FABADA

Fitting Algorithm for Bayesian Analysis of DAta

Version 0.3

Authors:

Luis Carlos Pardo
Muriel Rovira Esteva, Sebastian Busch

March 1, 2010

Contents

Introduction

> Is it *really* worth adding a parameter to my model, to fit better the data? If yes, when should I stop?
>
> Come on! that is clearly not fitting data, why is this program fitting stacked?
>
> Which of the models I am fitting is better?
>
> Are there more than a combination of parameters that would result in a "good fitting"?
>
> The fitting scientist

FABADA is a software developed in order to perform Bayesian Data Analysis in a simple way, being therefore applicable in a great number of cases. In a few words the program works "simply" varying randomly the parameters of your fitting model, and accepts those that are compatible with your data and its experimental error. In this way the probability distributions functions (PDF's) of the model parameters are obtained. In order to perform model selection, the program also calculates the probability distribution functions associated both to the likelihood and the classical chi-square figure of merit. You can find more information in the following paper arXiv:0907.3711v3 [physics.data-an] [1].

The program is written in plain foratran77, is an open code so that you can add any function you want, and works both for Linux and Windows systems. The graphics are generated from a data file, and then simply displayed using gnuplot (so, it is not "integrated" in the program, and therefore you may always have access to *all* data generated by the program). If you find this software interesting and you use it for your own research please cite this paper arXiv:0907.3711v3 [physics.data-an] [1], and write one of the authors an e.mail with the reference. It would be nice to know that so much effort was not wasted time.

This software can be found at: http://gcm.upc.edu/members/luis-carlos/bayesiano

Part I.

FABADA Usage

Usage

1. Introduction

What FABADA does is to generate in every step a new set of parameters randomly. These parameters will be accepted or rejected if they are consistent with the experimental data and its error (to know *exactly* how it works, please do read **??**). This way of fiting/analyzing data has two advantages:

- In the fitting process sets of parameters that better fit the data are always accepted (sets of parameters that maximize the likelihood or minimize χ^2). **BUT** also parameters that do not fit the data so well, but are compatibles with experimental error, are accepted. In other words the program is capable to go "uphill" in the $\chi^2 P_i$ hypersurface. This has the main advantage that the fit is quite robust, since it is not possible that the program gets stuck in local minima.
- In the analysis process, the program allows to obtain the probability distribution functions related to the parameters and to the figure of merit (likelihood or χ^2), allowing to perform model selection based on solid probabilistic grounds.

To control how every parameter is changed after an iteration, they have associated a "jump length", *i.e.* the maximum change allowed for this parameter (we will call this variable *jump* or sometimes *move* since that is the name in the fortran code). The choose of the jump values for each parameter is very important (see **??**): if they are too small you will need a loooot of time to explore parameter space, if they are too large the procedure will jump like crazy between points that are away in the parameter space and therefore a lot of times "stupid suggestions" (parameters not matching at all the experimental data) will be proposed, that will of course not be accepted, wasting again a lot of computing time. Moreover, if the jump values are not correctly adjusted, it can happens that one parameter is almost never changed (it has a too big jump), while other parameter is always changed. To choose a proper value for the jump values we have added to the program an option to calculate these jumps automatically. In the par file the parameters concerning the jump options will be explained, and also in the first tutorial example this (important) topic will be addressed.

In a standard fitting process you would therefore first to create, or recycle, a parameter file, where all the information is contained. Then you would run the program interactively until the generated curve nicely fits the data. Once the agreement is "nice" you would then proceed to put numbers to the word "nice": you would obtain the PDF for all parameters and your figure of merit. Of course you can also use FABADA in a

frequentist way, and stop the process once you have obtained a "nice" fitting of data. This topic will be addressed in the chapter 2.1, and also in the first tutorial.

2. Using FABADA

To run FABADA you can type FABADA to execute the program, and then it will ask for the parameter file where all information concerning the fitting is. You can alternatively type "FABADA parameter.par" (parameter.par is whatever name of your parameter file), and then the program will start using the proposed parameter file.

There are two ways of working with FABADA:

- you can run it and when the fitting process is finished the program simply ends generating all output files.
- or you can run it activating an interactive option (see 2.1). When the fitting process is over you can plot results and change the parameter file. This makes the fitting process much faster than working with an external plotting software and editor!

In any of the two cases, once you start to fit, FABADA will show in the screen a series of valuable information concerning the fitting:

- First of all the parameter file will be shown as the program reads it. That means that if the program has any problem you can detect where it is quite easy. A warning for windows users: if you execute FABADA clicking the executable, the window will close immediately if any error occurs, and so you will not be able see where the program crashes. It is a better option to open a DOS window, and then execute the program from there.
- Then the program will write the first five lines of the data (and the convolution function if the option is activated). If you do not have a column with a convolution function together with the data, the program will assign to this value m=666.6. The program will also write the number of points of your file, and the first and last point (again quite useful to detect errors).
- In next lines the number of step (written every nwrite steps, see description of parameter file), followed by the time needed to end the fitting, and the values of chi^2 for all functions together, and for each function will be shown.
- In the next line is written the percentage of times that a change of parameters was unsuccessful (nochange). This percentage reflects the times that a proposed set of parameters is "stupid",i.e. implies a too big increase of χ^2). Then it is shown the times that the proposed parameter sets made chi^2 go down (chidown), and go up (chiup). If you are at the beginning of the fitting you should have mostly no changes, and changes that make χ^2 go down. In the analysis process changes that make chi go down and up should be circa the same.

If the program is doing nothing (nochange=100%), you may have quite oft one of the following problems: whether the jump values for your parameters are extremely big, or you are stacked in a local minimum. To avoid the first problem change the jump parameter (obvious) using the "reset" option explained in the next chapter. If you are stacked in a local minima just increase the "temperature" of your fitting. Of course the problem may come for any other reason, but those are the most common.

- Between the first time and the second time the program shows , in addition to the aforementioned information, the real estimated time for the whole process. This time is "real" in the sense that is measured having into account the first two steps of the fitting.

2.1. List of commands

Once the fitting is over, if you have the interactive option switched on, you can use one of the following commands (you can obtain them also by typing h or help in the program prompt):

Help options

`hf` available functions for fitting

`hp` plt files to plot the figures. Usefull only if you are brave and want to change them.

`h or help` shows this list

Plot/Show options

`print` the default option is to show the plots in the screen. However if you type print, the program will generate an *.eps file to export figures. Type again print if you want to change again to see plots in the screen

`ps or p1` plot function 1, i.e. the function in the first line (see parameter file)

`pg or p2` plot function 2, i.e. the function in the first line (see parameter file)

`pi` plot function i (in i line of parameter file)

`pchi` plot the values of χ^2 as a function of the iteration step: nice to get a hint if the fitting is going OK

`pc` plot contours $\chi^2 P_i, P_j$

`pcs` plot contours and its surface $\chi^2 P_i, P_j$

`pparchi` plot χ^2 of a parameter

`pgpar or 1` plot the probability distribution of a parameter

`pcg` plot probability contours

`pcgs` plot probability contours and its surface

`pki` plot each peak for function i defined at the end of the parameter file

`gnn or nn` plot with linear axis

`glx or ln` plot log. scale for X

`gly or nl` plot log. scale for Y

`glxy or ll` plot log. scale for X and Y

`par` see parameters

`showd` see all distances, when fitting the intramolecular structure of molecules. Unfortunately this option MUST be activated changing the fortran code!!!

`showpmin` see parameters giving the minimum chisquare,i.e. the maximum likelihood.

`showmol` generates the molecule to be seen with rasmol, if you are fitting the intramolecular structure of a molecule.

`showchi` shows the calculated χ^2 for every function

Fitting options

`c` continue fitting (more iterations for the fitting process)

`cp` load par file and continue fitting. Will read the par file and start fitting. All previous results are lost!

`e` edit parameter file

`reset` set the jump values all "equal". If option r (relative) is chosen: they will be set as a % of the value of the parameter. If they are absolute (a), the given number will be the jump value. If set to zero NO parameter will be changed (of course!). This option is interesting when you are away from the best fitting. Then the automatic option to set parameter jump sometimes get nuts since we are not in a minimum of the parameter space! Then you have to reset the values to a reasonable value.

`end,fin` end program

FABADA receipt to cook a nice FABADA, ho!

Calculation options

`cc` calculate contours

`cgparpar` calculate contours using the Markov chain. Therefore you must have a Markov chain, i.e. saving the parameters obtained in each iteration of the program. This is done activating the option in the par file (be cautious, such files are enoooormous!).

`covar` calculate sqrt(covar) matrix from the Markov Chain

Parameter File

The parameter file is used to control the fitting process, as well as the analysis process. This parameter file is at the very heart of FABADA, and rules the way it manages the files, and the way parameters are changed. It also has some useful options as to set parameters equal, or convolute with a function. Some options are still in development, and so it is warned: use them at your own risk ;-).

And that is the description of the parameter file:

`Line 1:  title interactive plt nfunc`

`title` [character*15]
: is the name of the file without any consequences, but nice to know what are you working with ;-)

`interactive` [0/1]
: `0`: ends program after running par file.
`1`: asks for further instructions. Activates the dialog option

`plt` [0/1]
: 1: makes standard graphs.
`0`: allows fine-tuning of graphs. To know which plt file corresponds to which kind of graph see the appendix, or type "hp" when running FABADA. This option seems to work, but is not fully tested...

`nfunc` [$0 < nfunc \leq 20$]
: Number of data sets to be fitted within this par file: allows multi-dataset fitting.

`Line 2:  description line:  nomin nomout log Nint xmin xmax fitting option`
`Line 3:  nomin nomout log Nint xmin xmax fitting option`

`nomin` [character*40]
: *x-y-dy* text file with the data points. You can provide the errors in this file for each point *dy*, or you can set the errors as fixed with a value (see next lines!), in this case data set is only a text file *x-y*, or if it contains errors they will not be used.

`log` [`n` / `l` / `cf`]/ `c`]
: n Take the values of data as they are: do nothing (common option!). If this option is chosen, *the following line must be deleted* (that is the line with the convolution options).
`l` if you want to transform your data to logarithmic scale (not fully tested but

seems to work).
`cf` convolves the fitting function with the data set specified in the following line.
`c` convolves the fitting function with the data contained in the third line of the input data file. If this option is used your function should contain x=0 to indicate the program where is the starting point of the convolution function. This option was originaly designed for fitting molecular structures.

`Nint` [`Nint`ϵ $\mathbb{N}$]
if set to `0` program will do nothing about that.
If not set to zero $\neq 0$, the number of points you want your functions to be interpolate (nice option to speed up the beginning of a fitting process.)

`nomout` [character*40]
File only with the output functions. It has not all information (see output file), but it is very easy to introduce in standard software...

`xmin` (ϵ $\mathbb{R}$)
minimal x value that is used for fitting

`xmax` (ϵ $\mathbb{R}$)
maximal x value that is used for fitting

`fitting option` (ϵ $\mathbb{N}$)
number of the fitting function. `0` means no fitting, and will not be had into account for calculating χ^2 (in fact even the input data file is not going to be read!)

`Line 4: nameconv xmax zero. This line is only written if the convolution option is activated (log=cf in the previous line)`

`nameconv` [character*15]
file to perform convolution, will be called resolution further on (in the case of neutrons would be Vanadium data)

`xmax` (ϵ $\mathbb{R}$)
data will be convoluted from -Xmax to Xmax. If using vanadium for QENS analysis, be sure that your resolution is zero above and below this value!!!

`zero` (ϵ $\mathbb{R}$)
Where is the 0 of your resolution? if set to negative value, the maximum of the resolution function will be chosen.

AGAIN: do not write this line if you do not make convolution!!!!! (did we warn you already??)

`Line 5: nomout nomchisq`

`nomout` [character*40]
output filename. it will write EVERYTHING (last parameter values, bet fit parameter values, final χ^2, reduce χ^2, and a lot of useful information.

`nomchisq` [character*40]
It will write the value of $/chi^2$, and the acceptance ratio for each parameter. If the program is nicely working, at the end of the fitting *the acceptance rates for ALL parameters should be equal* (more or less).

`Line 6: Temperature`

`Temperature` [ϵ $\mathbb{R}$]
This parameter "modifies" the experimental errors in this way: error=Temperature*dy, where dy is the data error, therefore: if set to zero, the minimum of χ^2 will be found. In this case the program works (unfortunately) in a frequentist way. If set to T=1.0: errors will be had into account*i.e.* the program will admit set of parameters that increase χ^2 within the error of the data. If set to a big value (100.0) program will fit poorly, but will explore a lot of the parameter space... nice option at the beginning of a fit, if you have no idea about the initial values for you parameters!

`Line 7: Sigmaexp(i) (as many as nfunc!)`

`Sigmaexp(i)` should be set to 0.0, if errors are included in the data file. Should be set to any other value if all the errors of your data are the same. (of course you can also set it to 0 and write the same number in the third line of your data file!). If it is set to a negative values then the probability distribution function of the experimental points around the expected values is set to a Poisson PDF.

`Line 7: Npar`

`Npar` $1 < Npar \leq 100$ number of parameters of the fitting functions (also called models or hypothesis).

`Line 8: ncic nwrite`

`ncic` How many cycles the program will do.

`nwrite` Each how many cycles you want the program to write output files, and to actualize the jump of the parameters if the automatic option is activated.

`Line 10: varpar gpar incgpar incgchi notused`

`varpar` [`0`/`1`]
If set to `0` parameters will be changed one by one: this should be the standard option!!.

If set to `1` all parameters will be changed at the same time. The last option could seem to be faster, but it is only at the very beginning of the fitting process, afterwards is very difficult that ALL parameters go into the direction of a better fitting. Moreover all the Bayesian analysis crashes with this option. Only use if you really know what you are doing ;-).

`gpar` [`0`/`1`/`2`/`3`]
0: to not calculate PDF of parameters.
1: do calculate PDF.
2: Do calculate PDF, and write each step the parameters resulting of the fitting process (the program created then a huge file MMchain.dat depending on the number of iterations).
3: Do the same as before but ONLY write in MMchain.dat the parameters when all output files files are saved (number of steps defined in the previous line)

`incgpar` Size of the PDF boxes: they will be set to jump÷incgpar.

`incgchi` Binning of χ^2 (usually 0.1 is reasonable, but have a look at your values of χ^2!!!)

`notused` Archeoparameter: is no longer used, but kept to make old parameter files working... and who knows, maybe it'll be used in the future

`Line 11: annealing`

`annealing` [`0`/`1`]
annealing: This option changes the temperature of the fitting, and therefore allow the fitting process to relax to the best parameter combination, when parameters are poorly initialized. This option is designed for complicated functions with a lot of relative minima. Unfortunately has not been extensively used, and therefore is not fully tested.

`Line 12: step Tin Tfn incT`

`step` Number of steps to change temperature

`Tin` Initial temperature of the fitting

`Tfn` Final temperature of the fitting

`incT` Step of the fitting (together with the number of steps to change temperature sets the ramp for the annealing)

`Line 13: convergence factor (cfc)`

`cfc` Used only when you want to calculate automatically the jump of the parameters. They will be set in such a way that the average acceptance of a proposal of parameter set will be cfc, and that the acceptance for each parameter will be cfc/Npar, in this case.

Now the parameter file starts with the description concerning how to change each parameter. The order of the parameters is not important, but **IT IS** important the first number that tells the program which parameter is it.

`Lines 14 and so on: orden namepar logpar move parmin parmax`

`orden` Number of the parameter. This number identifies the parameter and should never be changed!

`namepar` is only a label, and DOES NOT identify the parameter. but helps quite a lot

`logpar` n: normal. a: automatic, and therefore the convergence factor will be used. The jump will be adjusted with a twofold objective: ALL parameter must have the same acceptance ratio, and this ratio should be equal to cfc/Npar (cfc defined in the former line). al: automatic but parameter jump can not be greater that a percentage of the parameter. l: log scala. Parameters will be changed in log scala, *i.e.* $P_{new} = P_{old} + 10^{jump}$.

`par` value of the parameter

`move` jump, the maximum move of this parameter.

`parmin` minimum parameter value allowed (lower boundary of prior)

`parmax` maximum parameter value allowed (upper boundary of prior)

```
Lines 22: nig (number of equalities between parameters)
nig following lines: ign, parig(1,ign)
```

`ign` number of parameters you want to set equal

`parig(1,ign)` the number of the parameters set to be equal. For example if you want parameters 4,6 and 8 to be equal, you should write 3 4 6 8, *i.e.* I want three parameters to be equal, and they are numbered as 4, 6 and 8 in the par file.

`Lines 23+nig: nrel` Number of relationships between parameters: should always be zero (nig=0) This option is not fully tested, and not to be used, unless you know exactly what you are doing.

```
Lines 23+nig+nrel: ncont (number of contours to be calculated
Ncont follwing lines: contpari contparj namecont
```

`Nc` Number of contour plots to be plotted

`Pari Parj namecontour Pari Parj`: number of the parameters to calculate the contour plot. `namecontour` Name of the contour plot file

`asd` min max min max step step

`ctminx ctmaxx ctminy ctmaxy ctdx ctdy`

`ctminx` minimum value of X for contour plot

`ctmaxx` maximum value of X for contour plot

`ctminy` minimum value of Y for contour plot

`ctmaxy` maximum value of Y for contour plot

`ctdx` increment of X for contour plot

`ctdy` increment of Y for contour plot

`npeaks`

If you want to plot separately different components of your model (for example each peak separately if you have more than one), that is your option. npeaks should be set equal to the number of components. Otherwise write 0

`nparpeak`

number of parameters you want to force to be zero, or any other value. The other values will be set to that of the better fitting. If you type "0" (that is to say a peak with zero parameters) the program will assign the resolution to this peak.

`parpeak,valpeak`

`parpeak` number of parameter

`valpeak` value you want to force the parameter to be.

Output Files

3. Out file

name=user defined, saved always every nwrite The output file is thought to condense **all** information needed to reproduce a fitting, and the last results obtained with this fitting.

- The first part of the file is simply the parameter file (again). Parameters are those of the last step of the fitting process.
- χ^2 (just in case: $\chi^2 = \sum^{i=Ndat} \frac{y_i^{calc} - y_i^{obs}{}^2}{\sigma_i{}^2}$ obtained for each fitted functions (when performing a multifit) followed by the χ^2 obtained for all functions together. If a function is NOT fitted, then $\chi^2 = 0$
- Parameters that give the best fitting.
- a series of useful values:
- -log(evidence): The evidence, when we use a χ^2-like fitting is: $evidence = \exp -\chi^2_{mean}/2$. χ^2_{mean} is obtained making the mean value of χ^2 using the obtained PDF.
- -log(Lmax): as before but using the minimum of χ^2 $Lmax = \exp -\chi^2_{min}/2$
- log(evidence) AIC (-2ln Lmax + 2k): this is a frequentist approximation to the evidence
- log(evidence) BIC (-2ln Lmax + k ln N): this is another frequentist approximation to the evidence
- Bayesian complexity ($\chi^2_{min} - \chi^2_{atPDFmaximum}$ The Bayesian complexity should be equal to the number of parameters that you are actually fitted. For example if your model has three parameters, and you obtain a Bayesian Complexity of 2.5 (for example) your parameters are correlated. If you get a Bayesian Complexity greater than the number of your parameters, your model has too many information for so little data.
- Mean value of parameters, and parameters at the maximum PDF. They should be almost the same if you have a Gaussian like PDF (better: it is not symmetric around the mean values). Otherwise: *do have a look* at your PDF's: things are not as simple as you thought with your model! (=PDF is not Gaussian probably)

- The same information as with your output files: data and fitting. It is redundant, but it might be veeeery helpful when recovering the fitting you performed last week.

4. Out data file

name=user defined, saved always every nwrite

First line Names of variables. They start with #, since that is what gnuplot likes to plot data.

next lines Simply the values of x you input function (ydat) the fitted function (ycalc) using the values of the parameters giving the best fit, and the difference between calculated and experiential data (diff).

5. Chi file

name=user defined, saved always every nwrite The description of the file is done by columns, not by lines. You will get the columns:
Cicle chitot nochange chidown chiup P1 P2 P3... Pn

First column: cicle Number of steps.

Second column: chitot The value of the total χ^2 added for all fitted functions.

Third column: nochange Percentage of how many times parameters were not cahnged

Fourth column: chidown Percentage of how many times a change of parameter made the χ^2 value go down

Fifth column: chiup Percentage of how many times a change of parameter made the χ^2 value go up. Of course this value is zero if the temperature is set to zero (=only changes in parameters that decrease χ^2 are accepted.

Sixth column and next: P1,P2... Percentage of how many times a parameter is changed (giving as a result a decrease or a increase of χ^2. In a good fitting the values for all parameter should be equal among them, and equal to (chidown+chiup)/(Npar).

LAST columns The values of χ^2 for each function. Makes sense only if more than one function is fitted.

6. Move file

name=move.dat, saved always every nwrite Again, this file is written to control if the fitting is developing correctly. It writes the values for the jump of parameters as a function of the step. Together with the previous file helps to solve problems when

fitting functions with a lot of parameters, or that have more than one minimum in the parameter space.

First column: cicle Number of steps.

Second column and next: P1,P2... The value for the jump of each parameter.

7. Initial parameter file

name=parini.par, always saved at the beginning of fitting This is a rescue parameter file when you perform a fitting, and you want to get back to the initial parameter file you used. It will always be saved at the very beginning of the fitting with the initial parameters.

8. Initial fitting file

name=inicif#.par, always saved at the beginning of fitting This file saves the function generated at the beginning of the fitting process to know at a fast glance if the fitting is going OK.

9. Initial convolution file

name=convf.dat, always saved at the beginning of fitting This is a control file (thought mainly to check errors) where the convolution file is written, from -xmaxvonv, to maxconv. The zero of the function should coincide with the maximum in the case the resolution is a peak, if not, please do check your definition of zero in the parameter file!

10. Parameter sets from fitting process

name=MMchain.dat, saved only if gpar is set to 2 or 3 WARNING: THIS FILE CAN BE HUGE!

This file is only saved if the option is activated in the parameter file (see line 10 of par file. Variable gpar). This file contains the values of all parameters obtained by the fitting process (if gpar=2 every step, if gpar=3 every nwrite steps). This file is very important to get a fast feeling if parameters are correlated (simply plot a parameter as a function of another). TO know if parameters are "frozen" (if they are not changing), by plotting the value of the parameter as function of the step.

This file is also necessary if you want to calculate the covariant matrix.

First column: cicle Number of steps.

Second column and next: P1,P2... The value for the jump of each parameter.

11. File with parameter information from PDF's

Calculated when program is finished, or when you calculate the PDF'd typing cgpar

First lines Calculated values of χ^2, averaged and the last one of the fitting process

line: Par Nchange Parminchisq, media, highprob, errdown,errup,errmed,diff,reldiff

Par name of parameter

Nchange Number times this parameter was changed

Parminchisq Parameter giving the minimum χ^2 value

media Media calculated from PDF for the parameter.

highprob Highest probability calculated form PDF (this last three values should be quite similar. Other wise check your PDF)

errdown Calculates the lower error bar as follows: From the highest probability point P_{max} the program will decrease the parameter and calculating the accumulated probability. Since error are usually defined by frequentist inside a 68% probability, when the cumulated probability is 0.34 (0.68/2), the process will stop giving that value as $Perrdown$. erdown will then be calculated as errdown=$P_{max} - Perrdown$:

$$0.34 = \int_{P_{errdown}}^{P_{max}} PDF(P)dp \tag{1}$$

errup The same as before, but increasing the parameter.

erred (errup+errdown)/2. Assumes both errup and down are equal and calculates the mean value. This is the error bar that will be written in the latex file.

diff Pmed-Pmax: should be close to zero if the PDF is symmetrical. Otherwise check waht happen to your PDF.

reldiff $vr = (p_{med} - P_{max})/(p_{med} + p_{max})$ to check very fast which PDF's are not symmetrical

12. Latex file with parameter and errors

Calculated when program is finished, or when you calculate the PDF'd typing cgpar This file generates a latex code to include a table in a paper that you surely will be written after analyzing your data with FABADA. This table contain the value 66.6 to eventually put the value calculated by other means (for example), followed by the highest probable value and the (-sic-) frequentist error calculated as aforementioned. If you can Fortran, go to gpar.for subroutine and modify it to have the table in your preferred format.

Adding a new Fit Function

You should dig into the code to add a new function. In fact we call it "function" but it can also be linked to external files (for example if you have a parameterized model) or to an external process (such as in the case of calculating dielectric spectra form the distribution of relaxation times, see the program `FIDEWA`, linked now to FABADA also in this web page). Adding a new function consists of three and a half steps:

- add the function (or whatever) itself in simplefunctions.for
- add the reference and number in functions.for... two times! with and without convolution. If you do not add it with the convolution option, it simply will not work if you want to convolute (obvious) using the cf option in the parameter file...
- add a short explanation in help.for
- change this manual to add a complete explanation about your functions does. This is considered as a half step.

Part II.

Tutorial

We recommend the user to start with this simple examples to get use to FABADA.

The simplest case: fitting a Gaussian

Let's start with the simple case of a Gaussian function

$$\frac{A}{\sqrt{2\pi)W}} \cdot \exp - \frac{(X-C)^2}{2W^2} \tag{2}$$

The function was generated using the parameter set $\{P_i\} = \{A, C, W\} = \{10, 5, 2\}$ We will use the following files to start the fitting:

- gauss.dat is the data file without errors. Since the function is generated setting all errors equal we will include them in the par file.
- pg.par is the parameter file
- check also that FABADA and gnuplot are in your directory

13. Starting the fitting, getting the minimum χ^2

executing FABADA We execute FABADA and it asks for the par file, in our case pg.par (you could also type FABADA pg.par, and you will execute the program using the pg.par file). The program starts writing the percentage of times χ^2 stays THE SAME, I goes up or it goes down. At the beginning the program does little (93% of the times). Then the values of the jumps, that are set to automatic (see par file chapter), adjust themselves and the program starts decreasing χ^2. Finally the times it goes up and down is equal: we reached the region of fittings compatible with data and its error. Moreover they both are of about 15%, so a 30% of times the program is changing χ^2. This exactly what we told the program to do adjusting the acceptance factor cfc to 0.3 (30%) in the parameter file: everything goes fine!!!

P1: plotting the fitting Let's now have a look at the fitting so write P1 in the prompt, and you will get 1: the fit looks nice (the red curve). The green curve is the curve generated with the starting parameters.

e: editing the parameter file We will now edit the parameter file to make the best fitting possible,i.e. finding the fit that minimizes χ^2 as frequentist do. Type e, you edit the parameter file. Now go to the temperature in the fifth line and set it to zero: you will now allow only changes of parameters that make χ^2 decrease. Go out from the editor (alt+two times down+enter to go fast ;-).

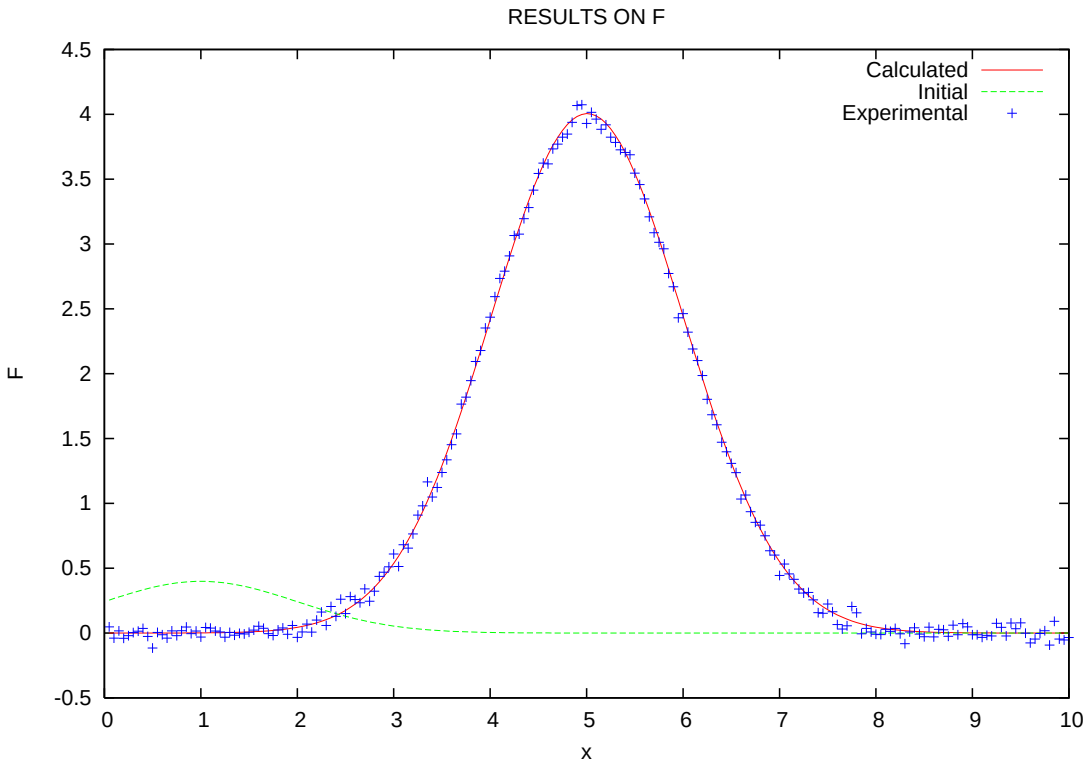

Figure 1.: Fitting to the gaussian. In green you see the starting function, in red the nicely fitted function.

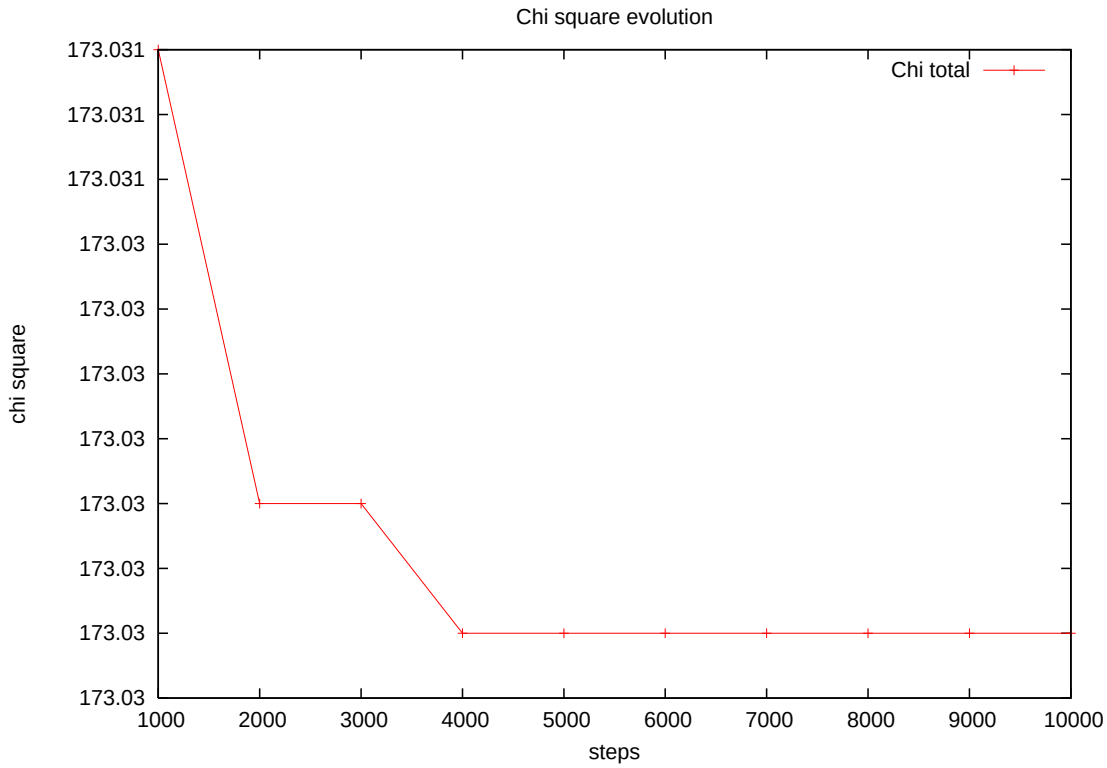

Figure 2.: Evolution of χ^2 as a function of the number of steps: it's going down, it's fitting!

cp: rerun FABADA with a new parameter file We now type cp: this will continue executing FABADA with the new parameter file. Be careful: if you type c you will continue with the old parameter file.

pchi: plotting χ^2 Looking at the prompt we already see that chiup is zero (only changes that makeχ^2 lower will be accepted). Let's see it: type pchi. χ^2 goes down, and then is constant: we reached the minimum χ^2 (see figure 2. You can type p1 to see the change from the last fitting: they are very close (of course!).

14. Analyzing the results

Checking that everything is ok type cp to repeat the fitting (or rerun FABADA with the las pg.par). Now open the chi.dat file and check that all parameters are changed with the same percentage... they should. In order to speed up the analysis of data, we are going to change the acceptance ratio (the 12th line) to 0.66: that will make parameter jumps smaller, and will decrease the number of times that the program do nothing. Rerun FABADA and parameter jumps should be smaller, and in chi.dat and the acceptance of each parameter should be of about 66%/3=22%.

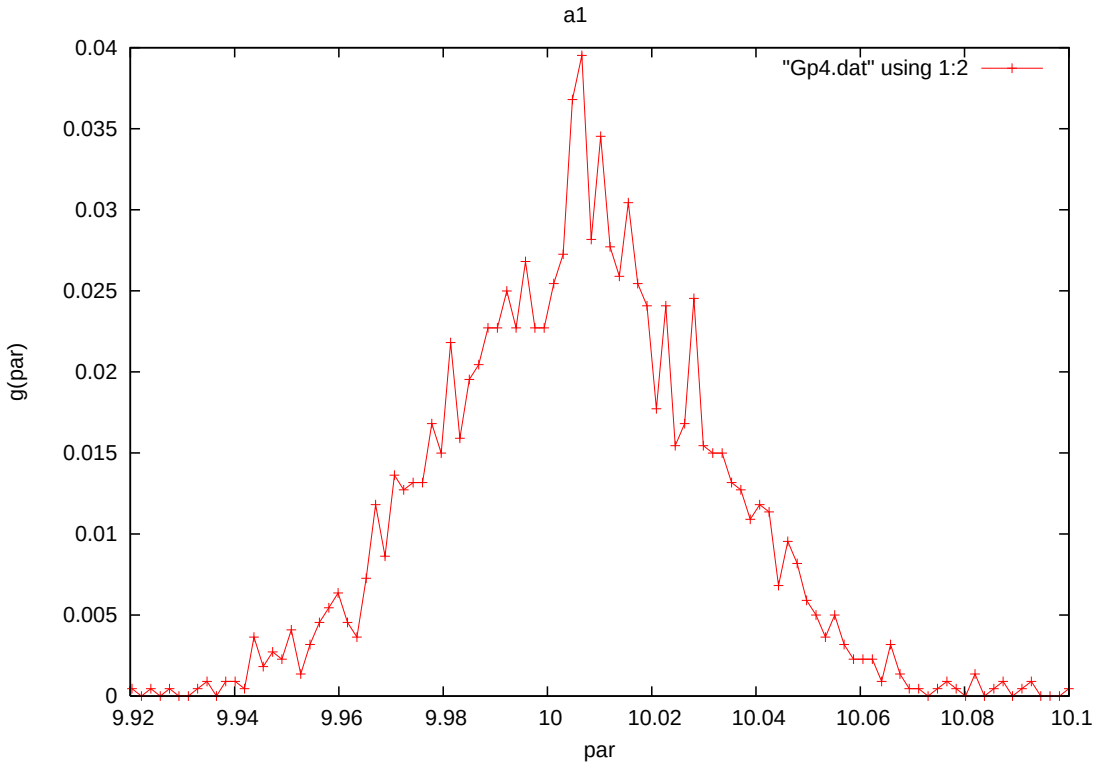

Figure 3.: Probability Distribution function associated to the gaussian fitting for the high of the gaussian (A=parameter 4).

Have a look at parameter PDF Once the program has ended, do calculate the parameter probability distribution functions (PDF) typing cgpar. To see the PDF of a parameter type pgpar and the number of the parameter for example parameter 4 (the height of the Gaussian). you should get a figure like this one **??** (a trick: when you have a lot of parameters is a pain to type every time pgpar... you can type als simply "1". In this way you can have a very fast look at all PDF'S to detect any possible problem) . The PDF seems a gaussian, but quite ugly. Let's repeat the process with more steps, let's say 100000 steps. Now it looks nicer, and takes more time of course (see Fig. 4)

You can in fact have a look at the (frequentist) errors calculated as containing a 68% of probability that the parameters is between some limits, just having a look at what is written when calculating the PDF's. If you do that you will see that parameter 5 has a problem (ERROR out of binning boxes: Nbins=10000)...

Adjusting parameters to get a PDF We must understand now how PDF are generated... In the 9th line of the parameter file you can find the parameters controlling the generation of the PDF's. As explained the first number should be zero (change parameters one by one). The second number it is 1: do calculate PDF's. The

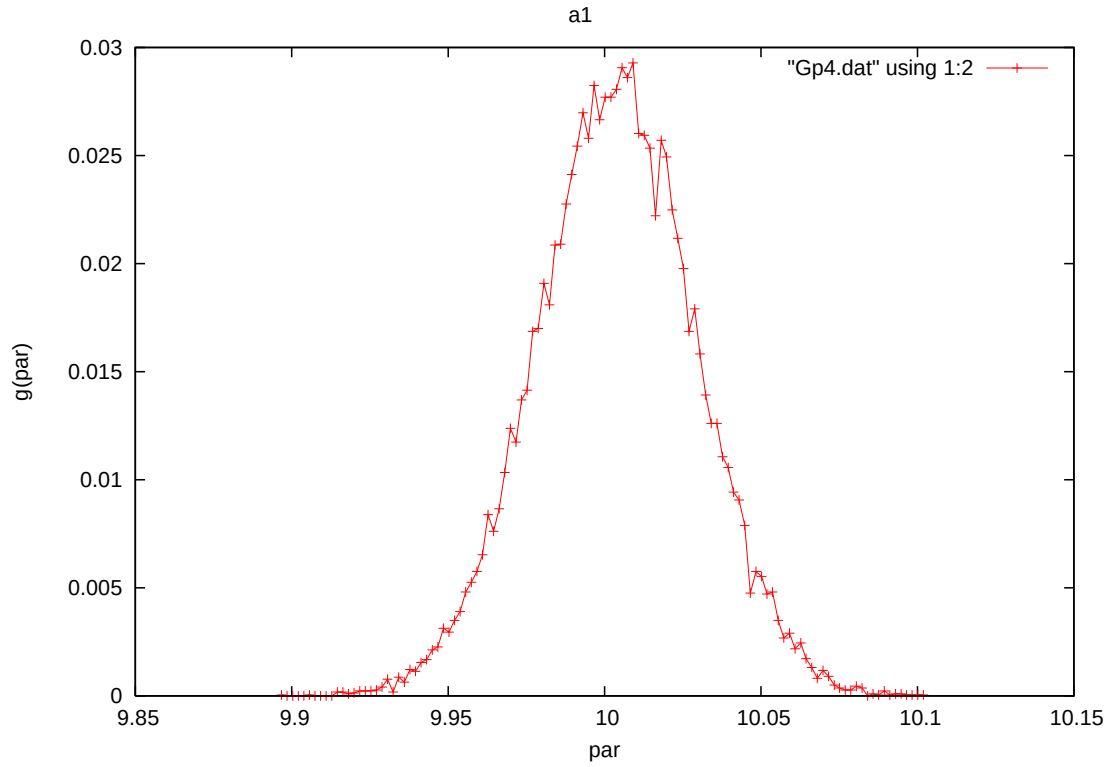

Figure 4.: Probability Distribution function associated to the gaussian fitting for the high of the gaussian (A=parameter 4).

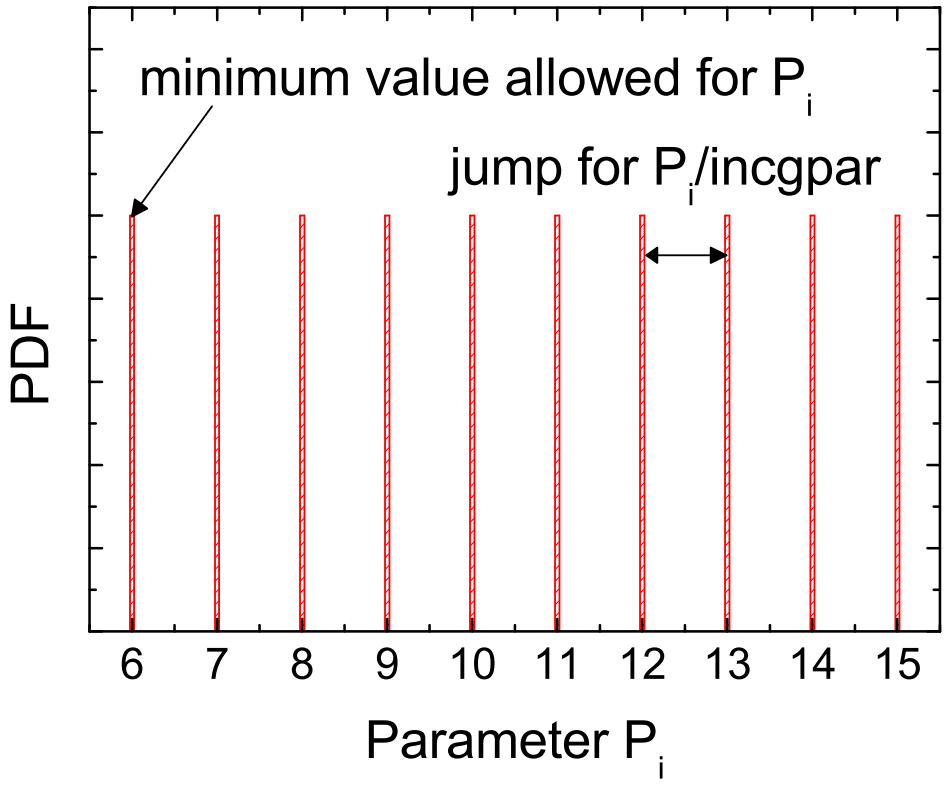

Figure 5.: Way the program set the binning for calculation of parameter PDF's.

third number (incgpar) is the one used to calculate the width of the boxes to bin the parameters generated in the analysis process: they will set to jump/incgpar, so a twentieth of the jump value for each parameter. The first box is set to the minimum value allowed for parameter (see fig. 5). The program allows only a maximum of 10000 boxes, so the error indicates that you do not have enough boxes to calculate the PDF. You have two solutions: you increase the minimum value for parameter, or you decrease incgpar. I would suggest the first option, and change the minimum value for parameter 4 to 4.0. Then type cp to repeat the analysis, and you will get a nice PDF.

Another way to get the parameter PDF You can also simply use the gpar option: as it is said in the explanation of the par file, by setting this value to 2 the program will create a MMchain.dat file with all the values of the parameters (so a quite long file). You can afterwards calculate the PDF's with another program. The

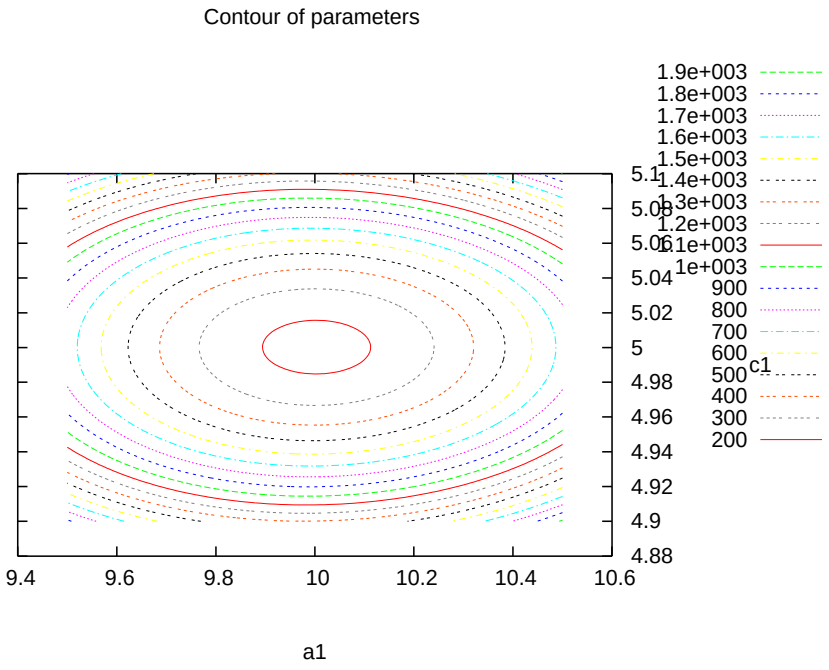

Figure 6.: $\chi^2(A, C)$ contour plot for the fitting of the gaussian.

problems is that you will need to handle with quite long files...

plotting the likelihood You can also plot the PDF associated to χ^2 by plotting "parameter" 0. This is useful to make model selection (see next section).

plotting the parameter space If you want to know if parameters are correlated, you may calculate contour plots of χ^2 as a function of two parameters ($\chi^2(P_i, P_j)$). For example let's calculate the contour plots of $\chi^2(A, C)$ and $\chi^2(A, W)$. You must first adjust the minimum, maximum and step of each parameter, and then type "cc" on the prompt: the program will calculate the functions. If you now want to plot them just type pc to plot the contour in two dimensions (in fig. 6 for $\chi^2(A, C)$) or if you want to see also the surface just type pcs (in fig. 7 for $\chi^2(A, WC)$)

a trick pgpar=1 Because plotting PDF's is usually done fast to see if there is any problem, you can type "1" instead of "pgpar", so only with the numeric keyboard you can have a fast look at all PDF's.

parstat.dat file all information concerning you PDF is also written in the file parstat.dat when you calculate the PDF's (i.e. when you type cgpar)

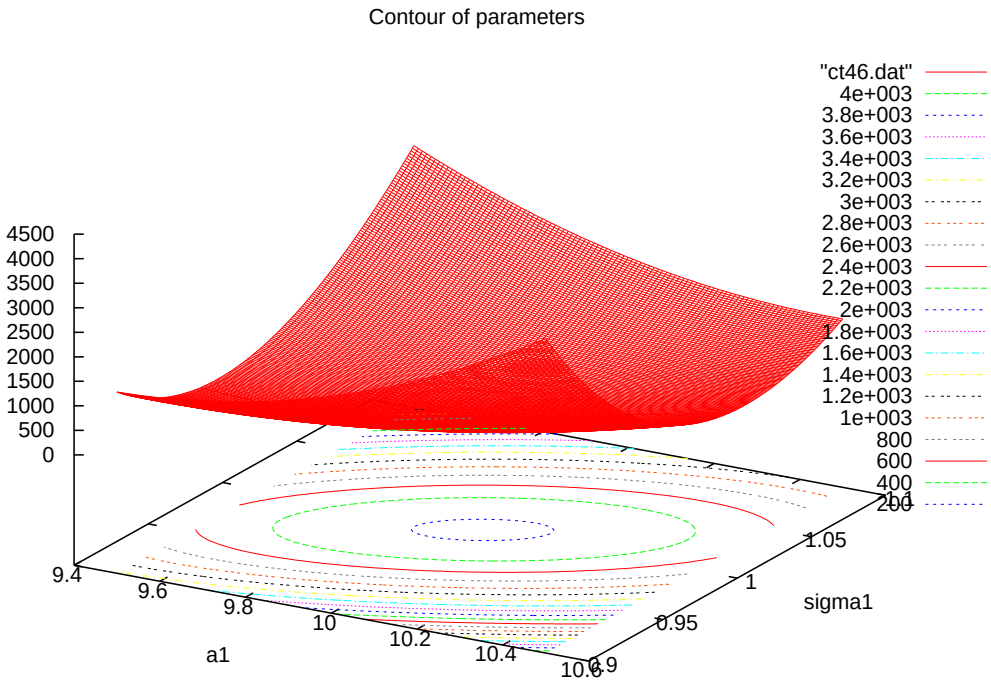

Figure 7.: $\chi^2(A, W)$ contour plot plus surface for the fitting of the gaussian.

A more complicated case: fitting with convolution

We are now going to proceed to perform fittings using the convolution option of FABADA. First of all, to understand what the program does let's remember what is to convolve. Let's see the definition first:

$$Y_{measured}(x) = Y_{exp}(s) * R(s) = \int_{-\infty}^{\infty} Y_{exp}(s)R(x-s)ds \tag{3}$$

In really inexactly and few words, convolution makes each experimental point to be the resolution. Just for fun... if in the former equation you replace the experimental function by a single point (i.e. a dirac's delta), the solution is the resolution.

In this example we are going to fit a Lorentzian function to some data set, convolved by a gaussian-like resolution (this is usual in Quasi Elastic Neutron scattering experiments). We will then try to fit also a gaussian and will perform some model selection. The gaussian function was already introduced in the last section (but to avoid searching that page):

$$\frac{A}{\sqrt{2\pi)W} \cdot \exp - \frac{(X-C)^2}{2W^2}} \tag{4}$$

and the lorentz function is defined by:

$$(A/\pi) * \frac{W}{(X-C)^2 + W^2} \tag{5}$$

15. Starting the fitting

We will use the following files to start the fitting:

- L01.dat is the data file with errors (so in the par files the experimental errors must be set to zero=read errors from file). We need also the function to convolve: resolution.dat.
- plconv.par is the parameter file.
- check also that FABADA and gnuplot are in your directory

To perform the convolution we should take care about a couple of things in the parameter file:

resolution file We should indicate that the resolution is taken from an external function. That is said in the parameter file in the third line: the option CF (convolute with function) is written. So that means in the next line we need the function

parameter file This function is going to be taken from "resolution.dat". The next two numbers are quite important. The first one (Xmax) indicates the program from which X value, up to which X value you want to make the convolution. Since this must be symmetric only a number is required to set the limits. Just be aware that in the case of QENS fittings, if your resolution does not decay to zero (i.e. a number much lower than your experimental data) within (-Xmax,Xmax) you will have numerical problems... just be careful. The next number says where is the zero of your convolution function. If you write a negative number the maximum of the function will be chosen, and that is the case in our example. (Just for fun, put for example 0.2 in this example, and you will see the effect of the convolution)

Let's now proceed with the fitting. First we run FABADA again. And then let's have a look at the final result by plotting the function in logarithmic Y scale. This is done to see how the fitting is for small Y values, just type nl (x scale is normal, and y scale is logarithmic), and type P1 to have a plot of fitting and "experimental" data.

Plotting the resolution together with fitting If you want to plot the resolution in addition to your fitting, then just type "pk1" (plot peaks for function 1). How is it done? Of course the secret is in the parameter file.

In the last line of the parameter file we see the comment "npeaks", this option is used to plot separately the components of your fitting, if it consists of a summation of components. This is done by setting each parameter to a certain value. To see in detail how it works, please have a look at the next section. However, here we want simply to explain how to generate a peak that contains the resolution. For that you only have to set the number of parameters that defines your component to "zero": the program will understand that, in fact, what you want to plot is the resolution. So if you do that, you will get the following figure in semilogarithmic scale (so type nl and afterwards pk1) **??**

16. Analyzing results: model selection

As we can see in figure 9 in linear scale both models are able to nicely fit the data. If we have a look at the logarithmic scale it is seen that in fact the lorentz model fits data better than that Gaussian one, but: is it possible to put that in numbers?

One option is to calculated the reduced chi-square ($\chi^2_\nu = \chi^2/(n-m)$) where n is the number of experimental points, and m is the number of parameters. However, as pointed out in reference [1], it is done assuming two risks: we suppose that parameters are not correlated, and we suppose that there is a single minima in the $\chi^2\{P_i\}$ hypersurface (see the aforementioned work for more details). So what we do is to calculate the values of χ^2 as the program is running, and then we calculate the Probability Distribution Function for χ^2. We have done it in our case. How? We explain it now!

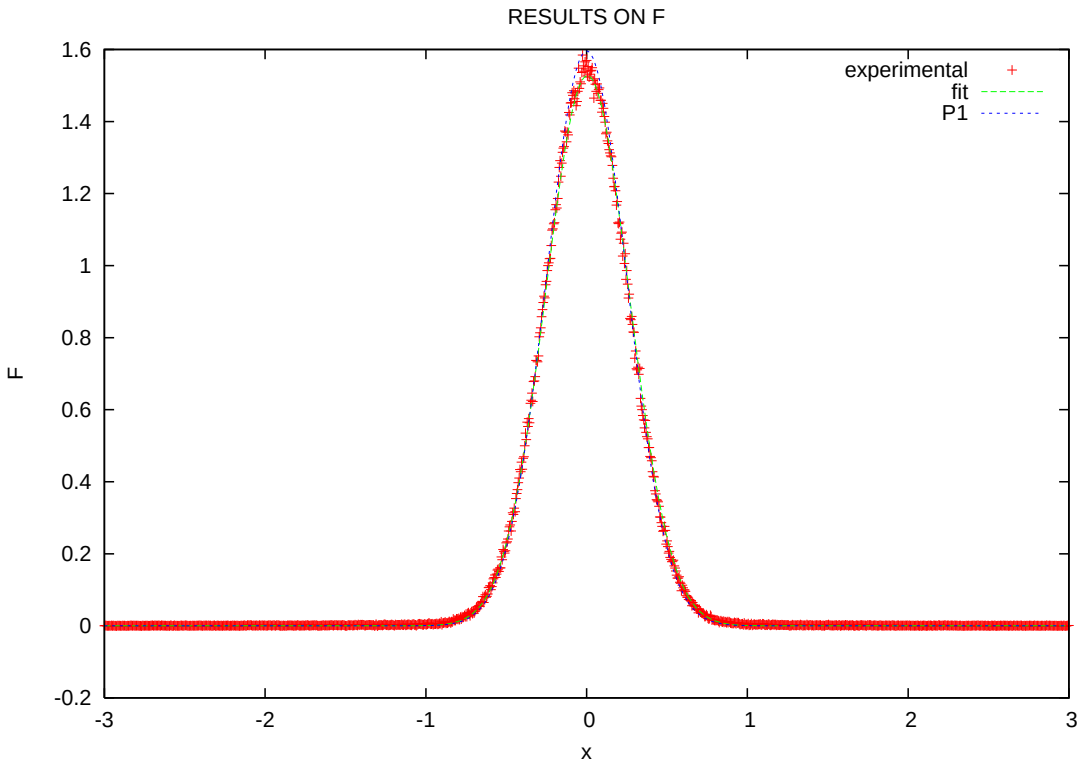

Figure 8.: Experimental data together with the fit and the resolution function generated by using the option "plot peaks, pk" in FABADA

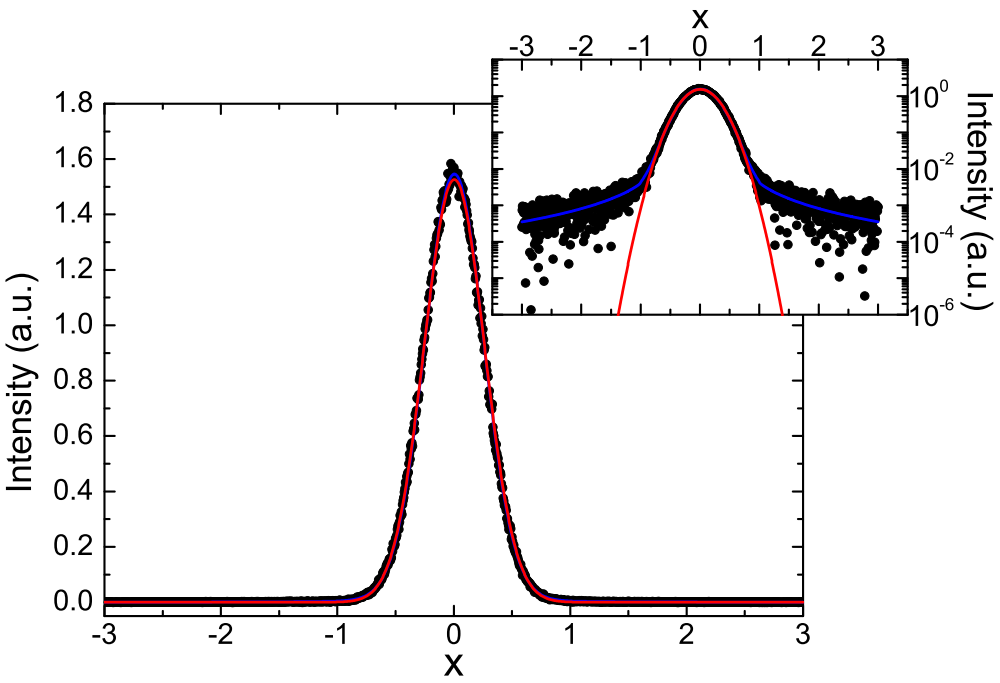

Figure 9.: Two models fitted to the experimental data of the example: a Lorentz function and a Gauss function. The inset shows the same graph in semi logarithmic scale. The question is: taking into account data error, is it possible to prefer one of the two models?, answer in the text.

- we run FABADA once the fitting is correctly done (it looks nice), and we make sure that acceptance for all parameters is the same, so that the jump values are calculated correctly. We take care to choose a value for incgchi (10th line of parameter file) that suites our needs: it is set to 0.2. If it is set too small, then the binning will not catch the calculated χ^2 values, and they will fall out of our bins (try to put 0.001 for example and an error message will appear). We then run the program.

- once we are done we type "cgpar". We immediately see the most probable value for chi^2, the media (sometimes called evidence) and its minimum value ("the frequentist solution").

- Let's now type "pgpar" and "0" to choose χ^2 instead of a parameter (you can also type "0 1" to go faster), and you will see the χ^2 PDF. If you repeat the same process for both models (contained in pgconv.par for gaussian model and plconv.par for lorentz model), you will get two PDF's as those in figure 10.Lorentz model is clearly preferred since *any* combination of parameters will led *always* to a better fit. You can have more complicated cases, for example those in [1] where you can find a multimodal χ^2 PDF, but that's another story and shall be told another time [3]

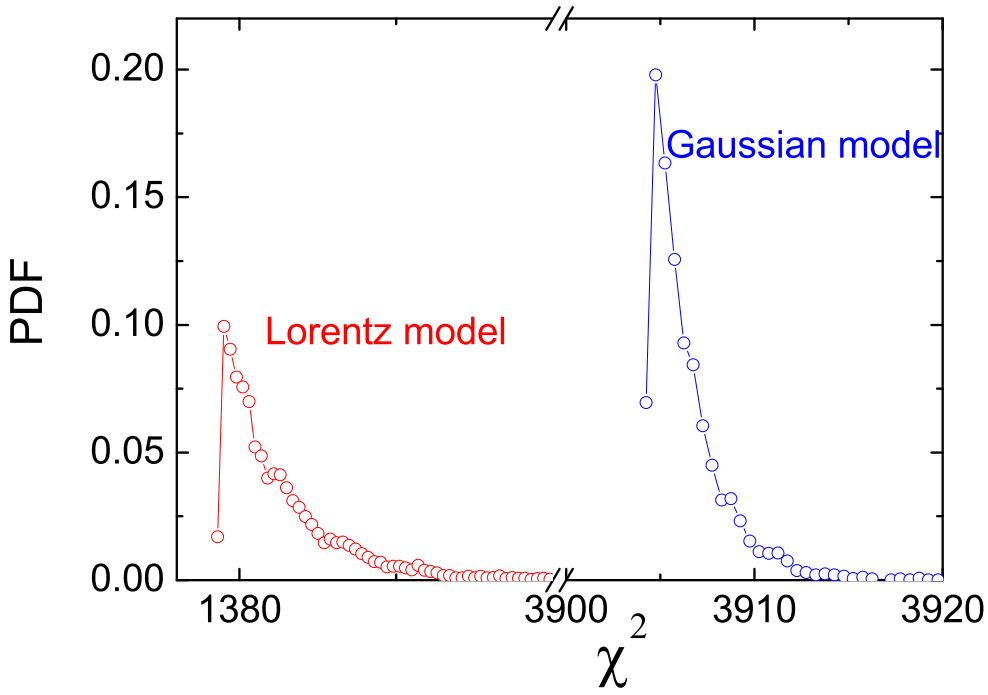

Figure 10.: Probability Distribution Function obtained for the two proposed models, gaussian and Lorentzian. Lorentz model is clearly preferred since *any* combination of parameters will led *always* to a better fit.

Simultaneous fitting of several data sets

Finally let's show an example where multifitting is performed. Let's take the example of dielectric spectra that are measured at different temperatures. At one temperature spectra can be nicely described by the sum of two peaks, one is describe by a function called Havriliak-Negami (HN), and the other is a special case called Cole-Cole. The Havriliak-Negami function is just the imaginary part of the following equation:

$$\varepsilon^*_{HN} = frac1\big(1 + (i\omega\tau_{HN})^{1-\alpha}\big)^{\beta} \tag{6}$$

so, explicitly the HN function to be fitted is the following

$$w = 2\pi\nu\tau_{HN} \tag{7}$$

$$lo = w^{1-\alpha} \tag{8}$$

$$\phi = \arctan\frac{lo\cos(\alpha\pi/2}{lo * \sin(\alpha\pi/2) + 1)} \tag{9}$$

$$ne = (1 + 2lo\sin(\alpha\pi/2) + lo^2)^{\beta/2}) \tag{10}$$

$$\varepsilon''_{HN} = \sin\beta\frac{\phi}{ne} \tag{11}$$

where we have kept the same variable names as it appears in the Fortran code. For the case of the Cole-Cole function, it is the imaginary part of:

$$\varepsilon^*_{HN} = frac1\big(1 + (i\omega\tau_{HN})^{1-\alpha}\big) \tag{12}$$

Which in fact is the HN function setting β to zero. The former equation yields:

$$w = 2\pi\nu\tau_{CC} \tag{13}$$

$$\varepsilon''_{CC} = \frac{w^{1-\alpha}\cos(\alpha\pi/2)}{(1 + 2w^{1-\alpha})\sin(\alpha\pi/2) + w^{2(1-\alpha)}} \tag{14}$$

where again we use the same names for variables as in the Fortran code. The function to be fitted is the following:

$$\varepsilon''(T) = \Delta_{CC}(T)\varepsilon''_{CC} + \Delta_{HN}(T)\varepsilon''_{HN} \tag{15}$$

What we want is to describe the whole set of temperature-dependent experiments with a unique fit, that has into account the shape of the peaks (equation **??**) and its temperature

dependence (equations **??** and **??**. We will assign to the temperature dependence for the main peak (α relaxation) an equation called Vogel-Fulcher-Tamman:

$$\tau_{HN} = \tau_{HN,0} \exp \frac{D * T_0}{T - T_0} \tag{16}$$

And for the secondary peak (β relaxation) an activated process:

$$\tau_{CC} = \tau_{CC,0} \exp \frac{E}{T} \tag{17}$$

We have therefore a set of experiments performed at different temperatures (see equation **??**) , being the position of the peak fixed by its temperature dependence(see equation **??**). Therefore in the parameter file we must include the parameters for temperature dependence, plus the values for Δ_{HN},Δ_{CC},α_{HN},β_{HN},α_{CC} for each temperature. Of course, the parameter jumps that we don't want to change are set to zero.

17. Starting the fitting

Because in this case we have a series of data sets we have to do a couple o things before starting to fit:

parameter file On the first line of the parameter line we have to increase the number of functions to be fitted to 4 (third number of the parameter file). Then you just only need to write the names of the input files and the output files, together with the number of the function you want to fit to the data.

data sets Data sets must contain information about the temperature at which spectra were measured. To do that, the first word in input files must be "zvalues", the program will then automatically look for them in the next columns. First column is the number of values that characterize your experiment (it can be more than one!), and then the values (they must be ordered equally for all spectra). If you have a look at the preceding example you will se that input files were not containing the word zvalues: the program understands that there is none!

fitting If you now fit the data, you will get every nwrite steps, as usually: the time left to finish the fitting, then the total χ^2, and then the χ^2 associated to each fit to each data set.

plotting If you want now to plot the first data set together with the fit type P1, to see the second data set type P2... and so on. However you will get quite ugly plots since data were taken at constant steps in logarithmic scale. To see the plots in loglog scale type "ll" (xaxis=log, and y axis=log). Plots are now muuuch nicer.

So, and now you can redo all the analysis as in the previous examples.

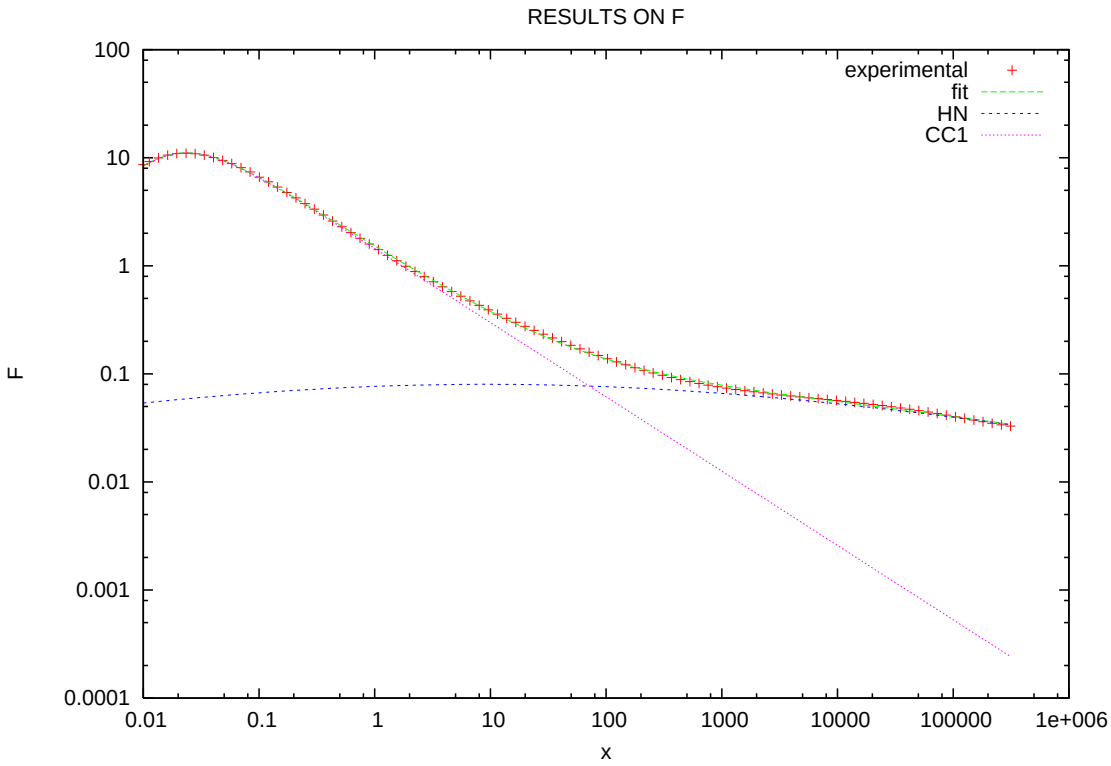

Figure 11.: Fitting of dielectric spectra, making explicit the components fitted using the PK option in FABADA.

18. Plotting separately each component

You can also plot the components for each peak separately by using the "pk" option in FABADA. Once your fitting is ready simply type pk followed by the number of function you want to plot separated in components, and you should get a graph such as this one 11. To do that let's have a look in the parameter file. As you can see in the last line (with the comment "Npeaks") there are 2 components (peaks) to be defined. To define the first one we set $\delta_{HN} = 0$ for all spectra (so parameters from 6 to 15 are set to zero), and the second one is defined by setting $\delta_{CC} = 0$ also for all functions (parameters from 16 to 25).

Part III.

Appendix

Functions currently included in the code

1 Molecular fitting: Freon11

2 Molecular fitting: Freon12

3 Molecular fitting: Benzophenone

4 Molecular fitting: CCl4

50 Dielectric functions. HN+CC Williams ansatz (FIDEWA is used)

51 Dielectric functions: HN+gauss Williams ansatz (FIDEWA is used)

52 Dielectric functions: HN+CC

53 Dielectric functions: HN+CC+CC

54 Dielectric functions: HN+CC

55 Dielectric functions: HN+CC as a function of the temperature using VFT

56 Dielectric functions: HN+CC1+CC2 as a function of the temperature using VFT

57 Dielectric functions: Polynome+CC1+CC2

58 Dielectric functions: (HN+CC1)with williams ansatz + CC2 (FIDEWA is used)

100 Polynomial (9 degree)

101 3 Gaussian+quadratic bckg

102 3 Gaussian+log(quadratic bckg)

103 5 lorentzians+quadratic bckg

110 2 norm lorentz+quadratic bckg

111 diffusion

112 diff+ROT iso(tau=0), +jumps(tau.ne.0)

113 diff+ROT anisot(tau=0)

114 Kohlrausch KWW

115 3 lorentz

116 1 gauss + 2 lorentz

117 kink

118 difusion + confined motion

119 difusion + 2 confined motions

120 difusion + rotation + confined motion

121 flow + rotation

122 delta + 2 lorentz

123 Gauss (x) 2 internal (Lor)

PLT files generated that may be changed

You can change the way to plot whatever (from data to parameter PDF or contour plots) by setting in the first line plotop equal to zero. But then you must write your own plt files. The best thing es let the program do it the first time, and then change what you do not like!. Here are the plt files used by FABADA:

f.plt plot function n

f2.plt plot function 2

chi.plt plot chi as function of steps

gpar.plt plot parameter PDF

ct.plt contour plots

cts.plt contour plots with surface

pk.plt peaks plotting

***** special plots (dielectric and mol. structure)***

p2cc.plt plot two cole-cole

phncc.plt plot HN plus CC

f2m.plt plot function 2 convoluted

Known Bugs

a lot, but still to be discovered...

Bibliography

[1] L.C. Pardo, M. Rovira-Esteva, S. Busch, M.D. Ruiz-Martin, J.Ll. Tamarit, T. Unruh. *Bayesian Analysis of QENS data: From parameter determination to model selection* arXiv:0907.3711v3 [physics.data-an].

[2] Sivia D., Data Analysis A bayesian tutorial. Oxford University Press (2006)

[3] Ende M., "Neverending story".Thienemann Verlag ISBN 3522128001 (1979)