Ω*MaxEnt* tutorial

Dominic Bergeron[*]

(Dated: November 7, 2019)

CONTENTS

[*] dominic.bergeron@usherbrooke.ca

I. 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

$$G(i\omega_n) = \int_{-\infty}^{\infty} d\omega \, \frac{A(\omega)}{i\omega_n - \omega} \ . \tag{I.1}$$

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

$$G(\tau) = -\int d\omega \frac{e^{-\omega\tau} A(\omega)}{1 \pm e^{-\beta\omega}} \ , \tag{I.2}$$

is that, for a piecewise polynomial approximation of $A(\omega)$, (I.1) can be integrated analytically in each interval, which is not the case of (I.2). This yields better accuracy and better scaling with temperature.

Now, one could try to invert (I.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

$$Q = \chi^2 - \alpha S \,, \tag{I.3}$$

whith

$$\chi^2 = \sum_{mn} \left(G_m - K_m A\right)^T C_{mn}^{-1} \left(G_n - K_n A\right) \,, \tag{I.4}$$

where A is the vector obtained after discretizing ω, K_n is a row vector such that $K_n A$ is an approximation to (I.1), and C is the covariance matrix, which, for stochastic calculations, can be approximated by

$$C_{mn}^{\mu\nu} = \frac{1}{N-1} \sum_{i=1}^{N} (G_{im}^{\mu} - \bar{G}_m^{\mu})(G_{in}^{\nu} - \bar{G}_n^{\nu}) \,, \tag{I.5}$$

where

$$\bar{G}_m^{\mu} = \frac{1}{N} \sum_{i=1}^{N} G_{im}^{\mu} \,, \tag{I.6}$$

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 (I.3) is a differential (or relative) entropy, defined as

$$S = -\int d\omega A(\omega) \ln \frac{A(\omega)}{D(\omega)} \,, \tag{I.7}$$

where $D(\omega)$ is called the *default model*, and α is a parameter to be determined. The solution that minimize (I.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 (I.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,

$$\tilde{\mathbf{C}} = \mathbf{U}^{\dagger}\mathbf{C}\mathbf{U}\,, \tag{I.8}$$

χ^2 can be written

$$\chi^2 = (\tilde{G} - \tilde{\mathbf{K}}A)^T(\tilde{G} - \tilde{\mathbf{K}}A)\,, \tag{I.9}$$

where

$$\begin{aligned} \tilde{G} &= \sqrt{\tilde{\mathbf{C}}^{-1}}\mathbf{U}^{\dagger}G\,, \\ \tilde{\mathbf{K}} &= \sqrt{\tilde{\mathbf{C}}^{-1}}\mathbf{U}^{\dagger}\mathbf{K}\,. \end{aligned} \tag{I.10}$$

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. [1] or [2]. The latest version's source code and binaries are available on the program's web page.[3]

This tutorial only covers the most basic functionalities of ΩMaxEnt. Detailed descriptions of all its functionalities can be found in the user guide.

II. 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 values of the parameters in the latter file is optimal for most cases however and you will not modify that file in this tutorial.

It is important not to modify the parameter names in the parameter files, namely, any character up to ':' on a line. If you modify a parameter and this parameter is not printed in the terminal 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.

III. EXAMPLE 1: FERMIONIC MATSUBARA FREQUENCY DATA

Open a terminal window and set the directory to *tutorial_examples/example_1*. If a file *OmegaMaxEnt_input_params.dat* exists in that directory, remove it. Execute *OmegaMaxEnt* to create a blank version of that file and open it in a text editor. Then, copy the name of the file starting with *G_err* after "data file:" on the first line, and the name of the file starting with *error_G* after "error file:" in section INPUT FILES PARAMETERS. Also, set parameter "compute Pade result" in section COMPUTATION OPTIONS to *yes*. Finally, in section OPTIONAL MINIMIZATION TIME PARAMETERS/DISPLAY OPTIONS (last section of *OmegaMaxEnt_input_params.dat*), set parameter "reference spectral function file" to the file name starting with *spectr_w*, which contains the exact spectrum. The latter will be plotted along with the MaxEnt and Padé results at the end of the calculation.

Before computing the spectrum, the program performs a few preprocessing operations. To learn about those operations, set the parameters "preprocess only" and "display preprocessing figures" to *yes* in section PREPROCESSING EXECUTION OPTIONS, save *OmegaMaxEnt_input_params.dat* and execute *OmegaMaxEnt*. If you are in a hurry to see the results, simply execute *OmegaMaxEnt* without that option. Figures showing the results will be displayed after a few seconds of computation. In that case you can skip the next two paragraphs.

Before looking at the preprocessing figures, note that the parameters modified in the parameter files are printed in the terminal, followed by informations on the data and the preprocessing steps. When "display preprocessing figures" is enabled, figure 1 shows the real and imaginary parts of the data. Close the figure to continue execution. Figure 2 shows the errors for $Re[G]$ and $Im[G]$. Then, after closing figure 2, figures 3 to 6 show the norm of the spectrum and its first, second and third moments, respectively, computed by fits of a high Matsubara frequency expansion of Eq. (I.1) to the data, as a function of the initial frequency of the fit. Those curves are used to find the onset Matsubara frequency of the asymptotic regime, corresponding to the plateaus in those curves. That part of the data is then removed and replaced by constraints on the moments. The latter are also used to determine where the spectrum is centered and its standard deviation, in order to define a real frequency grid and a default model well adapted to the spectrum. After closing those figures, 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. Then, figure 8 shows the default model, which is simply a gaussian with the same first and second moment as the spectrum. Finally, figures 9 and 10 show the Padé results if it has been computed.

Once all the figures are closed, you can 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 parameters 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. For simple spectra, the automated grid definition is usually sufficient, but for complex spectra, it can be necessary to adjust the grid with the parameters of section FREQUENCY GRID PARAMETERS. In that case, you could do it at this stage, save the parameter file and, once you resumed execution by pressing ENTER in the terminal, the preprocessing would start over using the new parameters provided. Now, to run the program down to the end, erase the *yes* at “preprocess only” in *OmegaMaxEnt_input_params.dat*, save the file and press ENTER in the terminal. The figures showing the final result will be displayed after a few seconds.

After the computation, the active figure should have the title “Spectrum at optimal α” and show three spectra at three different values of α, along with the default model, the reference spectrum (the exact one in the present case), and the Padé spectrum. 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 quite close to the exact spectrum. Note how the Padé result is very accurate at low frequency, and much less at high frequency. This is often the case with Padé results, but not always, as this approach is highly sensitive to the noise in the data. The MaxEnt results on the other hand are much better on average over the whole frequency range and much less sensitive to noise, but are often less accurate than Padé at low frequency when the errors are small.

Now, if you look at the figure titled "$\log_{10}(\chi^2)$ vs $\log_{10}(\alpha)$", you will notice three regions of different behaviors in the curve. Can-you guess what they correspond to, knowing that expression is minimal at a given α?[4] 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, α_{opt}, is chosen at the maximum of that curvature. It appears, along with the corresponding value of χ^2, in the last few lines printed in the terminal. There is also a figure showing the relative entropy S as a function of $\log_{10}(\alpha)$. Note the inflexion point around α_{opt} on that curve.

On the figure titled "Spectrum at lowest α" you can see how distorted the spectrum is when α is below α_{opt}. Why is that so? [5]

The figures titled "normalized deviation on $Re[G]$ at optimal α" and "autocorrelation of $\Delta\tilde{G}_{Re}$ 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 α_{opt}. The next figures show the same curves for the imaginary part. Can you explain why those curves look that way, with those particular amplitude? [6]

The figure titled "spectrum at sample frequencies" shows the spectrum at a few frequencies (11 by default) in the main spectral region (a few standard deviation wide 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_{opt}. The changes in those curves indeed mostly happen above α_{opt} 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 (try using right mouse button on the Python icon to quit Python and close all figures at once) and enter any letter other than 'y' in the terminal to end execution.

IV. EXAMPLE 2: REAL BOSONIC MATSUBARA FREQUENCY DATA

Set the terminal directory to *tutorial_examples/example_2*. If a file *OmegaMaxEnt_input_params.dat* exists in that directory, remove it. Execute *OmegaMaxEnt* and open *OmegaMaxEnt_input_params.dat*. For bosonic data, parameter "bosonic data" in section DATA PARAMETERS must be set to *yes*. Set parameters "data file" and "error file" to the file names starting with *chi_err* and *error_chi*, respectively. 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 "Im(G) column in data file" to 0 (or < 0) in section INPUT FILES PARAMETERS. Also, set parameters "compute Pade result" in section COMPUTATION OPTIONS to *yes* and "reference spectral function file" in section DISPLAY OPTIONS to the file name starting with *spectr_chi* in the *example_2* directory.

After executing *OmegaMaxEnt*, 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 diagnostic functions look like at the end of the execution, but it is also useful to look at intermediate results to better understand the process. To do so, close all the figures, set the parameter "number of values of alpha computed in one execution" to 50 and "initialize maxent" to *yes* in section MINIMIZATION EXECUTION OPTIONS, save and press ENTER in the terminal to resume execution. 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 latter two functions and their structure and smoothness. Note that, when interrupting the computation at an intermediate stage, the program could also display some results labeled "at optimal α", even though the actual optimal α has not been reached. If so, simply ignore those figures. Now, erase the *yes* at "initialize maxent", close the figures, and press ENTER to resume execution. The calculation will continue for another 50 values of α, after which you can look how the spectrum and the other curves have changed. Now you can erase the number provided on line “number of values of alpha computed in one execution”, save and resume execution to let the computation finish completely. Then, in the figure “spectrum at optimal α”, note how the Padé result is, again, very good close to $\omega = 0$, but not around the smaller peak at large frequency, while the MaxEnt result is very good on average, but not very stable at $\omega = 0$. Finally, you can verify that “normalized deviation of $Re[G]$ at optimal α” and its autocorrelation have the correct noisy behaviors previously discussed and take a look also at figure “spectrum at sample frequencies” to see where the optimal α (given in the terminal) is located on the curves.

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

V. EXAMPLE 3: FERMIONIC IMAGINARY TIME DATA

Set the terminal directory to *tutorial_examples/example_3* and execute *OmegaMaxEnt* to create the parameter file. Set parameter “imaginary time data” in section DATA PARAMETERS to *yes* and “data file” and “error file” to the file names starting with *G_tau_err* and *error_Gtau*, respectively. In addition, set the parameter “use non uniform grid in main spectral range” to *yes* in section FREQUENCY GRID PARAMETERS, “compute Pade result” to *yes* in section COMPUTATION OPTIONS, and “reference spectral function file” in section DISPLAY OPTIONS to the file name starting with *spectr_w*. Optionally, you can also set “display preprocessing figures” to *yes* in section PREPROCESSING EXECUTION OPTIONS to see the preprocessing figures, for instance, the grid density of the non-uniform grid.

When you launch the program, the preprocessing stage takes more time than for Matsubara frequency input data because the Fourier transforms (FT) that must be computed before the MaxEnt computation itself can begin, the most time-consuming one being the two-dimensional FT of the covariance matrix. Those FT need only to be performed once for a given set of data however, since their outputs are saved in directory *Fourier_transformed_data*, and can be used instead of the original imaginary time data if the preprocessing is restarted, for instance with different preprocessing parameters (see section 2.2.3 of the user guide for more details).

After execution, as in the previous example, you can verify that the optimal spectrum is close to the exact one, compare it with the Padé result, and verify that $\Delta\tilde{G}$ and its autocorrelations at optimal α have the correct behaviors, and that the optimal α is correctly located at the transition between the information-fitting and noise-fitting regimes in $\log_{10}(\chi^2)$ vs. $\log_{10}(\alpha)$ [4]. Look also at the other figures to find the other similarities with the previous examples.

VI. OUTPUT FILES AND FIGURES

The final results are saved in directory *OmegaMaxEnt_final_result*. The spectra are saved in the files starting with *optimal_spectral_function* and the complete real frequency Green function, including the real part, is saved in *real_frequency_Green_function.dat*. The user guide provides more information on how to control the output files. The figures displayed by the program can also be displayed again at any time after the computation by running the Python scripts with the name format *OmegaMaxEnt_figs_#.py* (with command "python OmegaMaxEnt_figs_#.py"). The final results are displayed by the script with the largest label.

VII. DATA FROM ACTUAL SIMULATIONS

In this tutorial, we only considered data generated from known spectra, for which the covariance matrices are known exactly. Those are the cases for which the algorithms are the most effective, and are therefore ideal to understand how the program works. With data generated in actual simulations however, the covariance matrix can only be estimated and the errors are sometimes very large, which complicates the analysis and reduce the quality of the results. The better you understand the approach used by the program and its functionalities, the better you can extract the information in your data. In practice, the best results are often obtained after a few trials with different choices of input parameters. A lot of discussion is given in the user guide to help you analyse and improve your results.

---

[1] D. Bergeron and A.-M. S. Tremblay, Algorithms for optimized maximum entropy and diagnostic tools for analytic continuation, Phys. Rev. E **94**, 023303 (2016).

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

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

[4] Flat region at high α: default model regime, steep region: information-fitting, small slope region at low α: noise-fitting.

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

[6] Since G_{out} fits the information in G_{in} but not the noise at α_{opt}, $G_{in}(i\omega_n) - G_{out}(i\omega_n)$ contains only noise and, if σ_n^{Re} and σ_n^{Im} are accurate, $Re[G_{in}(i\omega_n) - G_{out}(i\omega_n)]/\sigma_n^{Re}$ and $Im[G_{in}(i\omega_n) - G_{out}(i\omega_n)]/\sigma_n^{Im}$ are simply noise with a standard deviation of 1, and thus their autocorrelations are noisy Kronecker delta.