QMBF/MBF

Version 5.61

Stefan Meinel

June 7, 2013
(minor correction on June 27, 2023)

Contents

1 Introduction

QMBF is an application with a graphical user interface intended for correlated least-χ^2 fitting of Euclidean two-point and three-point functions in lattice field theory. QMBF allows both unconstrained fits and Bayesian fits with Gaussian priors for the fit parameters. The program comes with a wide range of pre-defined types of model functions and also a parser which can interpret custom functions entered as standard text. An accompanying command-line program called MBF is also available. MBF can read a previously saved QMBF session file and perform the corresponding fit / bootstrap without the need for a graphical environment.

2 Fit Algorithm

QMBF supports simultaneous fitting of an arbitrary number of real-valued model functions, which may share some common fitting parameters. Let the K model functions be

$$\{f^k(\mathbf{x},\, \mathbf{a})\}_{k=1,...,K} \tag{1}$$

with V variables $\mathbf{x} = (x^1, ..., x^V)$ and P fitting parameters $\mathbf{a} = (a_1, ..., a_P)$. Of course, it is not required that all K functions depend on all P parameters – each function can depend on a different subset.

We assume that there are N sets of M data points

$$y_n^k(\mathbf{x}_m) \equiv y(n,\, m,\, k) \tag{2}$$

with $n = 1...N$, $m = 1...M$, i.e. in a lattice QCD context the index n would label the gauge configuration, while m might label sites on the lattice (e.g. the sink time slice of a 2-point correlator).

The algorithm works as follows: First, the average

$$\overline{y}(m,\, k) = \frac{1}{N}\sum_{n=1}^{N} y(n,\, m,\, k) \tag{3}$$

and the data covariance matrix[1]

$$C_{(m,k),(m',k')} = \frac{1}{N(N-1)}\sum_{n=1}^{N}\left[y(n,\, m,\, k) - \overline{y}(m,\, k)\right]\left[y(n,\, m',\, k') - \overline{y}(m',\, k')\right] \tag{4}$$

are computed. The χ^2 function to be minimized is then defined as

$$\begin{aligned}\chi^2(\mathbf{a}) \;\;=&\;\; \sum_{(m,k),(m',k')} W_{(m,k),(m',k')}\left[\overline{y}(m,\, k) - f^k(\mathbf{x}_m, \mathbf{a})\right]\left[\overline{y}(m',\, k') - f^{k'}(\mathbf{x}_{m'}, \mathbf{a})\right] \\ &+\sum_{p=1}^{P}\frac{(a_p - A_p)^2}{\sigma_{A_p}^2},\end{aligned} \tag{5}$$

where, depending on the chosen settings, W is either equal to C^{-1} (full inverse of data covariance matrix C), or a truncated inverse of C (see Sec. 2.1). The second term in (5), which contains the parameter priors A_p and their widths σ_{A_p}, is only added if the user activates the Bayesian constraints. In addition to the terms shown in this section, QMBF also allows an arbitrary additional function of the paramaters to be added to the definition of χ^2 (see Sec. 4.5).

[1] The normalization $1/(N(N-1))$ is the default choice. See Sec. 4.4 for an option to change this.

QMBF uses the Levenberg-Marquardt method for minimization. It requires the derivatives of χ^2 with respect to the parameters:

$$\frac{\partial \chi^2}{\partial a_p} = -2\beta_p \tag{6}$$

$$\frac{\partial^2 \chi^2}{\partial a_p \partial a_{p'}} = 2\alpha_{p\,p'} \tag{7}$$

with

$$\beta_p(\mathbf{a}) = \sum_{(m,k),(m',k')} W_{(m,k),(m',k')} \frac{\partial f^k(\mathbf{x}_m, \mathbf{a})}{\partial a_p} \left[\overline{y}(m',\, k') - f^{k'}(\mathbf{x}_{m'}, \mathbf{a})\right] - \frac{a_p - A_p}{\sigma^2_{A_p}}, \tag{8}$$

$$\begin{aligned} \alpha_{p\,p'}(\mathbf{a}) &= \sum_{(m,k),(m',k')} W_{(m,k),(m',k')} \frac{\partial f^k(\mathbf{x}_m, \mathbf{a})}{\partial a_p} \frac{\partial f^{k'}(\mathbf{x}_{m'}, \mathbf{a})}{\partial a_{p'}} + \frac{\delta_{p\,p'}}{\sigma^2_{A_p}} \\ &\quad - \sum_{(m,k),(m',k')} W_{(m,k),(m',k')} \frac{\partial^2 f^k(\mathbf{x}_m, \mathbf{a})}{\partial a_p \partial a_{p'}} \left[\overline{y}(m',\, k') - f^{k'}(\mathbf{x}_{m'}, \mathbf{a})\right] \end{aligned} \tag{9}$$

With some $\lambda > 0$, a new matrix $\widetilde{\alpha}$ is defined by

$$\widetilde{\alpha}_{p\,p} = (1 + \lambda)\alpha_{p\,p} \tag{10}$$

$$\widetilde{\alpha}_{p\,p'} = \alpha_{p\,p'} \quad (p \neq p'). \tag{11}$$

With the starting parameters $\mathbf{a}$ and an initial value for λ, say $\lambda = 0.001$, the iteration proceeds as follows:

1. Solve

$$\sum_{p'=1}^{P} \widetilde{\alpha}_{p\,p'}(\mathbf{a})\; \delta a_{p'} = \beta_p(\mathbf{a}) \tag{12}$$

   for $\delta\mathbf{a}$

2. If $\chi^2(\mathbf{a} + \delta\mathbf{a}) \geq \chi^2(\mathbf{a})$, set $\lambda = b\lambda$ with $b \sim 10$, and go back to 1, otherwise set $\lambda = \lambda/b$, set $\mathbf{a} = \mathbf{a} + \delta\mathbf{a}$ and go back to 1.

The procedure is stopped on the first occasion of χ^2 *decreasing* by less than some very small amount, say 10^{-3}. Then, the parameter covariance matrix

$$[\alpha^{-1}(\mathbf{a})]_{p\,p'}. \tag{13}$$

is computed. The standard error estimate for parameter a_p is then given by $\sqrt{[\alpha^{-1}(\mathbf{a})]_{p\,p}}$.

Note that $\alpha_{p\,p'}(\mathbf{a})$ has a contribution with second derivatives of the model functions f^k, shown in (9), which is typically small for a good fit because of the factor $\left[\overline{y}(m',\, k') - f^{k'}(\mathbf{x}_{m'}, \mathbf{a})\right]$. QMBF allows the user to choose whether this term will be included by the fitter or replaced by zero (see Sec. 4.4). This choice can be made separately for the minimization process and for the final computation of the parameter covariance matrix.

2.1 Truncated inverse of data covariance matrix

2.1.1 Singular value decomposition

If the user chooses one of the "SVD" settings (see Sec. 4.4), the singular value decomposition of C is calculated,

$$C = U\,S\,V^T, \qquad S = \mathrm{diag}(\sigma_1, ..., \sigma_D) \qquad \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_D \geq 0, \qquad UU^T = VV^T = 1 \tag{14}$$

(here, $D = M \cdot K$), and the matrix W in (5) is defined as follows, with some $D' < D$:

$$W = (U\,S'\,V^T)^{-1} = V\,S'^{-1}\,U^T, \qquad S'^{-1} = \mathrm{diag}(\sigma_1^{-1}, ..., \sigma_{D'}^{-1}, \underbrace{0, ..., 0}_{D-D'}). \tag{15}$$

The number of singular values $D - D'$ which are eliminated is referred to as the "SVD cut". It is either specified directly by the user, or chosen based on the size of the singular values (see Sec. 4.4).

2.1.2 Diagonal approximation (uncorrelated fit)

QMBF also provides an option to ignore the off-diagonal entries of the data covariance matrix C. If the user chooses this setting, the matrix W used in (5) is taken to be

$$W = \mathrm{diag}(1/C_{11},\ 1/C_{22},\ ...,\ 1/C_{DD}). \tag{16}$$

This corresponds to an uncorrelated fit.

3 Data File Format

3.1 Text format

QMBF can read text ("ASCII") data files in a particular format, which contains a header in addition to the actual data. The files must have the entries as shown in Table 1, with the notation from section 2. The first 4 lines contain the integers K, V, M, and N in this order. Then follows a block which specifies the arguments $\mathbf{x}_m \in \mathbb{R}^V$. Finally, there are the N blocks with the data (e.g. for the N gauge configurations in lattice QCD).

3.2 Binary Format

A binary data file consists of a stream of 32 bit little endian floating point numbers. The storage order is similar to the text format, except that the indices are omitted in order to reduce the file size. That, is the file has the following content (all stored as floating point numbers, including K, V, M, N):

$$\begin{aligned}
&K,\ V,\ M,\ N,\quad x_1^1,\ x_1^2,\ ...,\ x_1^V,\quad x_2^1,\ x_2^2,\ ...,\ x_2^V,\ \ ...\ ,\ x_M^1,\ x_M^2,\ ...,\ x_M^V,\\
&y_1^1(\mathbf{x}_1),\ y_1^2(\mathbf{x}_1),\ ...,\ y_1^K(\mathbf{x}_1),\ \ y_1^1(\mathbf{x}_2),\ y_1^2(\mathbf{x}_2),\ ...,\ y_1^K(\mathbf{x}_2),\ \ ...\ ,\ y_1^1(\mathbf{x}_M),\ y_1^2(\mathbf{x}_M),\ ...,\ y_1^K(\mathbf{x}_M),\\
&y_2^1(\mathbf{x}_1),\ y_2^2(\mathbf{x}_1),\ ...,\ y_2^K(\mathbf{x}_1),\ \ y_2^1(\mathbf{x}_2),\ y_2^2(\mathbf{x}_2),\ ...,\ y_2^K(\mathbf{x}_2),\ \ ...\ ,\ y_2^1(\mathbf{x}_M),\ y_2^2(\mathbf{x}_M),\ ...,\ y_2^K(\mathbf{x}_M),\\
&\vdots\\
&y_N^1(\mathbf{x}_1),\ y_N^2(\mathbf{x}_1),\ ...,\ y_N^K(\mathbf{x}_1),\ \ y_N^1(\mathbf{x}_2),\ y_N^2(\mathbf{x}_2),\ ...,\ y_N^K(\mathbf{x}_2),\ \ ...\ ,\ y_N^1(\mathbf{x}_M),\ y_N^2(\mathbf{x}_M),\ ...,\ y_N^K(\mathbf{x}_M).
\end{aligned}$$

K					
V					
M					
N					
1	x_1^1	x_1^2	...	x_1^V	
2	x_2^1	x_2^2	...	x_2^V	
$\vdots$				$\vdots$	
M	x_M^1	x_M^2	...	x_M^V	
1	1	$y_1^1(\mathbf{x}_1)$	$y_1^2(\mathbf{x}_1)$	...	$y_1^K(\mathbf{x}_1)$
1	2	$y_1^1(\mathbf{x}_2)$	$y_1^2(\mathbf{x}_2)$	...	$y_1^K(\mathbf{x}_2)$
$\vdots$					$\vdots$
1	M	$y_1^1(\mathbf{x}_M)$	$y_1^2(\mathbf{x}_M)$	...	$y_1^K(\mathbf{x}_M)$
2	1	$y_2^1(\mathbf{x}_1)$	$y_2^2(\mathbf{x}_1)$	...	$y_2^K(\mathbf{x}_1)$
2	2	$y_2^1(\mathbf{x}_2)$	$y_2^2(\mathbf{x}_2)$	...	$y_2^K(\mathbf{x}_2)$
$\vdots$					$\vdots$
2	M	$y_2^1(\mathbf{x}_M)$	$y_2^2(\mathbf{x}_M)$	...	$y_2^K(\mathbf{x}_M)$
$\vdots$					
N	1	$y_N^1(\mathbf{x}_1)$	$y_N^2(\mathbf{x}_1)$	...	$y_N^K(\mathbf{x}_1)$
N	2	$y_N^1(\mathbf{x}_2)$	$y_N^2(\mathbf{x}_2)$	...	$y_N^K(\mathbf{x}_2)$
$\vdots$					$\vdots$
N	M	$y_N^1(\mathbf{x}_M)$	$y_N^2(\mathbf{x}_M)$	...	$y_N^K(\mathbf{x}_M)$

Table 1: QMBF “Text” data file format

4 QMBF

4.1 Compiling QMBF

4.1.1 Software Requirements

- For OS X, the necessary developer tools can be obtained for example by installing Apple Xcode.
- Qt version 4.x. On most Linux distributions, this is already installed or available through the package management. Note that the *development* packages are also needed (these usually have `-dev` or `-devel` in the package name). Free open source editions for Linux/X11, OS X and Windows are also available from http://qt-project.org/
- GNU Scientific Library, version ≥ 1.13 (see http://www.gnu.org/software/gsl/). On most linux distributions this is available through the package management. Note that the *development* packages are also needed (these usually have `-dev` or `-devel` in the package name). For OS X, installation via http://www.macports.org/ is recommended.
- For built-in plotting functionality: Grace (see http://plasma-gate.weizmann.ac.il/Grace/). On most linux distributions this is available through the package management. For OS X, installation via http://www.macports.org/ is recommended.

4.1.2 Adjusting `QMBF.pro` (if necessary)

This step is only necessary if libraries such as the GNU Scientific Library are installed in unusual locations (in the following example, `/path/to/mylib` and `/path/to/myinclude`). In this case, go to the directory with the QMBF source code, and add the following lines at the end of the file `QMBF.pro`, for each library where needed:

```
LIBS += -L/path/to/mylib
QMAKE_CXXFLAGS += -I/path/to/myinclude
```

4.1.3 Compiling on Linux

From the directory with the QMBF source code, run

```
qmake
```

(the `qmake` program is part of Qt). This generates a Makefile suitable for the system. Then run

```
make
```

This will produce an executable file called `QMBF`.

4.1.4 Compiling on OS X

From the directory with the QMBF source code, run

```
qmake -spec macx-g++
```

(the `qmake` program is part of Qt). This generates a Makefile suitable for the system. Then run

```
make
```

This will produce an application folder called `QMBF.app`.

4.2 Running QMBF

QMBF takes as optional command-line argument the name of a mbf file to be opened (see Sec. 4.6.1 for information on QMBF session files):

4.2.1 Running on Linux

```
QMBF [inputfile.mbf]
```

4.2.2 Running on OS X

```
open QMBF.app [inputfile.mbf]
```

Alternatively, after .mbf files have been associated with QMBF, use

```
open inputfile.mbf
```

4.3 Model Functions, Variables, and All That

4.3.1 User-defined Models

QMBF allows the user to enter model functions as text. To activate the user-defined mode, select "User-defined Model" in the "Model" menu. Parameters, constants and variables are alpha-numeric strings, must not contain spaces, and must not begin with a number. However, the underscore "_" may be used. The Functions may contain the following built-in operations:

```
+, -, *, /,
(, ),
exp(...), log(...),
sin(...), cos(...), tan(...),
sinh(...), cosh(...), tanh(...),
arcsin(...), arccos(...), arctan(...),
sqr(...), sqrt(...),
alt(...).
```

The function alt, which alternates between -1 and +1, is defined as follows:

$$\texttt{alt}(x) = (-1)^{\mathrm{int}(x)}$$

where $\mathrm{int}(x)$ gives the integer part of x (as in C++). Because $\texttt{alt}(x)$ is not differentiable, x should not contain fit parameters (only variables and constants).

In the Tab "Model Functions" the user can select the number of functions K, and enter the functions (as strings) themselves. The variable names and their respective fitting ranges must be entered in the Tab titled "Variables / Fitting Ranges". The total number of fit parameters (P) is selected in the "Parameters/Derivatives" Tab of QMBF. For each parameter, the parameter name must be entered. If numerical differentiation for the first-order derivatives is disabled (see Sec. 4.4), additionally the first-order derivatives of all functions with respect to that parameter must be entered, expressed using the same available built-in operations as above (second-order derivatives, if needed, will always be calculated numerically).

Also selected in the “Parameters/Derivatives” Tab is the number $P' \leq P$ of parameters to be subtracted from the number of degrees of freedom $(M \cdot K)$ when calculating the χ^2 per effective degrees of freedom[2]

$$\frac{\chi^2}{\mathrm{dof}} = \frac{\chi^2}{M \cdot K - P'}.$$

For non-Bayesian fits, the default value is $P' = P$, while for Bayesian fits, the default value is $P' = 0$.

Optional constants and their values for use in the functions and derivatives may be defined in the Tab “Constants”.

4.3.2 Predefined Models

Alternatively, the user can choose among commonly used predefined types of model functions via the other entries in the menu “Model”. For the predefined models, the derivatives are implemented internally and will not be shown in the “Parameters / Derivatives” Tab. The internal implementation of the derivatives is exact up to machine precision. If the “Numerical differentiation” option (see Sec. 4.4) is enabled, the built-in implementations of the derivatives are not used, and the derivatives are approximated using discrete differences of the model functions instead.

The complete list of predefined model types can be found in appendix A.

4.4 Fit Settings Tab

The Tab “Fit Settings” contains input fields that control the fit algorithm. These settings apply for all types of fit models.

The user can activate or deactivate the Gaussian priors by clicking on “Activate Bayesian constraints”. For the option “Use Gaussian random priors for bootstrap / multifit”, see sections 4.6.6 and 4.6.7.

When “Use numerical differentiation for first derivatives” is activated, the first-order derivatives in Eqs. (8) and (9) are approximated using a symmetric difference operation, using

$$\frac{\partial f^k(\mathbf{x}_m; a_1, ..., a_P)}{\partial a_p} \approx \frac{f^k(..., a_p + h, ...) - f^k(..., a_p - h, ...)}{2h}. \tag{17}$$

The option “Use numerical differentiation for first derivatives” is intended mainly for the user-defined models (Sec. 4.3.1), to avoid the need of entering the symbolic derivatives by hand. The predefined fit models (Appendix A) have built-in symbolic implementations of the first-order derivatives, and for those it is normally recommended to disable this option.

The step size h used in (17) can also be specified by the user in the input field “Numerical differentiation step size”. Note that making the steps size too large may give a poor approximation of the derivative, while making it too small may cause significant round-off errors. Because QMBF uses double-precision floating-point arithmetic, the default value $h = 10^{-8}$ is thought to be a good compromise between the two possible causes of error. To be safe, the user should check that the fit results are independent of the step size to the desired precision (or compare the results to the case where numerical differentiation is disabled and the symbolic derivatived are entered by hand in the “Parameters/Derivatives Tab”).

The options “Use second derivatives for parameter covariance matrix” and “Use second derivatives for minimization” control whether the term with the second derivatives of the model functions f^k, shown in (9), is included in the calculation of $\alpha_{p\,p'}$ or is replaced by zero. The first option only affects the

[2] If an SVD cut is used, $M \cdot K$ is replaced by D', as defined in Sec. 2.1.1.

final calculation of $\alpha_{p\,p'}$ after the minimum of χ^2 was found. Hence, it affects the parameter covariance matrix $[\alpha^{-1}]_{p\,p'}$, but not the central values of the fitted parameters. It is recommended that this option be enabled in order to get the most accurate estimate of the parameter covariance matrix. Conversely, the second option only affects the intermediate calculations of $\widetilde{\alpha}_{p\,p'}$ during the search for the minimum of χ^2. It is usually recommended that this option be disabled, because it may destabilize the iterations.

The second derivatives of the model functions, if enabled, will always be computed numerically. If the option "Use numerical differentiation for first derivatives" is *disabled*, the second derivatives will be calculated using

$$\frac{\partial^2 f^k(\mathbf{x}_m; a_1, ..., a_P)}{\partial a_p\, \partial a_q} \approx \frac{\partial_{a_q} f^k(..., a_p + h, ...) - \partial_{a_q} f^k(..., a_p - h, ...)}{2h}, \tag{18}$$

where $\partial_{a_q} f^k$ are the symbolic first-order derivatives (either from the built-in models, or entered by the user for a user-defined model).

On the other hand, if the option "Use numerical differentiation for first derivatives" is *enabled*, the second derivatives will be calculated using

$$\begin{aligned}\frac{\partial^2 f^k(\mathbf{x}_m; a_1, ..., a_P)}{\partial a_p\, \partial a_q} \quad &\approx \quad \frac{1}{4h'^2}\Bigg[f^k(..., a_p + h', ..., a_q + h', ...) - f^k(..., a_p + h', ..., a_q - h', ...) \\ &\qquad - f^k(..., a_p - h', ..., a_q + h', ...) + f^k(..., a_p - h', ..., a_q - h', ...) \Bigg],\end{aligned} \tag{19}$$

where the new step size $h' = \sqrt{h}$ is used to minimize round-off errors. Note that (18) is numerically more precise than (19).

The Fit Settings tab also contains input fields for the starting value for λ, the lambda factor b, the tolerance for decreases in χ^2 at which the iteration stops, and the maximum number of fitting iterations. See Sec. 2 for the meaning of these parameters.

The selector "Normalization of covariance matrix" specifies whether the data covariance matrix C in Eq. (4) is defined with the usual factor of

$$\frac{1}{N(N-1)},$$

(as shown in (4)) or alternatively with the factor

$$\frac{1}{N-1}.$$

The latter is needed to get the correct error estimates for fit parameters for the case that the original data file was created using bootstrap over data sets (e.g. in the calculation of ratios of three-point and two-point functions).

The selector "(Pseudo-)Inversion method for data covariance matrix" controls the definition of the matrix W in Eq. (5). The options are:

- "LU decomposition": full inversion, $W = C^{-1}$
- "SVD with fixed cut": truncated inversion (see Sec. 2.1.1), remove given number $D - D'$ of smallest eigenvalues

- “SVD with EV ratio cut”: truncated inversion (see Sec. 2.1.1), remove eigenvalues that are smaller than the specified fraction of the largest eigenvalue σ_1
- “SVD with EV value cut”: truncated inversion (see Sec. 2.1.1), remove eigenvalues smaller than some given value
- “Diagonal only (uncorrelated fit)”: keep only diagonal elements in data covariance matrix (see Sec. 2.1.2)

The option “Restrict data range” lets the user choose a restricted range for the data index n (cf. Sec. 2), allowing the analysis of only a subset of the data.

Additionally, the user can set a bin size, let us call it B, to reduce autocovariances. This means that before the steps from section 2 are applied, the data sets are grouped bins of size B, and within each bin the average is taken. If N (or what is left after restricting the data range) is not an integer multiple of B, the remaining data sets which would not completely fill a bin are ignored.

Finally, the user can enter the number of samples to be created for the bootstrap procedure, or select/generate an external file (“bootstrap ensemble file”) containing the information needed for the random sampling of configurations. See Sec. 4.6.6 for more information on this.

4.5 Additional term in chi sqr

The Tab “Additional term in chi^2” can be used to add an arbitrary function of the fit parameters to χ^2. This option is only intended for special applications, for example forcing two fit parameters to be close to each other by adding the square of their difference, divided by some width, to χ^2.

The elementary operations that can be used for defining such a function are the same as in Sec. 4.3.1. The function may contain any of the fit parameters defined in the Parameters/Derivatives Tab. However, the constants from the “Constants” tab can **not** be used here. Instead the “Additional term in chi^2” tab provides the option of defining new constants solely for use in this additional term in χ^2.

QMBF always computes the first and second derivatives of this additional function (as needed by the Levenberg-Marquardt algorithm) numerically. The step size used for this differentiation entered here is independent from that entered in the Fit Settings tab (Sec. 4.4).

4.6 The Main Tab

4.6.1 Opening and Saving `mbf` Files

Using the “File” Menu (or the toolbar), the user can open and save `.mbf` files, which contain all session settings of the graphical user interface, i.e. model functions, starting values, fitting ranges, plot settings etc.

4.6.2 Selecting a Data File

A data file name including the full path can be entered in the input field labelled “Data file”, or selected using a file open dialog by clicking the button next to the input field. The file will only be loaded when necessary for the fit.

4.6.3 Selecting an Output Directory

The output directory is used for plotting, for saving the fitted parameters and their covariance matrix, and for bootstrap.

4.6.4 Setting the Start Parameters, Priors and Prior Widths

The table in the middle of the main Tab allows the user to enter starting values for the fit parameters, and, if activated (see section 4.4), the central values of the Gaussian priors A_p and their widths σ_{A_p}.

4.6.5 Fitting

To start the fit, click on the botton "Start fit". While the fit is performed, the values of χ^2/dof and λ are shown after every iteration.

4.6.6 Bootstrap

When clicking on the button "Start bootstrap", QMBF will perform repeated fits for a selected number of "bootstrap samples". Each bootstrap sample is obtained by randomly choosing N out of the N configurations with allowed repetitions. QMBF will recompute and invert the data covariance matrix for every single bootstrap sample. QMBF will write the ensembles of fit results for each parameter into an individual file located at the specified output directory (cf. 4.6.3). The files are named after the current `.mbf` session file (see section 4.6.1) and the parameter names. When the bootstrap is completed, the bootstrap error estimates (based on the 68% range) of the fit results are shown next to the fit results.

By default, the random numbers of configurations are generated by QMBF just before doing the bootstrap. Alternatively, if the option "Use bootstrap ensemble file" is activated in the "Fit Settings" Tab, the numbers are read from a text file. The format of a bootstrap ensemble file is as follows: the number of bootstrap samples S, followed by the number of configurations N, followed by $S \cdot N$ random integer numbers in the range $1 ... N$. Such files can also be generated by QMBF, by clicking on the corresponding button in the "Fit Settings" Tab.

For Bayesian fitting, it is recommended to activate the option "Use Gaussian random priors for bootstrap / multifit" in the "Fit Settings" Tab, in order to get the (approximately) correct probability distribution for the fit parameters. If this is activated, in addition to randomly choosing data set ensembles, the priors will be choosen randomly from Gaussian distributions with the given prior widths and central values.

4.6.7 Multifit

The "Tools" menu contains an entry "Multifit". When clicking on this, QMBF will perform N successive fits, where the nth fit uses the following definition of χ^2:

$$\begin{aligned}\chi_n^2(\mathbf{a}) \;\; &= \sum_{(m,k),(m',k')} W_{(m,k),(m',k')} \left[y(n,\, m,\, k) - f^k(\mathbf{x}_m, \mathbf{a}) \right] \left[y(n,\, m',\, k') - f^{k'}(\mathbf{x}_{m'}, \mathbf{a}) \right] \\ &\quad + \sum_{p=1}^{P} \frac{(a_p - A_p)^2}{\sigma_{A_p}^2}. \end{aligned} \tag{20}$$

Compared to the standard definition of χ^2 in Eq. (5), here the average of the data, $\overline{y}(m,\, k)$, has been replaced by the nth data sample, $y(n,\, m,\, k)$. QMBF will write the N different fit results for each parameter into an individual file located in the specified output directory. The files are named after the current `.mbf` session file and the parameter names. When the "multifit" procedure is completed, the 68% widths of the fit parameter distributions will be calculated.

The "Multifit" option is useful when the data $y(n,\, m,\, k)$ themselves were obtained through a bootstrap procedure (where n is the nth bootstrap ensemble), and a corresponding resampling of the fit results is wanted. Note that the covariance matrix in Eq. (20) stays fixed and is computed as before from all data samples.

For Bayesian fitting, it is recommended to activate the option "Use Gaussian random priors for bootstrap / multifit" in the "Fit Settings" Tab, in order to get the (approximately) correct probability distribution for the fit parameters. If this is activated, the priors for each fit be choosen randomly from Gaussian distributions with the given prior widths and central values.

4.6.8 Other Tools

The "Tools" menu (and the corresponding buttons in the toolbar) provides some additional useful functionality:

- set the central values of the parameter priors equal to the start values
- set the central values of the parameter priors equal to the previous fit results
- set the parameter start values equal to the previous fit results
- compute the χ^2/dof for the current parameter start values
- reload the data file (use if the data file was modified by an external program while QMBF is running)
- show a report with the current fit settings (opens in new window and can be saved to a text file)
- write the fit results and the complete parameter covariance matrix to files in the output directory

4.7 Data and Function Plotting

The "Plot" menu (and the corresponding buttons in the toolbar) allow the automated execution of the plotting program "Grace" (see http://plasma-gate.weizmann.ac.il/Grace/) to generate two-dimensional plots of:

- the averaged data only
- the averaged data and the model function evaluated for the start values of the parameters
- the averaged data and the model function evaluated for the fitted values of the parameters
- an effective mass plot, showing the fitted ground state energy (for 2-point functions)

Invoking the plotting generates a Grace input file in the output directory, and executes Grace. Under "Plot → Options..." some settings for the plotting procedure can be entered.

5 MBF

5.1 Compiling MBF

Requirements:

- GNU Scientific Library, version $\geq$ 1.13 (see http://www.gnu.org/software/gsl/). Note that the *development* package is also needed (these usually have `-dev` or `-devel` in the package name).

A Makefile is supplied with the source code; the variables `INCPATH` and `LIBS` may require adjustment for the specific machine.

5.2 Using MBF

MBF is run from the terminal; it requires as command line argument the name of a complete .mbf session fileas generated by QMBF (below denoted as `inputfile`). The usage is as follows:

```
MBF [options] inputfile

Options:
  -b           perform bootstrap
```

MBF always reads the input file and performs a fit; the results are printed to stdout. Additionally the central values and complete covariance matrix of the resulting fit parameters are written to files created in the output directory specified in the .mbf file. With the option `-b`, MBF performs the bootstrap procedure and writes the results for each parameter into an individual file created in the output directory specified in the .mbf file.

A Predefined Model Functions

A.1 2-Point Correlator

A.1.1 Multiple Exponentials

- function(s):

$$f(\mathtt{t}) = \mathtt{A}\left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\, e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}}\right]$$

- variable(s): `t`
- parameter(s): `A`, $\{\mathtt{B\_}n\}$, `E`, $\{\mathtt{dE\_}n\}$

A.1.2 Multiple Exponentials with Exponential Energies

- function(s):

$$f(\mathtt{t}) = \mathtt{A}\left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\, e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+\ldots+e^{\mathtt{dE\_}n})\mathtt{t}}\right]$$

- variable(s), parameter(s): same as in Sec. A.1.1

A.1.3 Multiple Exponentials with Square Amplitudes

- function(s):

$$f(\mathtt{t}) = \mathtt{A}^2\left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} (\mathtt{B\_}n)^2\, e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}}\right]$$

- variable(s), parameter(s): same as in Sec. A.1.1

A.1.4 Multiple Exponentials with Square Amplitudes and Exponential Energies

- function(s):

$$f(\mathtt{t}) = \mathtt{A}^2 \left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1} (\mathtt{B\_}n)^2\, e^{-(e^{\mathtt{E}} + e^{\mathtt{dE\_1}} + \ldots + e^{\mathtt{dE\_}n})\mathtt{t}} \right]$$

- variable(s), parameter(s): same as in Sec. A.1.1

A.1.5 Multiple (incl. oscillating) Exponentials

- function(s):

$$\begin{aligned} f(\mathtt{t}) \;&=\; \mathtt{A} \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\, e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right] \\ &\quad +(-1)^{\mathtt{t}+1}\mathtt{Ao} \left[e^{-\mathtt{Eo}\ \mathtt{t}} + \sum_{m=1}^{M-1} \mathtt{Bo\_}m\, e^{-(\mathtt{Eo}+\mathtt{dEo\_1}+\ldots+\mathtt{dEo\_}m)\mathtt{t}} \right] \end{aligned}$$

- variable(s): $\mathtt{t}$
- parameter(s): $\mathtt{A}$, $\{\mathtt{B\_}n\}$, $\mathtt{E}$, $\{\mathtt{dE\_}n\}$, $\mathtt{Ao}$, $\{\mathtt{Bo\_}m\}$, $\mathtt{Eo}$, $\{\mathtt{dEo\_}m\}$

A.1.6 Multiple (incl. oscillating) Exponentials with Exponential Energies

- function(s):

$$\begin{aligned} f(\mathtt{t}) \;&=\; \mathtt{A} \left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\, e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+\ldots+e^{\mathtt{dE\_}n})\mathtt{t}} \right] \\ &\quad +(-1)^{\mathtt{t}+1}\mathtt{Ao} \left[e^{-e^{\mathtt{Eo}}\ \mathtt{t}} + \sum_{m=1}^{M-1} \mathtt{Bo\_}m\, e^{-(e^{\mathtt{Eo}}+e^{\mathtt{dEo\_1}}+\ldots+e^{\mathtt{dEo\_}m})\mathtt{t}} \right] \end{aligned}$$

- variable(s), parameter(s): same as in Sec. A.1.5

A.1.7 Multiple (incl. oscillating) Exponentials with Square Amplitudes

- function(s):

$$\begin{aligned} f(\mathtt{t}) \;&=\; \mathtt{A}^2 \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} (\mathtt{B\_}n)^2\, e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right] \\ &\quad +(-1)^{\mathtt{t}+1}\mathtt{Ao}^{\,2} \left[e^{-\mathtt{Eo}\ \mathtt{t}} + \sum_{m=1}^{M-1} (\mathtt{Bo\_}m)^2\, e^{-(\mathtt{Eo}+\mathtt{dEo\_1}+\ldots+\mathtt{dEo\_}m)\mathtt{t}} \right] \end{aligned}$$

- variable(s), parameter(s): same as in Sec. A.1.5

A.1.8 Multiple (incl. oscillating) Exponentials with Square Amplitudes and Exponential Energies

- function(s):

$$
\begin{aligned}
f(\mathtt{t}) \quad = \quad & \mathtt{A}^2 \left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1} (\mathtt{B\_}n)^2\, e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+\ldots+e^{\mathtt{dE\_}n})\mathtt{t}} \right] \\
& +(-1)^{\mathtt{t}+1}\mathtt{Ao}^2 \left[e^{-e^{\mathtt{Eo}}\ \mathtt{t}} + \sum_{m=1}^{M-1} (\mathtt{Bo\_}m)^2\, e^{-(e^{\mathtt{Eo}}+e^{\mathtt{dEo\_1}}+\ldots+e^{\mathtt{dEo\_}m})\mathtt{t}} \right]
\end{aligned}
$$

- variable(s), parameter(s): same as in Sec. A.1.5

A.2 2-Point Correlator, Vector Fit

All the "scalar" two-point models listed in section A.1 are also available as "vector" two-point models. These models require the dimension `dim` of the vector. A vector model has then `dim` functions $f_i(\mathtt{t})$ ($i = 1 \ldots \mathtt{dim}$) of the same form as the underlying scalar model. These functions have individual amplitude parameters, but share all the energy parameters. For example, the functions for `multi_-exp_vec_model` are

$$
f_i(\mathtt{t}) = \mathtt{A\_\_}i \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\, e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right]
$$

for $i = 1 \ldots \mathtt{dim}$.

A.3 Periodic B.C.

All the "scalar" and "vector" two-point models listed in A.1 and A.2 are also available with periodic boundary conditions. The models with periodic boundary conditions have the same parameters as the underlying models. The only difference is the replacement

$$
f_i(\mathtt{t}) \quad \to \quad f_i(\mathtt{t}) + f_i(\mathtt{T} - \mathtt{t})
$$

for all functions f_i of the model, where `T` is a constant.

A.4 Two-point models with time-independent contributions

For all the "scalar" and "vector" two-point models listed in A.1 and A.2, as well as their versions with periodic boundary conditions (Sec. A.3), an additional version exists, which adds time-independent pieces to the fit function:

$$
f(\mathtt{t}) \quad \to \quad f(\mathtt{t}) + \mathtt{C}
$$

for scalar models,

$$
f(\mathtt{t}) \quad \to \quad f(\mathtt{t}) + \mathtt{C} + (-1)^{\mathtt{t}+1}\mathtt{Co}
$$

for scalar models with oscillating contributions,

$$
f_i(\mathtt{t}) \quad \to \quad f_i(\mathtt{t}) + \mathtt{C\_\_}i
$$

for vector models, and

$$
f_i(\mathtt{t}) \quad \to \quad f_i(\mathtt{t}) + \mathtt{C\_\_}i + (-1)^{\mathtt{t}+1}\mathtt{Co\_\_}i
$$

for vector models with oscillating contributions. The quantities `C`, `Co`, $\{\mathtt{C\_\_}i\}$, $\{\mathtt{Co\_\_}i\}$ (as appropriate) are additional fit parameters.

A.5 2-Point Correlator, Matrix Fit

Matrix models are very different from vector models. In matrix models, it is assumed that the amplitudes factor into an outer product of a vector with itself, like $\mathtt{A\_\_}i\ \mathtt{A\_\_}j$, where the $\mathtt{A\_\_}i$ are used as fit parameters.

In the following, the functions are labelled by two indices i, j. The required storage order in the data files is such that the first index (i) runs slow and the second index (j) runs fast.

A.5.1 Multiple Exponentials

- function(s): for $i = 1 ... \mathtt{dim\_1}$, $j = 1 ... \mathtt{dim\_2}$:

$$f_{ij}(\mathtt{t}) = \mathtt{A\_\_}i\ \mathtt{A\_\_}j \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\ \mathtt{B\_}n\mathtt{\_\_}j\ e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right]$$

- variable(s): $\mathtt{t}$
- parameter(s): $\{\mathtt{A\_\_}i\}$, $\{\mathtt{B\_}n\mathtt{\_\_}i\}$ (for $i = 1 ... \max(\mathtt{dim\_1},\ \mathtt{dim\_2})$), $\mathtt{E}$, $\{\mathtt{dE\_}n\}$

A.5.2 Multiple Exponentials with Exponential Energies

- function(s): for $i = 1 ... \mathtt{dim\_1}$, $j = 1 ... \mathtt{dim\_2}$:

$$f_{ij}(\mathtt{t}) = \mathtt{A\_\_}i\ \mathtt{A\_\_}j \left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\ \mathtt{B\_}n\mathtt{\_\_}j\ e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+\ldots+e^{\mathtt{dE\_}n})\mathtt{t}} \right]$$

- variable(s), parameter(s): same as in Sec. A.5.1

A.5.3 Multiple (incl. oscillating) Exponentials

- function(s): for $i = 1 ... \mathtt{dim\_1}$, $j = 1 ... \mathtt{dim\_2}$:

$$\begin{aligned} f_{ij}(\mathtt{t}) \;&=\; \mathtt{A\_\_}i\ \mathtt{A\_\_}j \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\ \mathtt{B\_}n\mathtt{\_\_}j\ e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right] \\ &\quad +(-1)^{\mathtt{t}+1}\mathtt{Ao\_\_}i\ \mathtt{Ao\_\_}j \left[e^{-\mathtt{Eo}\ \mathtt{t}} + \sum_{m=1}^{M-1} \mathtt{Bo\_}m\mathtt{\_\_}i\ \mathtt{Bo\_}m\mathtt{\_\_}j\ e^{-(\mathtt{Eo}+\mathtt{dEo\_1}+\ldots+\mathtt{dEo\_}m)\mathtt{t}} \right] \end{aligned}$$

- variable(s): $\mathtt{t}$
- parameter(s):
  $\{\mathtt{A\_\_}i\}$, $\{\mathtt{B\_}n\mathtt{\_\_}i\}$ (for $i = 1 ... \max(\mathtt{dim\_1},\ \mathtt{dim\_2})$), $\mathtt{E}$, $\{\mathtt{dE\_}n\}$,
  $\{\mathtt{Ao\_\_}i\}$, $\{\mathtt{Bo\_}m\mathtt{\_\_}i\}$ (for $i = 1 ... \max(\mathtt{dim\_1},\ \mathtt{dim\_2})$), $\mathtt{Eo}$, $\{\mathtt{dEo\_}m\}$,

A.5.4 Multiple (incl. oscillating) Exponentials with Exponential Energies

- function(s): for $i = 1 ... \texttt{dim\_1}$, $j = 1 ... \texttt{dim\_2}$:

$$
\begin{aligned}
f_{ij}(\texttt{t}) \quad = \quad & \texttt{A\_\_}i\ \texttt{A\_\_}j \left[e^{-e^{\texttt{E}}\ \texttt{t}} + \sum_{n=1}^{N-1} \texttt{B\_}n\texttt{\_\_}i\ \texttt{B\_}n\texttt{\_\_}j\ e^{-(e^{\texttt{E}} + e^{\texttt{dE\_1}} + ... + e^{\texttt{dE\_}n})\texttt{t}} \right] \\
& +(-1)^{\texttt{t}+1} \texttt{Ao\_\_}i\ \texttt{Ao\_\_}j \left[e^{-e^{\texttt{Eo}}\texttt{t}} + \sum_{m=1}^{M-1} \texttt{Bo\_}m\texttt{\_\_}i\ \texttt{Bo\_}m\texttt{\_\_}j\ e^{-(e^{\texttt{Eo}} + e^{\texttt{dEo\_1}} + ... + e^{\texttt{dEo\_}m})\texttt{t}} \right]
\end{aligned}
$$

- variable(s), parameter(s): same as in Sec. A.5.3

A.6 Matrix two-point models, type II

In type II matrix models, the ground state is not special. All amplitudes, including the ground-state amplitude, are written as a product $\texttt{A\_\_}i\ \texttt{B\_}n\texttt{\_\_}i$ (i.e., n now starts from 0). This means that max($\texttt{dim\_1}$, $\texttt{dim\_2}$) of the parameters $\{\texttt{B\_}n\texttt{\_\_}i\}$ are redundant, and Bayesian constraints must be activated. The typical usage is to constrain the parameters $\texttt{B\_}(i-1)\texttt{\_\_}i$ to $1 \pm \epsilon$ with a very small prior width ϵ, which effectively eliminates these parameters from the functions.

A.6.1 Multiple Exponentials

- function(s): for $i = 1 ... \texttt{dim\_1}$, $j = 1 ... \texttt{dim\_2}$:

$$
f_{ij}(\texttt{t}) = \texttt{A\_\_}i\ \texttt{A\_\_}j \sum_{n=0}^{N-1} \texttt{B\_}n\texttt{\_\_}i\ \texttt{B\_}n\texttt{\_\_}j\ e^{-(\texttt{E}+...+\texttt{dE\_}n)\texttt{t}}
$$

- variable(s): $\texttt{t}$
- parameter(s): $\{\texttt{A\_\_}i\}$, $\{\texttt{B\_}n\texttt{\_\_}i\}$ (for $i = 1 ...$max($\texttt{dim_1}$, $\texttt{dim\_2}$)), $\texttt{E}$, $\{\texttt{dE\_}n\}$
- properties:

Key	content	type
n_exp	N	integer ≥ 1
A_name	A	string
B_name	B	string
E_name	E	string
dE_name	dE	string
t_name	t	string
dim_1	$i = 1 ... \texttt{dim\_1}$	integer ≥ 1
dim_2	$j = 1 ... \texttt{dim\_2}$	integer ≥ 1

A.6.2 Multiple Exponentials with Exponential Energies

- function(s): for $i = 1 ... \texttt{dim\_1}$, $j = 1 ... \texttt{dim\_2}$:

$$
f_{ij}(\texttt{t}) = \texttt{A\_\_}i\ \texttt{A\_\_}j \sum_{n=0}^{N-1} \texttt{B\_}n\texttt{\_\_}i\ \texttt{B\_}n\texttt{\_\_}j\ e^{-(e^{\texttt{E}}+...+e^{\texttt{dE\_}n})\texttt{t}}
$$

- variable(s), parameter(s), properties: same as in Sec. A.6.1

A.7 Triangular matrix two-point models

These models are like matrix models with `dim_1=dim_2`, but the triangular models consist of only the functions with $j \geq i$. This is intended for matrix fits with exactly symmetric (i.e. symmetrized) data.

A.7.1 Multiple Exponentials

- function(s): for $i = 1...\mathtt{dim}$, $j = i...\mathtt{dim}$ (total number of functions $= \mathtt{dim}(\mathtt{dim}+1)/2$):

$$f_{ij}(\mathtt{t}) = \mathtt{A\_\_}i\ \mathtt{A\_\_}j \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\ \mathtt{B\_}n\mathtt{\_\_}j\ e^{-(\mathtt{E}+\mathtt{dE\_1}+...+\mathtt{dE\_}n)\mathtt{t}} \right]$$

- variable(s): `t`
- parameter(s): $\{\mathtt{A\_\_}i\}$, $\{\mathtt{B\_}n\mathtt{\_\_}i\}$ (for $i = 1...\mathtt{dim}$), `E`, $\{\mathtt{dE\_}n\}$
- properties:

Key	content	type
`n_exp`	N	integer ≥ 1
`A_name`	`A`	string
`B_name`	`B`	string
`E_name`	`E`	string
`dE_name`	`dE`	string
`t_name`	`t`	string
`dim`	$i = 1...\mathtt{dim}$, $j = i...\mathtt{dim}$	integer ≥ 1

A.7.2 Multiple Exponentials with Exponential Energies

- function(s): for $i = 1...\mathtt{dim}$, $j = i...\mathtt{dim}$ (total number of functions $= \mathtt{dim}(\mathtt{dim}+1)/2$):

$$f_{ij}(\mathtt{t}) = \mathtt{A\_\_}i\ \mathtt{A\_\_}j \left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\ \mathtt{B\_}n\mathtt{\_\_}j\ e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+...+e^{\mathtt{dE\_}n})\mathtt{t}} \right]$$

- variable(s), parameter(s), properties: same as in Sec. A.7.1

A.8 Triangular matrix two-point models, type II

These models are like type II matrix models with `dim_1=dim_2`, but the triangular models consist of only the functions with $j \geq i$. This is intended for matrix fits with exactly symmetric (i.e. symmetrized) data.

A.8.1 Multiple Exponentials

Note: some parameters are redundant (see Sec. A.6).

- function(s): for $i = 1...\mathtt{dim}$, $j = i...\mathtt{dim}$ (total number of functions $= \mathtt{dim}(\mathtt{dim}+1)/2$):

$$f_{ij}(\mathtt{t}) = \mathtt{A\_\_}i\ \ \mathtt{A\_\_}j \sum_{n=0}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\ \ \mathtt{B\_}n\mathtt{\_\_}j\ \ e^{-(\mathtt{E}+\ldots+\mathtt{dE\_}n)\mathtt{t}}$$

- variable(s): $\mathtt{t}$
- parameter(s): $\{\mathtt{A\_\_}i\}$, $\{\mathtt{B\_}n\mathtt{\_\_}i\}$ (for $i = 1...\max(\mathtt{dim\_1}, \mathtt{dim\_2})$), $\mathtt{E}$, $\{\mathtt{dE\_}n\}$
- properties:

Key	content	type
`n_exp`	N	integer ≥ 1
`A_name`	`A`	string
`B_name`	`B`	string
`E_name`	`E`	string
`dE_name`	`dE`	string
`t_name`	`t`	string
`dim`	$i = 1...\mathtt{dim}$, $j = i...\mathtt{dim}$	integer ≥ 1

A.8.2 Multiple Exponentials with Exponential Energies

Note: some parameters are redundant (see Sec. A.6).

- function(s): for $i = 1...\mathtt{dim}$, $j = i...\mathtt{dim}$ (total number of functions $= \mathtt{dim}(\mathtt{dim}+1)/2$):

$$f_{ij}(\mathtt{t}) = \mathtt{A\_\_}i\ \ \mathtt{A\_\_}j \sum_{n=0}^{N-1} \mathtt{B\_}n\mathtt{\_\_}i\ \ \mathtt{B\_}n\mathtt{\_\_}j\ \ e^{-(e^{\mathtt{E}}+\ldots+e^{\mathtt{dE\_}n})\mathtt{t}}$$

- variable(s), parameter(s), properties: same as in Sec. A.8.1

A.9 2-Point Correlator, Non-symmetric Matrix Fit

Here the amplitudes are factorized into an outer product of *two different vectors*, rather than the outer product of a vector with itself, as in the models of Sec. A.5. Because of a reparametrization invariance, some amplitude parameters need to be eliminated to get unique results. This has already been done in the following models, so that the fit functions are different for $i = 1$ vs $i > 1$ (see below).

As in Sec. A.5, the required storage order in the data files is such that the first index (i) runs slow and the second index (j) runs fast.

A.9.1 Multiple Exponentials

- function(s):

  for $i = 2 \ldots \mathtt{dim\_1}$, $j = 1 \ldots \mathtt{dim\_2}$:

$$f_{ij}(\mathtt{t}) = \mathtt{Ax\_\_}i\ \mathtt{Ay\_\_}j\ \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{Bx\_}n\mathtt{\_\_}i\ \mathtt{By\_}n\mathtt{\_\_}j\ e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right]$$

  for $i = 1$, $j = 1 \ldots \mathtt{dim\_2}$:

$$f_{1j}(\mathtt{t}) = \mathtt{Ay\_\_}j\ \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1}\ \mathtt{By\_}n\mathtt{\_\_}j\ e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right]$$

- variable(s): $\mathtt{t}$
- parameter(s): $\{\mathtt{Ax\_\_}i, \mathtt{Bx\_}n\mathtt{\_\_}i\}$ (for $i = 2 \ldots \mathtt{dim\_1}$), $\{\mathtt{Ay\_\_}j, \mathtt{By\_}n\mathtt{\_\_}j\}$ (for $j = 1 \ldots \mathtt{dim\_2}$), $\mathtt{E}$, $\{\mathtt{dE\_}n\}$

A.9.2 Multiple Exponentials with Exponential Energies

- function(s):

  for $i = 2 \ldots \mathtt{dim\_1}$, $j = 1 \ldots \mathtt{dim\_2}$:

$$f_{ij}(\mathtt{t}) = \mathtt{Ax\_\_}i\ \mathtt{Ay\_\_}j\ \left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{Bx\_}n\mathtt{\_\_}i\ \mathtt{By\_}n\mathtt{\_\_}j\ e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+\ldots+e^{\mathtt{dE\_}n})\mathtt{t}} \right]$$

  for $i = 1$, $j = 1 \ldots \mathtt{dim\_2}$:

$$f_{1j}(\mathtt{t}) = \mathtt{Ay\_\_}j\ \left[e^{-e^{\mathtt{E}}\ \mathtt{t}} + \sum_{n=1}^{N-1}\ \mathtt{By\_}n\mathtt{\_\_}j\ e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+\ldots+e^{\mathtt{dE\_}n})\mathtt{t}} \right]$$

- variable(s), parameter(s): same as in Sec. A.9.1

A.9.3 Multiple (incl. oscillating) Exponentials

- function(s):

  for $i = 2 \ldots \mathtt{dim\_1}$, $j = 1 \ldots \mathtt{dim\_2}$:

$$\begin{aligned} f_{ij}(\mathtt{t}) &= \mathtt{Ax\_\_}i\ \mathtt{Ay\_\_}j\ \left[e^{-\mathtt{E}\ \mathtt{t}} + \sum_{n=1}^{N-1} \mathtt{Bx\_}n\mathtt{\_\_}i\ \mathtt{By\_}n\mathtt{\_\_}j\ e^{-(\mathtt{E}+\mathtt{dE\_1}+\ldots+\mathtt{dE\_}n)\mathtt{t}} \right] \\ &\quad +(-1)^{\mathtt{t}+1}\ \mathtt{Aox\_\_}i\ \mathtt{Aoy\_\_}j\ \left[e^{-\mathtt{Eo}\ \mathtt{t}} + \sum_{m=1}^{M-1} \mathtt{Box\_}m\mathtt{\_\_}i\ \mathtt{Boy\_}m\mathtt{\_\_}j\ e^{-(\mathtt{Eo}+\mathtt{dEo\_1}+\ldots+\mathtt{dEo\_}m)\mathtt{t}} \right] \end{aligned}$$

  for $i = 1$, $j = 1 \ldots \mathtt{dim\_2}$:

$$\begin{aligned} f_{1j}(\texttt{t}) \;=\;& \texttt{Ay\_\_}j \left[e^{-\texttt{E t}} + \sum_{n=1}^{N-1} \texttt{By\_}n\texttt{\_\_}j \; e^{-(\texttt{E+dE\_1+...+dE\_}n)\texttt{t}} \right] \\ & +(-1)^{\texttt{t+1}}\, \texttt{Aoy\_\_}j \left[e^{-\texttt{Eo t}} + \sum_{m=1}^{M-1} \texttt{Boy\_}m\texttt{\_\_}j \; e^{-(\texttt{Eo+dEo\_1+...+dEo\_}m)\texttt{t}} \right] \end{aligned}$$

- variable(s): $\texttt{t}$
- parameter(s): $\{\texttt{Ax\_\_}i,\ \texttt{Bx\_}n\texttt{\_\_}i\}$ (for $i = 2...\texttt{dim\_1}$), $\{\texttt{Ay\_\_}j,\ \texttt{By\_}n\texttt{\_\_}j\}$ (for $j = 1...\texttt{dim\_2}$), $\texttt{E}$, $\{\texttt{dE\_}n\}$, $\{\texttt{Aox\_\_}i,\ \texttt{Box\_}m\texttt{\_\_}i\}$ (for $i = 2...\texttt{dim\_1}$), $\{\texttt{Aoy\_\_}j,\ \texttt{Boy\_}m\texttt{\_\_}j\}$ (for $j = 1...\texttt{dim\_2}$), $\texttt{Eo}$, $\{\texttt{dEo\_}n\}$

A.9.4 Multiple (incl. oscillating) Exponentials with Exponential Energies

- function(s):

  for $i = 2...\texttt{dim\_1}$, $j = 1...\texttt{dim\_2}$:

$$\begin{aligned} f_{ij}(\texttt{t}) \;=\;& \texttt{Ax\_\_}i\ \texttt{Ay\_\_}j \left[e^{-e^{\texttt{E}}\,\texttt{t}} + \sum_{n=1}^{N-1} \texttt{Bx\_}n\texttt{\_\_}i\ \texttt{By\_}n\texttt{\_\_}j \; e^{-(e^{\texttt{E}}+e^{\texttt{dE\_1}}+...+e^{\texttt{dE\_}n})\texttt{t}} \right] \\ & +(-1)^{\texttt{t+1}}\, \texttt{Aox\_\_}i\ \texttt{Aoy\_\_}j \left[e^{-e^{\texttt{Eo}}\,\texttt{t}} + \sum_{m=1}^{M-1} \texttt{Box\_}m\texttt{\_\_}i\ \texttt{Boy\_}m\texttt{\_\_}j \; e^{-(e^{\texttt{Eo}}+e^{\texttt{dEo\_1}}+...+e^{\texttt{dEo\_}m})\texttt{t}} \right] \end{aligned}$$

  for $i = 1$, $j = 1...\texttt{dim\_2}$:

$$\begin{aligned} f_{1j}(\texttt{t}) \;=\;& \texttt{Ay\_\_}j \left[e^{-e^{\texttt{E}}\,\texttt{t}} + \sum_{n=1}^{N-1} \texttt{By\_}n\texttt{\_\_}j \; e^{-(e^{\texttt{E}}+e^{\texttt{dE\_1}}+...+e^{\texttt{dE\_}n})\texttt{t}} \right] \\ & +(-1)^{\texttt{t+1}}\, \texttt{Aoy\_\_}j \left[e^{-e^{\texttt{Eo}}\,\texttt{t}} + \sum_{m=1}^{M-1} \texttt{Boy\_}m\texttt{\_\_}j \; e^{-(e^{\texttt{Eo}}+e^{\texttt{dEo\_1}}+...+e^{\texttt{dEo\_}m})\texttt{t}} \right] \end{aligned}$$

- variable(s), parameter(s): same as in Sec. A.9.3

A.10 3-Point Correlator

A.10.1 Multiple Exponentials

- function(s):

$$f(\texttt{t},\ \texttt{T}) = \texttt{A} \left[e^{-\texttt{F t}} e^{-\texttt{E(T-t)}} + \sum_{\substack{n\ = 0\,...\,N\ -1, \\ n' = 0\,...\,N' - 1, \\ (n,n') \neq (0,0)}} \texttt{B\_}n'\texttt{\_}n \; e^{-(\texttt{F+dF\_1+...+dF\_}n')\texttt{t}}\, e^{-(\texttt{E+dE\_1+...+dE\_}n)(\texttt{T-t})} \right]$$

- variable(s): $\texttt{t}$, $\texttt{T}$
- parameter(s): $\texttt{A}$, $\{\texttt{B\_}n'\texttt{\_}n\}$, $\texttt{E}$, $\{\texttt{dE\_}n\}$, $\texttt{F}$, $\{\texttt{dF\_}n'\}$

A.10.2 Multiple Exponentials with Exponential Energies

- function(s):

$$f(\mathtt{t},\ \mathtt{T}) = \mathtt{A} \left[e^{-e^{\mathtt{F}}\mathtt{t}} e^{-e^{\mathtt{E}}(\mathtt{T}-\mathtt{t})} + \sum_{\substack{n\ = 0 \dots N\ - 1,\\ n' = 0 \dots N' - 1,\\ (n,n') \neq (0,0)}} \mathtt{B\_}n'\_n\ e^{-(e^{\mathtt{F}}+e^{\mathtt{dF\_1}}+\dots+e^{\mathtt{dF\_}n'})\mathtt{t}}\ e^{-(e^{\mathtt{E}}+e^{\mathtt{dE\_1}}+\dots+e^{\mathtt{dE\_}n})(\mathtt{T}-\mathtt{t})} \right]$$

- variable(s), parameter(s): same as in Sec. A.10.1

A.10.3 Multiple (incl. oscillating) Exponentials

- function(s):

For $M > 0$ and $M' > 0$:

$$\begin{aligned}
f(\mathtt{t},\ \mathtt{T}) \quad = \quad & \mathtt{Aee} \left[e^{-\mathtt{F}\ \mathtt{t}} e^{-\mathtt{E}(\mathtt{T}-\mathtt{t})} + \sum_{\substack{n\ = 0 \dots N\ - 1,\\ n' = 0 \dots N' - 1,\\ (n,n') \neq (0,0)}} \mathrm{Bee\_}n'\_n\ e^{-(\mathtt{F}+\mathtt{dF\_1}+\dots+\mathtt{dF\_}n')\mathtt{t}}\ e^{-(\mathtt{E}+\mathtt{dE\_1}+\dots+\mathtt{dE\_}n)(\mathtt{T}-\mathtt{t})} \right] \\
+ \quad (-1)^{t} \quad & \mathtt{Aoe} \left[e^{-\mathrm{Fo}\ \mathtt{t}} e^{-\mathtt{E}(\mathtt{T}-\mathtt{t})} + \sum_{\substack{n\ = 0 \dots N\ - 1,\\ m' = 0 \dots M' - 1,\\ (n,m') \neq (0,0)}} \mathrm{Boe\_}m'\_n\ e^{-(\mathrm{Fo}+\mathtt{dFo\_1}+\dots+\mathtt{dFo\_}m')\mathtt{t}}\ e^{-(\mathtt{E}+\mathtt{dE\_1}+\dots+\mathtt{dE\_}n)(\mathtt{T}-\mathtt{t})} \right] \\
+\ (-1)^{(T-t)}\ & \mathtt{Aeo} \left[e^{-\mathtt{F}\ \mathtt{t}} e^{-\mathtt{Eo}(\mathtt{T}-\mathtt{t})} + \sum_{\substack{m\ = 0 \dots M\ - 1,\\ n' = 0 \dots N' - 1,\\ (m,n') \neq (0,0)}} \mathrm{Beo\_}n'\_m\ e^{-(\mathtt{F}+\mathtt{dF\_1}+\dots+\mathtt{dF\_}n')\mathtt{t}}\ e^{-(\mathrm{Eo}+\mathtt{dEo\_1}+\dots+\mathtt{dEo\_}m)(\mathtt{T}-\mathtt{t})} \right] \\
+ \quad (-1)^{T} \quad & \mathtt{Aoo} \left[e^{-\mathrm{Fo}\ \mathtt{t}} e^{-\mathtt{Eo}(\mathtt{T}-\mathtt{t})} + \sum_{\substack{m\ = 0 \dots M\ - 1,\\ m' = 0 \dots M' - 1,\\ (m,m') \neq (0,0)}} \mathrm{Boo\_}m'\_m\ e^{-(\mathrm{Fo}+\mathtt{dFo\_1}+\dots+\mathtt{dFo\_}m')\mathtt{t}}\ e^{-(\mathrm{Eo}+\mathtt{dEo\_1}+\dots+\mathtt{dEo\_}m)(\mathtt{T}-\mathtt{t})} \right]
\end{aligned}$$

Note: for $M' = 0$, the second and fourth row disappear.
for $M = 0$, the third and fourth row disappear.

- variable(s): `t`, `T`

- parameter(s):

Aee, $\{\mathrm{Bee\_}n'\_n\}$, $\mathtt{E}$, $\{\mathtt{dE\_}n\}$, $\mathtt{F}$, $\{\mathtt{dF\_}n'\}$

For $M > 0$ also $\quad \texttt{Aeo},\ \{\texttt{Beo\_}n'\texttt{\_}m\},\quad \texttt{Eo},\ \{\texttt{dEo\_}m\}$

For $M' > 0$ also $\quad \texttt{Aoe},\ \{\texttt{Boe\_}m'\texttt{\_}n\},\quad \texttt{Fo},\ \{\texttt{dFo\_}m'\}$

For ($M > 0$ and $M' > 0$) also $\quad \texttt{Aoo},\ \{\texttt{Boo\_}m'\texttt{\_}m\}$

A.10.4 Multiple (incl. oscillating) Exponentials with Exponential Energies

- function: same as in Sec. A.10.3, but with the following replacements:

$$\begin{aligned}
\texttt{E} &\to e^{\texttt{E}} \\
\texttt{dE\_}n &\to e^{\texttt{dE\_}n} \\
\texttt{F} &\to e^{\texttt{F}} \\
\texttt{dF\_}n' &\to e^{\texttt{dF\_}n'} \\
\texttt{Eo} &\to e^{\texttt{Eo}} \\
\texttt{dEo\_}m &\to e^{\texttt{dEo\_}m} \\
\texttt{Fo} &\to e^{\texttt{Fo}} \\
\texttt{dFo\_}m' &\to e^{\texttt{dFo\_}m'}
\end{aligned}$$

- variable(s), parameter(s): same as in Sec. A.10.3

A.11 3-Point Correlator, Vector Fit

The "scalar" three-point models listed in section A.10 are also available as "vector" three-point models. These models require the dimension `dim` of the vector. A vector model has then `dim` functions $f_i(\texttt{t})$ ($i = 1...\texttt{dim}$) of the same form as the underlying scalar model. These functions have individual amplitude parameters, but share all the energy parameters. For example, the functions for `threept_multi_exp_vec_model` are

$$f_i(\texttt{t},\ \texttt{T}) = \texttt{A\_\_}i \left[e^{-\texttt{F t}} e^{-\texttt{E(T}-\texttt{t)}} + \sum_{\substack{n\ = 0\,...\,N\ - 1,\\ n' = 0\,...\,N' - 1,\\ (n,n') \neq (0,0)}} \texttt{B\_}n'\texttt{\_}n\texttt{\_\_}i \ e^{-(\texttt{F+dF\_1+}...\texttt{+dF\_}n')\texttt{t}} \, e^{-(\texttt{E+dE\_1+}...\texttt{+dE\_}n)(\texttt{T}-\texttt{t})} \right]$$

for $i = 1...\texttt{dim}$.