Ω*MaxEnt* tutorial

Dominic Bergeron

August 15, 2016

Contents

1 Introduction

This is the tutorial for Ω*MaxEnt*, a software for analytic continuation of Matsubara data using a maximum entropy approach. The tutorial teaches you step by step how to obtain the spectrum corresponding to fermionic or bosonic data, given in Matsubara frequency or imaginary time, with diagonal covariance. The program can also treat non-diagonal covariance, for which you can refer to the user guide, where you can also learn about the other functionalities. The program treats fermionic data with positive spectral functions $A(\omega)$, or bosonic data with $A(\omega)/\omega$ positive.

Before starting the tutorial, let us first give a few definitions. The functions used internally by the program are Matsubara frequency Green functions (or correlation functions). This, and the wide use of ω as the symbol for frequencies, is why it is called Ω*MaxEnt*. When the input data is in imaginary time, it is first Fourier transformed to obtain Matsubara frequency data before the main calculation starts. Thus, we can assume that the input data is always a function $G(i\omega_n)$. In that case, the analytic continuation problem consists in finding the spectrum $A(\omega)$ such that

$$G(i\omega_n) = \int_{-\infty}^{\infty} d\omega \, \frac{A(\omega)}{i\omega_n - \omega} \, . \tag{1.1}$$

The reason why we use this form instead of its Fourier transform

$$G(\tau) = -\int d\omega \frac{e^{-\omega\tau} A(\omega)}{1 \pm e^{-\beta\omega}} \, , \tag{1.2}$$

is that, for a piecewise polynomial approximation of $A(\omega)$, (1.1) can be integrated analytically in each interval, which is not the case of (1.2). This gives better accuracy and better scaling with temperature.

Now, one could try to invert (1.1) by discreitizing ω, using the same number of real and Matsubara frequencies, and solve the resulting linear system. This would however give very bad result because the system would be ill-conditioned. This means that very small variations in G can produce large variations in the solution A. Analytic continuation using

the maximum entropy principle uses a different approach, which consists in minimizing the function

$$Q = \chi^2 - \alpha S \,, \tag{1.3}$$

whith

$$\chi^2 = \sum_{mn} \left(G_m - K_m A\right)^T C^{-1}_{mn} \left(G_n - K_n A\right) \,, \tag{1.4}$$

where A is the vector obtained after discretizing ω, K_n is a row vector such that $K_n A$ is an approximation to (1.1), and C is the covariance matrix, which, for stochastic calculations, can be approximated by

$$C^{\mu\nu}_{mn} = \frac{1}{N-1} \sum_{i=1}^{N} (G^{\mu}_{im} - \bar{G}^{\mu}_{m})(G^{\nu}_{in} - \bar{G}^{\nu}_{n}) \,, \tag{1.5}$$

where

$$\bar{G}^{\mu}_{m} = \frac{1}{N} \sum_{i=1}^{N} G^{\mu}_{im} \,, \tag{1.6}$$

i is a sample index, N is the number of samples, and μ, ν refer to Re or Im in the case where G is complex.

S in (1.3) is a differential (or relative) entropy, defined as

$$S = -\int d\omega A(\omega) \ln \frac{A(\omega)}{D(\omega)} \,, \tag{1.7}$$

where $D(\omega)$ is called the *default model*, and α is a parameter to be determined. The solution that minimize (1.3) if χ^2 is negligible, namely that maximizes the entropy S, is $e^{-1}D(\omega)$, hence the name *default model*. $D(\omega)$ is defined in a way to include some of the information known in advance about the spectrum and not more.

Once the data, the covariance and the default model are defined, the solution that minimizes (1.3) is a function of α. What the program does is to compute the spectrum as a function of α for a large range of values. The optimal α is then determined by analyzing χ^2 as a function of α. In addition, some diagnostic tools can be used by the user to assess the quality of the result. Options are also available in the program to refine the solution.

Note that, if we diagonalize the covariance matrix C,

$$\tilde{\mathbf{C}} = \mathbf{U}^\dagger \mathbf{C} \mathbf{U}\,, \tag{1.8}$$

χ^2 can be written

$$\chi^2 = (\tilde{G} - \tilde{\mathbf{K}} A)^T (\tilde{G} - \tilde{\mathbf{K}} A)\,, \tag{1.9}$$

where

$$\begin{aligned} \tilde{G} &= \sqrt{\tilde{\mathbf{C}}^{-1}}\mathbf{U}^\dagger G\,, \\ \tilde{\mathbf{K}} &= \sqrt{\tilde{\mathbf{C}}^{-1}}\mathbf{U}^\dagger \mathbf{K}\,. \end{aligned} \tag{1.10}$$

This form is more useful for practical calculations and analysis of the results.

A detailed description of the algorithms and also benchmarks are presented in Ref. [3] or [2]. The latest version's source code and binaries are available on the program's web page.[1]

2 Parameter files

There are two parameter files that are used to interact with the program: The main parameter file *OmegaMaxEnt_input_params.dat*, and *OmegaMaxEnt_other_params.dat*, which contains the internal parameters used by the program. The default value of the parameters in the latter file is optimal for most cases however and you will not modify it in this tutorial.

It is important not to modify the parameter names in the parameter files, namely, any caracter up to ':' on a line. If you modify a parameter and this parameter is not printed on screen when the program is executed, it means you modified its name and the program cannot find it anymore. If that happens, simply create a new default version of the parameter file by destroying the file and executing the program again.

The program can display figures. When it happens, execution is paused, and you have to close all the figures to resume it.

3 Example 1: fermionic Matsubara frequency data

If a file *OmegaMaxEnt_input_params.dat* is present in the program's directory, erase it, then execute file *OmegaMaxEnt* to create a blank version, and open the file in a text editor.

To define the input data, first set the parameter "input directory:" in section INPUT FILES PARAMETERS to *tutorial_examples/example_1*. Then, copy the name of the file starting with *Green_test_diag_err* in that directory after "data file:" on the first line and the name of the file starting with *error_Green_test* after "error file:" in section INPUT FILES PARAMETERS. Also, set the parameter "reference spectral function file" in section OPTIONAL MINIMIZATION TIME PARAMETERS/DISPLAY OPTIONS to the file name starting with *test_spectr_func*. This file contains the exact spectrum. It will then be plotted with the result to compare the two spectra.

Before computing the spectrum, let us look first at the preprocessing stage. In section PREPROCESSING EXECUTION OPTIONS, set the parameters "preprocess only" and "display preprocessing figures" to *yes*. Save *OmegaMaxEnt_input_params.dat* and execute the program (file *OmegaMaxEnt*).

Note first that the parameters modified in the parameter files are printed in the terminal. Then, some information on the data and the preprocessing steps are displayed. Figures 1 and 2 show the real and imaginary parts of the data, respectively. To continue execution, close the figures. Figures 3 and 4 show the errors for $Re[G]$ and $Im[G]$, respectively. Then, after closing the figures again, figures 5 and 6 show the norm M_0 and the first moment M_1 computed by simple local fits to the data. Those curves are used to find the onset frequency of asymptotic regime, where the plateaus appear (the actual values of moments used by the code are obtained with a non-local least-square fit). Figure 7 shows the real frequency grid density $1/\Delta\omega$ as a function of ω. Note how the grid is dense in the center and gets more sparse as $|\omega|$ increases. Finally, figure 8 shows the default model, a gaussian with the same first and second moment as the spectrum.

Once all the figures are closed, take some time to look also at the information displayed in the terminal after the preprocessing. In particular, the numbers of real frequencies in

the grid and Matsubara frequencies used in χ^2 are the most important parameter for the calculation time. Also, the boundaries of main spectral range and the frequency step at the grid origin determine how well the spectrum will be resolved. It is a good habit to look at those parameters after every preprocessing to get some idea of the calculation time for a given grid. Those parameters can be modified with section FREQUENCY GRID PARAMETERS.

Now, to run the program down to the end, erase the *yes* on the line "preprocess only" in *OmegaMaxEnt_input_params.dat*, save it and continue execution by pressing ENTER. The figures should be displayed after less than a minute.

The active figure at the end of the execution should have title "Spectrum at optimal α". Three spectra, at three different values of α are shown on that figure along with the default model and the reference spectrum, which here is the exact one. The spectra at the three values of α help you evaluate if there is convergence of the spectrum. The closer they are to one another, the more reliable the result should be. In the present case, you can see the very small difference between those curves only by zooming on a small portion of the spectrum. This suggest a very accurate result and, indeed, the curves are also very close to the exact spectrum.

Look at the figure titled "$\log_{10}(\chi^2)$ vs $\log_{10}(\alpha)$". Note the three regions of different behaviors in that curve. What do they correspond to?[1] The next figure shows the curvature in $\log \chi^2$ as a function of $\gamma \log \alpha$, where γ is set to 0.2 by default, plotted as a function $\log_{10}(\alpha)$. The optimal α, say α^*, is chosen at the maximum of that curvature. It is given, along with the value of χ^2 on the last few lines of the terminal. There is also a figure showing the relative entropy S as a function of $\log_{10}(\alpha)$. Note where the optimal α is located on that curve.

On the figure titled "Spectrum at lowest α" you can see how the spectrum becomes distorted when α is below the optimal value. Why is that so? [2]

The figures titled "normalized deviation on $Re[G]$ at optimal α" and "autocorrelation of

[1] high α: default model regime, steep region: information-fitting, small α: noise-fitting

[2] It is distorted because some of the noise in G_{in} has been fitted.

$\Delta\tilde{G}_{Re}$ at optimal α" show the normalized distance $\Delta\tilde{G}_{Re} = Re[G_{in}(i\omega_n) - G_{out}(i\omega_n)]/\sigma_n^{Re}$, where σ_n^{Re} is the standard deviation of $Re[G(i\omega_n)]$, and its autocorrelation, respectively, at the optimal α. The next figures show the same curves for the imaginary part. Why do those curves look as such and have those particular amplitude at α^*? [3]

The figure titled "spectrum at sample frequencies" shows the spectrum at a few frequencies (11 by default) in the main spectral region (within a standard deviation around M_1) as a function of $\log_{10}(\alpha)$. This figure tells you to what extent the result can be trusted. A clear plateau at a given frequency suggest quantitative accuracy of the spectrum around that frequency.

Other figures show the normalized deviations of $Re[G]$ and $Im[G]$ and their autocorrelations at the lowest α. Note how they look almost identical to those at α^*. The changes in those curves indeed mostly happen above α^* as you will see in the next example. Finally, the normalized deviations of moments at optimal and lowest α are also shown.

Close all the figures and enter any letter other than 'y' in the terminal to end execution.

4 Example 2: real bosonic Matsubara frequency data

Erase the file *OmegaMaxEnt_input_params.dat* and execute *OmegaMaxEnt* to create a blank version.

For bosonic data, parameter "bosonic data" in section DATA PARAMETERS must be set to *yes*. Then, set "input directory" to *tutorial_examples/example_2* and "data file" and "error file" to the file names starting with *chi_test_diag_err* and *error_chi_test_err*, respectively, in that directory. The data $\chi(i\omega_n)$ for that example is real ($Im[\chi(i\omega_n)] = 0$), which means that $A(\omega) = -2Im[\chi(\omega)]$ is odd and the "spectrum" $A(\omega)/\omega$ computed by the program will be even. In that case, you can tell the program to take only the real part into account and use only the part of the spectrum for $\omega \geq 0$ by setting parameter

[3] If the provided error is accurate, the normalized errors at α^* should look like noise with a standard deviation of 1 and the autocorrelations should look like noisy Kronecker deltas.

"Im(G) column in data file" to 0 (or < 0) in section INPUT FILES PARAMETERS. Also, set parameter "reference spectral function file" in section DISPLAY OPTIONS to the file name starting with *spectr_chi* in the example 2 directory.

After execution, as in example 1, you can verify on figure "spectrum at optimal α" that the result is close to the exact spectrum ("reference spectrum"), look at the $\Delta\tilde{G}$ curves and their autocorrelations at optimal α and see where the optimal α is located on the functions plotted as a function of $\log_{10}(\alpha)$.

You have seen what the different functions plotted look like at the end of the execution. But it is useful to look at intermediate results to understand better the process. To do so, close all the figures, set the parameter "number of values of alpha computed in one execution" to 100 and "initialize maxent" to *yes* in section MINIMIZATION EXECUTION OPTIONS, and continue execution simply by pressing ENTER in the terminal. Note that the program might display some results labeled "at optimal α" even though the optimal α has not been reached (this is clear from $\log_{10}(\chi^2)$ vs. $\log_{10}(\alpha)$). If so, simply ignore those figures. Look at the spectrum at the lowest α and the normalized deviation of $Re[G]$ and its autocorrelation at the lowest α. Notice the order of magnitude of the those functions, how structured and smooth they are. Now, erase the *yes* at "initialize maxent", close the figures, and press ENTER to continue execution. The calculation will continue for another 100 values of α. You can repeat and follow the results as α decreases until it is below the optimal value, which happens when the small slope region appears at low α in $\log_{10}(\chi^2)$ vs. $\log_{10}(\alpha)$ and a large positive peak appears at low α in the curvature of $\log_{10}(\chi^2)$ vs. $\gamma \log_{10}(\alpha)$. Look at the "spectrum at optimal α" and verify that "normalized deviation of $Re[G]$ at optimal α" looks like noise and its autocorrelation is a noisy Kronecker delta. Also look where the optimal α (given in the terminal) is located on the curves in "spectrum at sample frequencies".

Close all the figures and enter any letter other than 'y' in the terminal to end execution.

5 Example 3: fermionic imaginary time data

Erase the file *OmegaMaxEnt_input_params.dat* and execute *OmegaMaxEnt* to create a blank version.

This example is for a fermionic Green function given in imaginary time. For that case, parameter "imaginary time data" in section DATA PARAMETERS must be set to *yes*. Then, set the "input directory" to *tutorial_examples/example_3* and the names of the "data file" and "error file" with those starting with *Gtau_test_diag_err* and *error_Gtau_err*, respectively, in that directory. Finally, set the parameter "reference spectral function file" to the file name starting with *test_spectr_func* in the *example_3* directory.

Set the parameter "use non uniform grid in main spectral range" to *yes* in section FREQUENCY GRID PARAMETERS, "display preprocessing figures" to *yes* in section PREPROCESSING EXECUTION OPTIONS and "number of values of alpha computed in one execution" to 200 in section MINIMIZATION EXECUTION OPTIONS. When you launch the program, after closing the figures displaying $G(\tau)$ and its error, the program will take slightly less than a minute to compute the Fourier transforms of those functions. Then, close the figures until "frequency grid density" is displayed, and zoom in on the grid density in the central part. What you see is the type of grid defined when "use non uniform grid in main spectral range" is enabled and if the program detects a sharp peak around $\omega = 0$ (the peak width and weight are printed in the terminal). Note that you can also define yourself a similar grid with parameters "use parameterized real frequency grid" and "grid parameters" (see the user guide for more details). Now, close the remaining figures until the actual calculation starts.

After execution, look at the figure "normalized deviation of G at lowest α" (ignore the figures labeled "at optimal α"). Here, although $\Delta\tilde{G}$ is not a smooth function, the oscillations are not noise, as can be seen from "autocorrelation of $\Delta\tilde{G}$ at lowest α", which is not a noisy Kronecker. Here, the covariance is diagonal for $G(\tau)$, and thus it is not diagonal for $G(i\omega_n) = K_n A$, the starting quantity in the calculation. Therefore, the transformed G in expression (1.10) do not depend on ω_n but on a covariance matrix eigenvector index,

which is ordered more or less arbitrary. That is why $\Delta\tilde{G}$ is not smooth. In general, the real and imaginary parts of $G(i\omega_n)$ are also mixed by the transformation (1.10) when the covariance is not diagonal in ω_n, which is why there are no separate curves for G_{Re} and G_{Im}.

Now, to see the final result, close the figures, erase the value at "number of values of alpha computed in one execution" and press ENTER in the terminal to continue execution. When the figures are displayed (this will take close to a minute), as in the previous examples, see on "spectrum at optimal α" how close the optimal spectrum is to the exact one (reference spectrum) and verify that "normalized deviation of G at optimal α" and "autocorrelation of $\Delta\tilde{G}$ at optimal α" have the correct properties. Then, look where the optimal value of α is located on the curves plotted as functions of $\log_{10}(\alpha)$. Finally, take a look at "spectrum at the lowest α" to see how the spectrum looks like when α is too small.

6 Output files

By default, the program puts all the saved output files in directory *OmegaMaxEnt_files* and the final results in directory *OmegaMaxEnt_final_result*. Those directories are created in the directory given at parameter "input directory" in section INPUT FILES PARAMETERS, or the current directory if that parameter is left blank. For the result, you can use a spectrum given in one of the two files starting with *optimal_spectral_function* in directory *OmegaMaxEnt_final_result*, but you can also obtain the spectrum at other values of α around the optimal value in directory *OmegaMaxEnt_files*. The user guide provides more information on how to control what results are saved and where.

7 Other functionalities

The program offers several other options. In particular, section FREQUENCY GRID PARAMETERS offers a few different ways of controlling the frequency grid, with different

levels of complexity and automatization. Those options are described in details in the user guide.

8 Data from actual simulations

In this tutorial, we only considered data generated from a known spectrum. The examples are ideal cases where the covariance matrix is known exactly, and for which the algorithms work perfectly. This is the best way to understand how the program works. With data generated in actual simulations, the covariance matrix can only be estimated and the errors are sometimes very large. This complicates the analysis and reduce the quality of the results. Less information contained in the data necessarily means less converged results. However, the better you understand the approach used by the program, the better you can extract the information in your data. Do not hesitate to try different options to try to improve the results, and take advantage of all the information provided by the program. A lot of discussion is also given in the user guide to help you analyse and improve your results.

References

[1] `http://www.physique.usherbrooke.ca/MaxEnt/index.php/Main_Page` or `https://github.com/amstremblay/OmegaMaxEnt`, 2016. Program's source code, binaries, and user guide.

[2] D. Bergeron and A.-M.S. Tremblay. Algorithms for optimized maximum entropy and diagnostic tools for analytic continuation. `http://arxiv.org/abs/1507.01012`, 2015.

[3] Dominic Bergeron and A.-M. S. Tremblay. Algorithms for optimized maximum entropy and diagnostic tools for analytic continuation. *Phys. Rev. E*, 94:023303, Aug 2016.