PAUL SCHERRER INSTITUT

Maximum Entropy Emacs Environment

for Tanya Riseman's MaxEnt

from Andreas Suter — May 18, 2001

Labor für Myonspinspektroskopie
Paul Scherrer Institut
CH – 5232 Villigen PSI
Switzerland

Andreas Suter, May 18, 2001

Contents

1 Introduction 3

2 Structure of the Control <mein>–Input File for MaxEnt and emacs handling 3

2.1 <mein>–Input File 3

2.2 emacs commands for MaxEnt 7

3 Installation Procedure 8

3.1 Already installed on the system, but you are a new user 8

3.2 How to install the whole stuff 9

4 Used emacs lisp File 9

5 Used c–Files 11

5.1 me_shell 11

5.2 me2paw 11

5.3 me_plot 11

A <mein>–default file 12

1 Introduction

This document describes the, hopefully, user friendly implementation of MaxEnt into the emacs environment. Here the MaxEnt fortran program of Tanya Riseman (TR) was used. It does the essential job in terms of the maximum entropy analysis, however is rather inconvenient to be used alone. This is the very reason, which motivated all this pain to write a kind of shell for a more convenient handling. To be more precise, it describes to Version 0.9β.

It has a similar set up then the WKM package from Braunschweig. Especially the emacs lisp stuff is a modification of their WKM emacs lisp implementation.

For an user only the Sec. 2 is need to understand what (s)he has to do in order to get some work done. I still hope the at one glorious day TR will actually write something like a documentation. This not only would help me, but most probably all the guys which are going to blame me for what I did here. In Sec. 2 I also will try to give my best to guide an user through the thrilling challenge of getting all the parameters in coherent way fixed.

The whole package is now controlled directly out of the emacs which is available on most platforms. It has the advantage that by just changing an input parameter file and selecting some pulldowns. To start MaxEnt , plotting the outcome of the fit, having the paw–files [1, 2, 3] ready. For people which do not have any clue about maximum entropy analysis, I am very sorry but I only can give you here some references [4, 5, 6].

2 Structure of the Control <mein>–Input File for MaxEnt and emacs handling

This section is going to describe how to use the emacs as an environment to do maximum entropy analysis for the LEμSR experiment at the Paul Scherrer Institut (PSI) in Switzerland. Especially i.e. that I *only* tested the BRA data format of the MaxEnt package of TR so far! To control the MaxEnt program a whole bunch of parameters are needed. This parameters are collected in a simple text file with the extension .mein (maximum entropy in), which I will call <mein>–input file. In a first step I will describe the structure of this <mein>–input file followed by a semi–rigorous description of all the parameters. In a next paragraph the emacs –related commands will be presented.

2.1 <mein>–Input File

Since the MaxEnt analysis is *not* quite trivial, I decided form the very beginning that a user should be able to add *lots* of comments into the <mein>–input file. This seems also to be useful, since additional run–related comments can be added. The idea is that in the near future a standard <mein>–input file will be generate automatically from the midas–database, controlling the run. This is not implemented yet, though the needed dummy–routines are already there. I will come back to that later on. Comments in the <mein>–input file start with a '%'. The rules are like in LaTeX, i.e. that they can be a line consisting only of a comment, starting with a '%', or after a key–line a comment can be added, again starting with a '%'. Allowed lines would e.g. be

```
% RUN DESCRIPTION IN WORDS ------------------------------------------
%   In this run (r0000), we ...
```

or

```
BRA    : default data type  % available types are (BRA, PAW, ...)
```

Also any number of empty lines are allowed, which are just parsed over. In the Appendix A such a default <mein>–file can be found.

The relevant input–parameter syntax for the MaxEnt is as following:

```
  value : keyword [% comment]
```

where value–structure depends very much on the keyword and will be explained in detail below. I would like to stress that '`: keyword`' is *one* entity, i.e. the '`: `' are part of it! At the moment all the entries are *key sensitive*, i.e. upper and lower case is *not* the same. In Tab. 1 is an overview of all the keywords used. If the value is set to `####` it means that `MaxEnt` is evaluating it, and after an analysis this fields are changed to numbers.

typical value	keyword	comment	changeable by `MaxEnt`
	% What so ever	comment	no
BRA	`: default data type`	available data types see the text	no
/afs/.../data	`: data source directory`	where will `MaxEnt` find the data	no
2001-04-26, 12-23	`: date & time`		no
`r0000_0`	`: run`		no
4	`: total number of groups`	# groups/detectors	no
2, 4	`: remove groups`	instead `NONE` is possible	no
Y	`: phases fixed`	flag telling `MaxEnt` if phases are to be optimized	no
$(0.0, -90.0, 180.0, 90.0)$	`: phase values`	depending on `: phases fixed`	yes/no
$(0, 0, 12000)$	`: data bins group1`	for each group needed!	no
$(0, 0), (0, 0), (0, 0), (0, 0)$	`: bins used to estimate bkg`	$(0, 0)$ means theoretical esitmate	no
$(0.0, 1.2, 0.002)$	`: time window for fit`	in μs	no
$(20.0, 2.0, 100.0)$	`: var time binning`	in μs	no
Y	`: time binning convolution`		no
0.0	`: bkg signal asymmetry`		no
0.0	`: sigma apodization`	in μs	no
1.0	`: sigma looseness factor`	$\sigma \to \ell \cdot \sigma$	no
1.0	`: twiddle factor`		no
$(0.0, 0.04)$	`: field range`	in Tesla	no
13	`: FFT power n`	2^n used in iFFT for `MaxEnt`	no
Y	`: use sqrt(theory) errors`		no
Y	`: use final fitting enhancement factor`		no
(0.001/`####`)	`: termination level`		no/yes
`####`	`: number of fited events`		yes
`####`	`: default level`		yes
`####`	`: noise`		yes
`####`	`: normalized Chi^2`		yes
`####`	`: used enhancement`	if $\gg 1 \Rightarrow$ something is worng	yes
`####`	`: averaged asymmetry`		yes
`####`	`: mean value`		yes
`####`	`: sqrt of M2`		yes
`####`	`: alpha skew`		yes

Table 1: *Compilation of all the need keywords. For a detailed description see the text.*

Lets start describing them one by one.

: default data type The following formats are available and implemented: `BRA` which is the LEμSR data format we are using here at the PSI in the LEμSR group. The other formats are: `USR, PSI, P3E, TRI, PAW, ORG`. Unfortunately I still have no documentation from TR at the moment and I am to lazy to go through here whole source code, i.e. except for `BRA` you are at your one at the moment!

: data source directory Here you have to enter the path, where `MaxEnt` will find your data to be analyzed. An example is

```
% where are the data?
/afs/psi.ch/project/nemu/FeAg : data source directory
```

: data & time Should be quite obvious. Unfortunately I *mustn't* use a ':' in the time (sorry!)

: run for the data format `BRA` this is something like `r0000_0`.

: total number of groups The number of groups/detectors in this run. An example is

```
4 : total number of groups
```

: remove groups one can remove every set of groups one doesn't want to include in the analysis. An example is

```
1, 3 : remove groups
```

If one wants to keep all the groups, the entry is

```
NONE : remove groups
```

: phases fixed This flag indicates if `MaxEnt` is allowed also to optimize the relative phases between the groups. As far as I can see at the moment, it seems to be healthier not to use this extensively, especially *not* without care! The two possible values are `Y` or `N`.

: phase values This is a set of phase default entries which, depending on the '`: phases fixed`' flag, can be modified by `MaxEnt` . The default setting is for 4 groups and looks like this

```
(0.0, -90.0, 180.0, 90.0) : phase values
```

: data bins group1 For *every* group there must be such an entry, even if they are not going to be used in a particular analysis! The syntax is: `(it0, start, end)`, e.g.

```
(0, 0, 12000) : data bins group1
```

: bins used to estimate bkg Here the values have the structure `(100,233), (102, 235), (100,233), (101,230)`. If one doesn't want/can't do this it would look like

```
(0,0),(0,0),(0,0),(0,0) : bins used to estimate bkg
```

: time window for fit syntax is `(start, end, step-resolution)` in μs, e.g.

```
(0.1, 1.8, 0.002) : time window for fit
```

: var time binning This is something fancy implemented by TR. Since, in a decay spectrum, the statistic is getting worse and worse as function of time, it might be clever to try to compensate this by combining more data at longer times into a virtual bin to "improve" the statistics. The syntax is (`start-time` t_0, `pwr, width`) where the parameter describe a generalized Gaussian

$$\text{binning} \propto \begin{cases} 1 & t < t_0, \\ \exp\left(+\dfrac{(\text{width}\cdot(t-t_0))^{\text{pwr}}}{\text{pwr}}\right) & t > t_0. \end{cases}$$

If you do not want to use it, just use the following entry, which shifts the starting–time to very late times.

```
(20.0, 2.0, 100) : var time binning
```

: time binning convolution This flags indicates if `MaxEnt` tries to compensate for "loss" of asymmetry at higher frequencies due to the binning (sampling). Possible values are `Y` or `N`. If you do not understand this, just keep it to `Y`, i.e.

```
Y : time binning convolution
```

: bkg signal asymmetry It is, what it is.

: sigma apodization given in μs. The idea is to blowing up the errors at longer times. Something like

$$\sigma_{t_i} \to f(t_i) \cdot \sigma_{t_i}$$

where $f(t_i)$ is a monotonic increasing function of time. Since I still do not have an exact description of TR, I am not quite sure how she implemented it, i.e. which $f(t_i)$ she used. If one initializes `: sigma apodization` as

```
0.0 : sigma apodization
```

no apodization is done.

: sigma looseness factor is simply a value $\ell > 1$ with $\sigma \to \ell \cdot \sigma$. This definition already shows that one should only choose values rather close to unity. Default is

```
1.0 : sigma looseness factor
```

: twiddle factor I have no clue what this thing is doing. Let's hope TR documentation will illuminate us. Therefore the best choice at the moment is

```
1.0 : twiddle factor
```

: field range given in Tesla. The Syntax is `(start, end)`.

: FFT power n tells `MaxEnt` which power n it should use to perform the iFFT used in the algorithm. n should be of the order $2^n \approx$ (end − start)/step-resolution defined in `: time window for fit`. Testing showed that $n < 17$ otherwise `MaxEnt` crashes quite often. For a reasonable set of data, n will be between 10 and 16. Default

```
13 : FFT power n
```

: use sqrt(theory) errors is a tag with the values '`Y`' or '`N`'. '`N`' uses an estimate from the data. Default is

```
Y : use sqrt(theory) errors
```

: use final fitting enhancement factor This tag is by default '`Y`' ('`Y`' or '`N`' possible). `MaxEnt` always underestimates the time evolution signal amplitude (smoothness of the spectrum). If this value is set '`Y`' `MaxEnt` tries to compensate for it. Further below there will be a description of this correction value `: used enhancement`.

: termination level Syntax: `(input-value/return-value)`. If the `input-value` is 0.001 then a very accurate maximum entropy is looked for, if it is 0.02 the analysis is quicker but less reliable. the `return-value` is calculate during the analysis and written back as soon as the first analysis is done. The `return-value` should always be $\leq$ `input-value`, otherwise it is a clear sign that `MaxEnt` had a hard time to converge. Default is

```
(0.001/####) : termination level
```

: number of fited events return value of `MaxEnt` , therefore initialized to the value `####`.

: default level return value of `MaxEnt` , therefore initialized to the value `####`. It is a kind of confidence level of the spectrum.

: noise return value of `MaxEnt` , therefore initialized to the value `####`. Values well above the noise level are more or less reliable. The noise level should be always above the default level.

: normalized Chiˆ2 return value of `MaxEnt` , therefore initialized to the value `####`. If it is *not* within its error bars, the fitting output should be taken with quite some suspicion.

: used enhancement return value of `MaxEnt` , therefore initialized to the value `####`. If `use final fitting enhancement factor` was set 'Y', then the time data were corrected be this factor and hence the asymmetry. If this number after a fit deviates to strongly from unity, the fit is *not* good (as a thumb rule, values below 1.3 are kind of acceptable).

: averaged asymmetry return value of `MaxEnt` , therefore initialized to the value `####`.

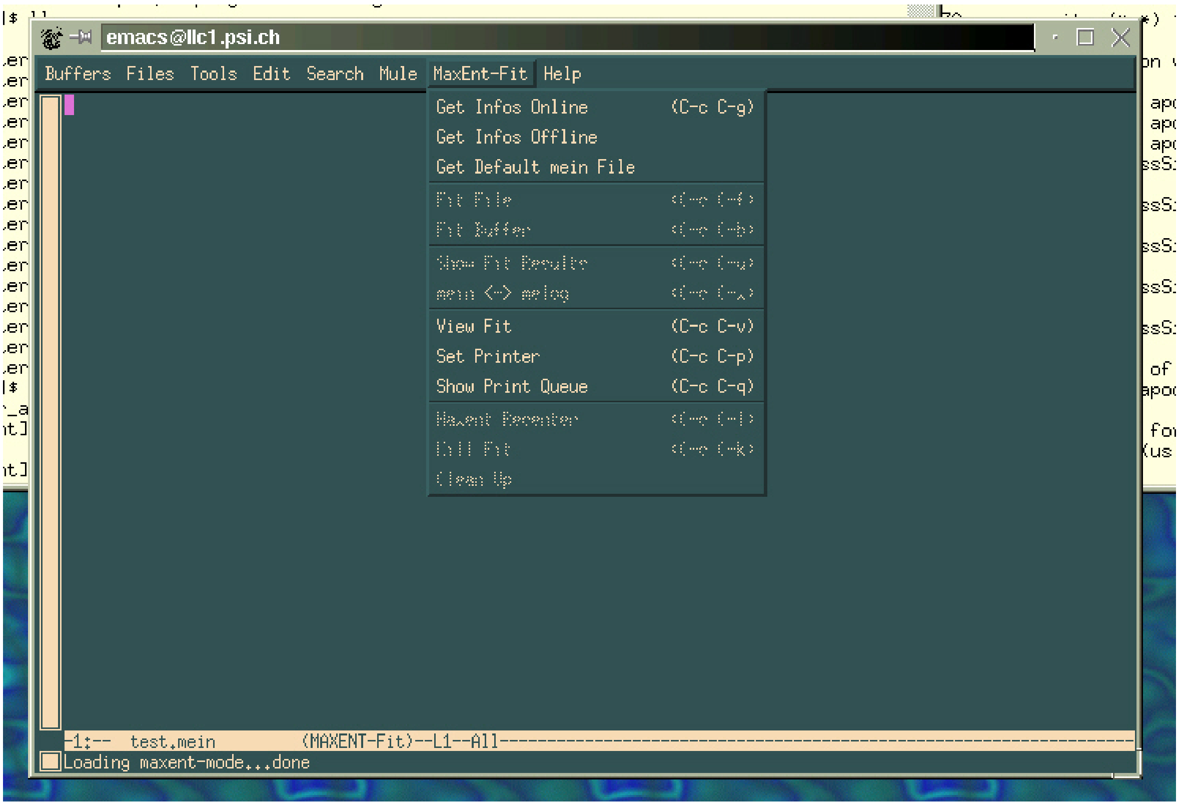
: mean value return value of `MaxEnt` , therefore initialized to the value `####`.

: sqrt M2 return value of `MaxEnt` , therefore initialized to the value `####`.

: alpha skew return value of `MaxEnt` , therefore initialized to the value `####`.

2.2 emacs commands for MaxEnt

For an overview how to use the `emacs` have a look at Ref. [7]. Do *not* use the `xemacs`, since a lot of the features I'm going to describe wont work! If everything is installed properly (see Sec. 3) you can generate a new file with extension `.mein` and automatically there will be an additional pull–down option '`MaxEnt-Fit`' available. It will look something like this



```
-1:--  test.mein        (MAXENT-Fit)--L1--All-----------------------------------------------------------
Loading maxent-mode...done
```

Figure 1: `emacs` *start up window with the* `MaxEnt-fit` *menu.*

The pull–down consist of the following commands:

Get Infos Online	not implemented yet
Get Infos Offline	not implemented yet
Get Default mein File	reads a default file
Fit File	only available after reading a default mein file
Fit Buffer	only available after reading a default mein file
Show Fit Results	only available after fitting
mein <-> melog	only available after fitting
View Fit	only available after fitting
Set Printer	
Show Printer Queue	
MaxEnt recenter	
Kill Fit	
Clean Up	

Some of these commands are ghosty and some straight available. The reason for it is that to execute some of these commands other have to been executed first. All this stuff is not bullet proofed yet so that it might do something slightly odd.

Next the meaning of these commands will be elucidated by going through a imaginary fitting session. Having generate first an empty new <mein>–file. We can proceed by entering all the comments and keywords and so on. This is very tiresome and will not be done. In the future the first two command will dock to the midas–database which controls the whole experiment. Since this is not implemented yet, a default <mein>–file can be loaded via the command 'Get Default mein File'.

Having loaded this default, one should adjust the file to the requirements of its one run. After this work is done one can use the two actual fitting commands, namely 'Fit File' or 'Fit Buffer'. The difference is that the first directly overrides the working file, whereas the later produces an additional bunch of temporary files leaving the working file untouched. Having started the fitting command, the emacs window splits into two and in the lower half panel some action will take place. As snap–shot of this is shown in Fig. 2.
The fitting takes its time, so relax and have a cup of coffee. If all the action stops in the lower window (MaxEnt shell window) the fit is done and you are proud owner of a whole bunch of new files. The essential once are (assuming the name of your run is r0000_0):

r0000_0.mein	contains the updated <mein>–file.
r0000_0.mdb	this is the output of MaxEnt and contains in principle all the information.
r0000_0.hbook	This is a cernlib histogram file, which holds all the histogram files produced by MaxEnt .

There is going to be a whole bunch of additional files which you do not really need. They can be easily removed by the command 'Clean Up'. The .hbook–file might be used in paw for further analysis of the spectra.

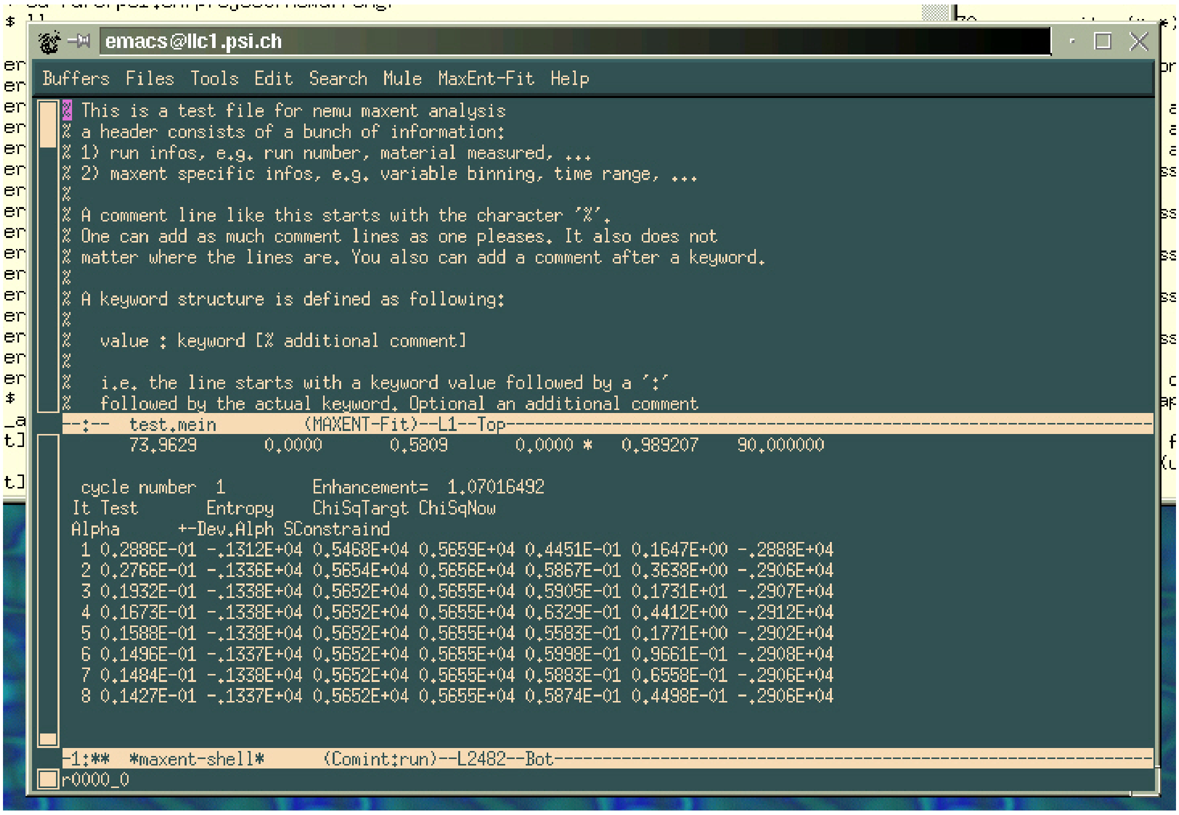
To view the fit–result, you should use the command 'View Fit'. It will read the .hbook–file and plot them into a separate window. There will be again the MaxEnt shell window poping up (splitted emacs window) and at the very bottom there is the question whether you want to print this window. This is not very elegant, however for the moment it was the most easy solution. A typical output–file is shown in Fig. 3.

I guess all the other commands are straight forward so that I do not want to spend too much time in going into them.

3 Installation Procedure

3.1 Already installed on the system, but you are a new user

If the main routines are installed on the system you are lucky, since you only have to make the following changes (assuming that you are using the bash–shell)



```
% This is a test file for nemu maxent analysis
% a header consists of a bunch of information:
% 1) run infos, e.g. run number, material measured, ...
% 2) maxent specific infos, e.g. variable binning, time range, ...
%
% A comment line like this starts with the character '%'.
% One can add as much comment lines as one pleases. It also does not
% matter where the lines are. You also can add a comment after a keyword.
%
% A keyword structure is defined as following:
%
%   value : keyword [% additional comment]
%
%   i.e. the line starts with a keyword value followed by a ':'
%   followed by the actual keyword. Optional an additional comment
--:--  test.mein         (MAXENT-Fit)--L1--Top-----------------------------------------------------------
      73.9629       0.0000       0.5809       0.0000 *    0.989207    90.000000

  cycle number  1         Enhancement=  1.07016492
 It Test       Entropy    ChiSqTargt ChiSqNow
 Alpha      +-Dev.Alph SConstraind
  1 0.2886E-01 -.1312E+04 0.5468E+04 0.5659E+04 0.4451E-01 0.1647E+00 -.2888E+04
  2 0.2766E-01 -.1336E+04 0.5654E+04 0.5656E+04 0.5867E-01 0.3638E+00 -.2906E+04
  3 0.1932E-01 -.1338E+04 0.5652E+04 0.5655E+04 0.5905E-01 0.1731E+01 -.2907E+04
  4 0.1673E-01 -.1338E+04 0.5652E+04 0.5655E+04 0.6329E-01 0.4412E+00 -.2912E+04
  5 0.1588E-01 -.1338E+04 0.5652E+04 0.5655E+04 0.5583E-01 0.1771E+00 -.2902E+04
  6 0.1496E-01 -.1337E+04 0.5652E+04 0.5655E+04 0.5998E-01 0.9661E-01 -.2908E+04
  7 0.1484E-01 -.1338E+04 0.5652E+04 0.5655E+04 0.5883E-01 0.6558E-01 -.2906E+04
  8 0.1427E-01 -.1337E+04 0.5652E+04 0.5655E+04 0.5874E-01 0.4498E-01 -.2906E+04


-1:**  *maxent-shell*      (Comint:run)--L2482--Bot-----------------------------------------------------
r0000_0
```

Figure 2: *Snap shot during the time the* `MaxEnt-fit` *is running.*

1. In your `bash_profile` the following environment variables have to be set

   ```
   # Systemvariablen fuer MaxEnt
   export MAXENTEXECF=/afs/psi.ch/project/nemu/max_entropy/bin/max0103
   export MAXENTEXECC=/afs/psi.ch/project/nemu/me_shell
   ```

   In your particular case it might look slightly different, however `MAXENTEXECF` holds the command for the `MaxEnt` routine of TR.

2. Your `.emacs`–file has to be modified or you have to make it. Add the following lines:

   ```
   ;;; setting load path for emacs lisp file maxent-mode.elc
   ;;;;;;;;;;;;;;;;;
   (setq load-path (cons "/afs/psi.ch/project/nemu/me_shell/lisp" load-path))
   ;;; autoload maxent-mode with .mein files
   (setq auto-mode-alist (cons '("\\.mein\\'" . maxent-mode) auto-mode-alist)) (autoload 'maxent-mode "maxent-mode")
   ```

3.2 How to install the whole stuff

Will be done at some time in the always fast escaping future ...

4 Used emacs lisp File

Will be done at some time in the always fast escaping future ...

For an `emacs` lisp introduction see Ref. [8].

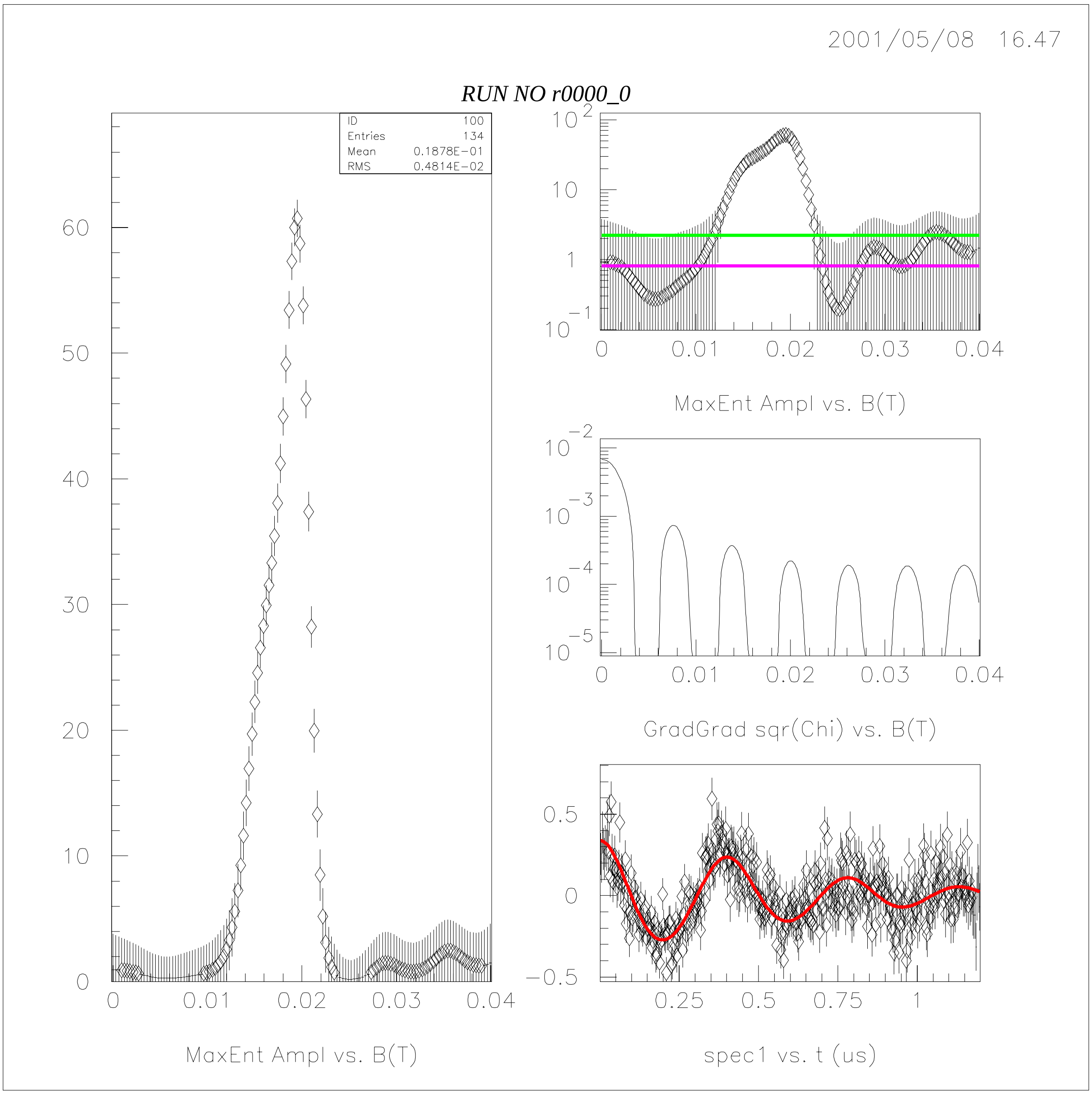


Figure 3: *Output from '`View Fit`'*

5 Used c–Files

There are three programs used to wrap TR MaxEnt , namely

1. me_shell: reads the <mein>–file, generates the MaxEnt input files and starts MaxEnt .
2. me2paw: reads the output of MaxEnt (.mdb file) and produces and updated <mein>–file and .hbook file for paw.
3. me_plot: reads the .hbook file and plots it (HIGZ, HPLOT).

all these programs can be used online, i.e. without the binding into emacs .

5.1 me_shell

This program reads the <mein>–file (<mein> = maximum entropy input). The <mein>–file comprises all the necessary input information for MaxEnt . For the structure and detail description of the <mein>–file see Sec. 2.1. The command syntax of me_shell is:

SYNTAX: me_shell <mein>.

It produces the following output files:

1. musrdataloc.dat: This file contains the information about the data format (e.g. BRA) and the path where to find the data to be analyzed by MaxEnt .
2. vmarker_???.dat: where ??? stands for the selected data format (e.g. BRA). This file is used by MaxEnt to define things like "# of groups", binning range, etc. For a detail description see the documentation of TR MaxEnt .
3. runname.macro: This file is an execution macro for MaxEnt . runname.macro is piped into MaxEnt . This is not a very elegant solution but the only simple one at the moment.

In a last step me_shell executes MaxEnt . MaxEnt itself produces a bunch of output–files (for a detailed description see the documentation of TR MaxEnt):

1. .adb: Actually I don't know what this file is really being used/needed for.
2. .mdb: This is the essential file which contains the MaxEnt spectrum, all the output–parameter settings (e.g. χ^2, $\sqrt{M_2}$), $\nabla\nabla\chi^2$ vs. field, etc.
3. spec0?.dat: These files contain the time spectra of MaxEnt analysis, i.e. the data (rebinned according to the settings in the <mein>–file) and the time evolution of the MaxEnt spectrum (iFFT).

5.2 me2paw

5.3 me_plot

Will be done at some time in the always fast escaping future ...

A <mein>–default file

Hereafter a default <mein>–file is given. There are still a lot of silly comments like

```
% >>!!explain that stuff!!<<
```

which just shows that there is still some improvement necessary.

```
% This is a test file for nemu maxent analysis
% a header consists of a bunch of information:
% 1) run infos, e.g. run number, material measured, ...
% 2) maxent specific infos, e.g. variable binning, time range, ...
%
% A comment line like this starts with the character '%'.
% One can add as much comment lines as one pleases. It also does not
% matter where the lines are. You also can add a comment after a keyword.
%
% A keyword structure is defined as following:
%
%   value : keyword [% additional comment]
%
%   i.e. the line starts with a keyword value followed by a ':'
%   followed by the actual keyword. Optional an additional comment
%   can be added.
%
%   e.g. the run number info line will look like this:
%
%   r2123_9 : run % This was the ultimative measurement.
%
% The keywords ARE case sensitive, so do not mess around with them!
%
% The file is further structured in different parts:
%
% 0) Part RUN DESCRIPTION IN WORDS
%   Here all the stuff, describing specially the run itself, is added in
%   a bunch of comment lines.
%
% 1) Part RUN PARAMETERS
%   These are infos which describe the specific run without being
%   necessary for the actual maxent analysis.
%
% 2) Part MAXENT INPUT PARAMETERS
%   These are input parameters for the maxent algorithm written by
%   Tanya Riseman.
%
% 3) Part MAXENT OUTPUT PARAMETERS
%   These are output parameters from the maxent algorithm like
%       chi^2 etc.
%
% RUN DESCRIPTION IN WORDS ---------------------------------------------
%   In this run we ...
%   0.001    : time resolution  % absolute time resolution in (us) is part of the header of the nemu-file
%   213      : temp                  % temperature in (K)
%   26454894 : total number events   % all groups together
%
%
% RUN PARAMETERS ------------------------------------------------------
%   This are infos which describe the specific run without being
%   necessary for the actual maxent analysis.

BRA  : default data type        % available types are (BRA, PAW, ...)

% where are the data?
/afs/psi.ch/user/s/suter_a/MaxEnt/ : data source directory

2001-04-26, 12-24 : date & time r0000_0             : run

% MAXENT INPUT PARAMETERS ---------------------------------------------
%   This are infos needed by Tanya Riseman maxent routine.
```

```
4 : total number of groups           % i.e. the max. # detectors possible in an experiment

NONE : remove groups                 % e.g. NONE:  means keep all
                                     %     1, 3: means number 1 and 3 are removed

Y    : phases fixed                  % Y=Yes, N=No

(0.0, -90.0, 180.0, 90.0) : phase values % phases of the different groups,
                                         % if ': phases fixed' = N
                                         %   these phases are going to be 'optimized' by maxent.

(0, 0, 12000) : data bins group1   % syntax: it0, start, end
                                   %   every group must have such an entry (even if it is NOT used!)
(0, 0, 12000) : data bins group2   % syntax: it0, start, end
(0, 0, 12000) : data bins group3   % syntax: it0, start, end
(0, 0, 12000) : data bins group4   % syntax: it0, start, end

(0,0),(0,0),(0,0),(0,0) : bins used to estimate bkg % e.g.: (100,233),(102,235),...
                                    %  if both values are equal, e.g. (0,0)
                                    %  the background will be fitted!

(0.0, 4.1, 0.002) : time window for fit  % syntax: start, end, step-resolution
                                     %   e.g.: (0.0, 3.5, 0.005) all given in (us)

(20.0, 2.0, 100) : var time binning  % syntax: start-time, power, width
                                     %   e.g.: (20.0, 2.0, 100) all given in (us)
                                     %   >>!!explain that stuff!!<<

Y    : time binning convolution      % Y=Yes, N=No
                                     %   >>!!explain that stuff!!<<

0.0  : bkg signal asymmetry          % background signal asymmetry

0.0  : sigma apodization             % time in (us)
                                     %   >>!!explain that stuff!!<<

1.0  : sigma looseness factor        % sigma -> L sigma, where L >= 1.0

1.0  : twiddle factor                % should be almost always =1.0,
                                     %   if you don't know what it is do NOT touch!

(0.0, 0.04) : field range            % syntax: start end
                                     %   e.g. (0.0, 0.004) values given in (T)

13   : FFT power n                   % i.e. 2^n points used in FFT (n<17)

Y    : use sqrt(theory) errors       % Y=Yes, N=No

Y    : use final fitting enhancement factor % >>!!explain that stuff!!<<

(0.001/####) : termination level     % >>!!explain that stuff!!<<


% MAXENT OUTPUT PARAMETERS -------------------------------------------
%   This are output values from the maxent routines.
%   They only do have sensible values AFTER a run.

#### : number of fited events
#### : default level        % >>!!explain that stuff!!<<

#### : noise                % noise level estimated from the data

#### : normalized Chi^2
#### : used enhancement     % >>!!explain that stuff!!<<

#### : averaged asymmetry   % >>!!explain that stuff!!<<

#### : mean value #### : sqrt of M2
#### : alpha skew
```

References

[1] CERN Program Library Office, HBOOK, Ver. 4.21, 1993.

[2] CERN Program Library Office, HIGZ, 1993.

[3] CERN Program Library Office, HPLOT, 1993.

[4] J. Skilling and R.K. Bryan, Mon. Not. R. astr. Soc. **211**, 111 (1984).

[5] J.C. Hoch and A.S. Stern, *NMR – Data Processing* (Wiley, John Wiley & Sons, New York, 1996).

[6] N. Wu, *The Maximum Entropy Method*, Vol. 32 of *Springer Series in Information Science* (Springer, Springer–Verlag Berlin, 1997).

[7] R. Stallman, *GNU emacs manual – 13. Edition, Updated for emacs Ver. 20.3* (Free Software Foundation, Inc., Boston, USA, 1998).

[8] R.J. Chassell, *Programming in Emacs Lisp* (Free Software Foundation, Inc., Boston, USA, 1997).