one.

1 Copy the file ”inpu_test.in_Ker_2_set_0” into file ”inpu_test.in” and execute the code test_maker_x01.f90”. The code creates real-frequenct spectrum ”spe.dat” and imaginary time file ”infloa.in” which can be seen by Gnuplot using ”setau.gp”. Copy ”p.gp_Ker_2” to ”p.gp” and ”fp.gp_Ker_2” to ”fp.gp”.

2 Note, the first line of the file ”infloa.in” contains the digit ”250”. To transform file to correct input for given kernel one has to change this digit into three digits

```
250 150 100
```

or just copy file ”infloa.in_Ker_2” into file ”infloa.in”

3 Copy the file ”control.in_Ker_2_att_1” to the file ”control.in”. Note, parameter ”!Write intermediate data” is set to 0, and, hence, the files ’dev_flo.dat’ and ’his_flo.dat’ are not written at every step. You may chage this parameter to 1 of you want to see the files

4 Copy the file ”anzac_1.in_Ker_2_att_1” to the file ”anzac_1.in”. Note, z-factors limits of anzac weight re set both to zero. Indeed, δ-functional anzac applies only to kernel 0.

5 Copy the file ”brek.in_Ker_2_att_1” to the file ”brek.in”.

6 Copy the file ”precision.in_Ker_2_att_1” to the file ”precision.in”.

7 Start code ”xenon02.f90”.

8 Refinement and super-refinement of the data, if necessary, is made in the same way as for kernel 0.

9 We remind that for general understanding of the code it is useful to pass first the tutorial for kernel 0.

10.4 Kernel 3

1 Copy the file ”inpu_test.in_Ker_3_set_0” into file ”inpu_test.in” and execute the code test_maker_x01.f90”. The code creates real-frequenct spectrum ”spe.dat” and imaginary time file ”infloa.in” which can be seen by Gnuplot using ”setau.gp”. Copy ”p.gp_Ker_3” to ”p.gp” and ”fp.gp_Ker_3” to ”fp.gp”.

2 Copy the file ”control.in_Ker_3_att_1” to the file ”control.in”. Note, parameter ”!Write intermediate data” is set to 0, and, hence, the files ’dev_flo.dat’ and ’his_flo.dat’ are not written at every step. You may chage this parameter to 1 of you want to see the files

3 Copy the file ”anzac_1.in_Ker_3_att_1” to the file ”anzac_1.in”. Note, z-factors limits of anzac weight re set both to zero. Indeed, δ-functional anzac applies only to kernel 0.

4 Copy the file ”brek.in_Ker_3_att_1” to the file ”brek.in”.

5 Copy the file ”precision.in_Ker_3_att_1” to the file ”precision.in”.

6 Start code ”xenon02.f90”.

7 Refinement and super-refinement of the data, if necessary, is made in the same way as for kernel 0.

8 We remind that for general understanding of the code it is useful to pass first the tutorial for kernel 0.

10.5 Kernel 4

1 Copy the file ”inpu_test.in_Ker_4_set_0” into file ”inpu_test.in” and execute the code test_maker_x01.f90”. The code creates real-frequenct spectrum ”spe.dat” and matsubara frequency file ”matsubara.in” which can be seen by Gnuplot using ”setau.gp”. Copy ”p.gp_Ker_4” to ”p.gp” and ”fp.gp_Ker_4” to ”fp.gp”.

2 Copy the file ”control.in_Ker_4_att_1” to the file ”control.in”. Note, parameter ”!Write intermediate data” is set to 0, and, hence, the files ’dev_flo.dat’ and ’his_flo.dat’ are not written at every step. You may chage this parameter to 1 of you want to see the files

3 Copy the file ”anzac_1.in_Ker_4_att_1” to the file ”anzac_1.in”. Note, z-factors limits of anzac weight re set both to zero. Indeed, δ-functional anzac applies only to kernel 0.

4 Copy the file ”brek.in_Ker_4_att_1” to the file ”brek.in”.

5 Copy the file ”precision.in_Ker_4_att_1” to the file ”precision.in”.

6 Start code ”xenon02.f90”.

7 Refinement and super-refinement of the data, if necessary, is made in the same way as for kernel 0.

8 We remind that for general understanding of the code it is useful to pass first the tutorial for kernel 0.

10.6 Kernel 5

1 Copy the file ”inpu_test.in_Ker_5_set_0” into file ”inpu_test.in” and execute the code test_maker_x01.f90”. The code creates real-frequenct spectrum ”spe.dat” and matsubara frequency file ”matsubara.in” which can be seen by Gnuplot using ”setau.gp”. Copy ”p.gp_Ker_5” to ”p.gp” and ”fp.gp_Ker_5” to ”fp.gp”.

2 Copy the file ”control.in_Ker_5_att_1” to the file ”control.in”. Note, parameter ”!Write intermediate data” is set to 0, and, hence, the files ’dev_flo.dat’ and ’his_flo.dat’ are not written at every step. You may chage this parameter to 1 of you want to see the files

3 Copy the file ”anzac_1.in_Ker_5_att_1” to the file ”anzac_1.in”. Note, z-factors limits of anzac weight re set both to zero. Indeed, δ-functional anzac applies only to kernel 0.

4 Copy the file ”brek.in_Ker_5_att_1” to the file ”brek.in”.

5 Copy the file ”precision.in_Ker_5_att_1” to the file ”precision.in”.

6 Start code ”xenon02.f90”.

7 Refinement and super-refinement of the data, if necessary, is made in the same way as for kernel 0.

8 We remind that for general understanding of the code it is useful to pass first the tutorial for kernel 0.

References

[1] M. Jarrell and J.E. Gubernatis, Phys. Rep. **269**, 133 (1996)

[2] R. Kress, *Linear Integral Equations* (Springer, New York, 1999)

[3] G.D. Mahan, *Many Particle Physics* (Plenum Press, New York, 1990)

[4] A Damascelli, Z. Hussain, and Z.X. Shen, Rev. Mod. Phys. **75**, 473 (2003)

[5] C. Huscroft, R. Gass, and M. Jarrell, Phys. Rev. B **61**, 9300 (2000)

[6] H. Matsui, A.S. Mishchenko, and T. Hasegawa, Phys. Rev. Lett. **104**, 056602 (2010)

[7] A.S. Mishchenko, H. Matsui, and T. Hasegawa, Phys. Rev. B **85**, 085211 (2012)

[8] J. Kaipio and S. Erkki, *Statistical and Computational Inverse Problems*, (Applied Mathematical Sciences Vol. 160) (Springer, Berlin, 2005)

[9] A.N. Tikhonoff and V.Y. Arsenin, *Solutions of Ill-Posed Problems* (Winston & Sons, Washington, 1977)

[10] G.H. Golub and C. Reinsch, Numerische Mathematik **14**, 403 (1970)

[11] R.T. Cox, *The Algebra of Probable Inference* (Johns Hopkins University Press, 1961); A. Papoulis, *Probability and Statistics* (Prentice Hall, New York, 1990)

[12] A.N. Tikhonoff, Dokladyu Akademii Nauk SSSR **39**, 195 (1943)

[13] A.N. Tikhonoff, Dokladyu Akademii Nauk SSSR **151**, 501 (1963) (Soviet Mathematics **4**, 1035 (1963))

[14] D.L. Phillips, J. ACM **9**, 84 (1962)

[15] A.E. Hoerl, Chemical Engineering Progress **58**, 54 (1962)

[16] A.E. Hoerl and R.W. Kennard, Technometrics **12**, 55 (1970)

[17] M. Foster, J. Soc. Industr. Appl. Math. **9**, 387 (1961)

[18] P.C. Hansen, Soc. Industr. Appl. Math. Rev. **34**, 561 (1992)

[19] P.C. Hansen and D.P. O'Leary, Soc. Industr. Appl. Math. J. Sci. Comput. **14**, 487 (1993)

[20] D. Krawchuk-Stando and M. Rudnicki, Int. J. Appl. Math. Comput. Sci. **17**, 157 (2007)

[21] I.S. Krivenko and A.N. Rubtsov, arXiv:cond-mat/0612233

[22] I.S. Krivenko and A.N. Rubtsov, JETP Lett. **94**, 768 (2012)

[23] M. Jarrell and O. Biham, Phys. Rev. Lett. **63**, 2504 (1989)

[24] S.R. White, D.J. Scalapino, R.L. Sugar, and N.E. Bickers, Phys. Rev. Lett. **63**, 1523 (1989)

[25] K. Vafayi and O. Gunnarsson, Phys. Rev B **76**, 035115 (2007)

[26] J. Skilling, J. Microsc. **190**, 28 (1998)

[27] A.W. Sandvik, Phys. Rev B **57**, 10287 (1998)

[28] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, J. Chem. Phys. **21**, 1087 (1953)

[29] O.F. Syljuasen, Phys. Rev B **78**, 174429 (2008)

[30] K.S.D. Beach, arXiv:cond-mat/0403055

[31] S. Fuchs, T. Pruschke and M. Jarrell, Phys. Rev E **81**, 056701 (2010)

[32] A.S. Mishchenko, N.V. Prokofév, A. Sakamoto and B.V. Svistunov, Phys. Rev. B **62**, 6317 (2000)

[33] S.R. White, *Computer Simulation Studies of Condensed Matter Physics III* (Springer, Heidelberg, 1991), p. 145

[34] E. Vitali, M. Rossi, L. Reatto and D.E. Galli, Phys. Rev. B **82**, 174510 (2010)

[35] D.M. Ceperley, J. Comput. Phys. **51**, 404 (1983)

[36] D.M. Ceperley and B.J. Alder, J. Chem. Phys. **81**, 5833 (1984)

[37] N.V. Prokofév, B.V. Svistunov, and I.S. Tupitsyn, Phys. Rev. Lett. **82**, 5092 (1999)

[38] A.S. Mishchenko, N.V. Prokof'ev, and B.V. Svistunov, Phys. Rev. B **64**, 033101 (2001)

[39] A.S. Mishchenko, N.V. Prokof'ev, A. Sakamoto, and B.V. Svistunov, Int. J. Mod. Phys. B **15**, 3940 (2001)

[40] A.S. Mishchenko, N. Nagaosa, N.V. Prokof'ev, A. Sakamoto, and B.V. Svistunov, Phys. Rev. B **66**, 020301(R) (2002)

[41] A.S. Mishchenko, N. Nagaosa, N.V. Prokof'ev, A. Sakamoto, and B.V. Svistunov, Phys. Rev. Lett. **91**, 236401 (2003)

[42] A.S. Mishchenko and N. Nagaosa, Phys. Rev. Lett. **93**, 036402 (2004)

[43] A.S. Mishchenko and N. Nagaosa, Phys. Rev. B **73**, 092502 (2006)

[44] A.S. Mishchenko and N. Nagaosa, J. Phys. Chem. Solids **67**, 259 (2006)

[45] G. De Filippis, V. Cataudella, A.S. Mishchenko, C.A. Perroni, and J.T. Devreese, Phys. Rev. Lett. **96**, 136405 (2006)

[46] A.S. Mishchenko, Proceedings of the International School of Physics "Enrico Fermi", Course CLXI, 177-206 (2006)

[47] A.S. Mishchenko and N. Nagaosa, Polarons in Complex Matter, Springer Series in Material Science, Springer, ed. by A.S. Alexandrov, 503-544 (2007)

[48] V. Cataudella, G. De Filippis, A.S. Mishchenko, and N. Nagaosa, Phys. Rev. Lett. **99**, 226402 (2007)

[49] A.S. Mishchenko, in "Computational Many-Particle Physics", ed. by H. Fehske, R. Scheider and A. Weisse, Lect. Notes Phys. 739, pp. 367-395 (Springer, Berlin Heidelberg 2008)

[50] A.S. Mishchenko, N. Nagaosa, Z.-X. Shen, G. De Filippis, V. Cataudella, T.P. Devereaux, C. Bernhard, K.W. Kim, and J. Zaanen, Phys. Rev. Lett. **100**, 166401 (2008)

[51] V. Cataudella, G. De Filippis, A.S. Mishchenko, and N. Nagaosa, J. Supercond. Nov. Magn., **22**, 17 (2009)

[52] A.S. Mishchenko, N. Nagaosa, A. Alvermann, H. Fehske, G. De Filippis, V. Cataudella, and O.P. Sushkov, Phys. Rev. B, **79**, 180301(R) (2009)

[53] A.S. Mishchenko, Usp. Phys. Nauk **179**, 1259 (2009) [Phys. Usp. **52**, 1193 (2009)]

[54] A.S. Mishchenko, Advances in Condensed Matter Physics **2010**, 306106 (2010)

[55] G.L. Goodvin, A.S. Mishchenko, and M. Berciu, Phys. Rev. Lett. **107**, 076403 (2011)

[56] A.S. Mishchenko, N. Nagaosa, K.M. Shen, Z.-X. Shen, X.J. Zhou, T.P. Devereaux, Europhys. Lett. **95**, 57007 (2011)

[57] G. De Filippis, V. Cataudella, A.S. Mishchenko and N. Nagaosa, Phys. Rev. B, **85**, 094302 (2012)

[58] S.S. Aplesnin, Zh. Eksp. Teor. Fiz. **124** 1080 (2003) [JETR **97** 969 (2003)]

[59] H. Hafermann, S. Brener, A.N. Rubtsov, M.I. Katsnelson, and A.I. Lichtenstein, J. Phys.: Condens. Matter **21**, 064248 (2009)

[60] H. Hafermann, M.I. Katsnelson, and A.I. Lichtenstein, Europhys. Lett., **85**, 37006 (2009)

[61] E. Gorelov, M. Karolak, T.O. Wehling, F. Lechermann, A.I. Lichtenstein, and E. Pavarini, Phys. Rev. Lett. **104**, 226401 (2010)

[62] E. Gorelov, J. Kolorenč, T. Wehling, H. Hafermann, A.B. Shick, A.N. Rubtsov, A. Landa, A.K. McMahan, V.I. Anisimov, M.I. Katsnelson, and A.I. Lichtenstein, Phys. Rev. B **82**, 085117 (2010)

[63] O. Gunnarssson et al, Phys. Rev B **81**, 155107 (2010)

[64] A. S. Mishchenko, N. Nagaosa, G. De Filippis, A. De Candia, and V. Cataudella, Phys. Rev. Lett. **114** 146401 (2015).