Manual ModelFreeFFC tool
(Draft ver. 1.3)

1. Short introduction

ModelFreeFFC is a freeware standalone Matlab® tool, developed at the University of Bologna (Bologna, Italy), with a graphical interface (GUI) for Windows, Linux OSs useful for fitting nuclear magnetic relaxation dispersion (NMRD) curves obtained by fast field cycling nuclear magnetic resonance (FFC-NMR) relaxometry measurements. The model used to fit NMRD curve is supposed to be the "model free" one [1].

The Software is provided 'AS IS', without warranty of any kind, express or implied. In no event shall the copyright holder(s) be liable for any claim, data loss, loss of profits, damages or other liability arising from the Software's use or misuse. You use it at your own risk. Please read the License document for the terms of use ("Licence_ModelFreeFFC", see **Figure 3** or **Figure 4.c** upper-right "Licence" button).

In addition to the compiled standalone GUI version (developed with the App Designer Matlab tool), there is also a standalone command- line executable version, both for Windows OS and Linux OS (See SOFTWARE INSTALLATION paragraph). The tool allows the user to handle some parameters that greatly influence the quality of the NMR maps and the computation time to perform the inversion. There is also a set of diagnostic tools that allow the user to evaluate the quality of the measured data. Remember that to have a good inversion, it is fundamental to have high-quality data (data without too many outliers or electronic baseline, relaxation curves not correctly sampled or distorted, etc.).

After a brief introduction of the theory, the algorithm used in the program is presented. Then the installation of the software, the user interface and how to use the program are described.

2. General Theory

The FFC-NMRD technique is a powerful and versatile tool for investigating the molecular dynamics of condensed matter (solid, amorphous, liquid, or even heterogeneous systems). The FFC technique evaluates how the longitudinal relaxation rate (R_1), or the longitudinal relaxation time ($T_1=1/R_1$), varies on changing the strength of an applied magnetic field.

The mathematical model implemented in ModelFreeTool to fit the NMRD data follows the "model free" approach, where the mathematical treatment of NMRD curves is proposed as a linear combination of Lorentzian terms, the number of which must be discretionally/empirically set, i.e.

$$R_1 = (T_1)^{-1} = \sum_i \frac{c_i \tau_i}{1 + (\omega_L \tau_i)^2}$$

where c_i is the weight coefficient of the correlation time τ_i and R_1 depends on the nuclear Larmor angular frequency ω_L. From the set of τ_i an weighted average correlation time $\langle\tau_c\rangle$ can be calculated as $\langle\tau_c\rangle=(\Sigma_i c_i\tau_i)/\Sigma_i c_i$. For details on the models, the reader is referred to the papers [1] and the bibliography cited therein. Specifically, the Model-free method implemented in the software for the analysis of NMRD is based on the following expression:

$$R_1(\omega_L) = R_0 + R^{HH}$$

where R_0 is an offset keeping into account very "fast" molecular motions ($\tau_c < 1$ ns) and

$$R^{HH} = \int_0^\infty \left(\frac{\tau_c}{1 + (\omega_L \tau_c)^2} + \frac{4\tau_c}{1 + 4(\omega_L \tau_c)^2}\right) f^*(\tau_c) d\tau_c \approx Kf$$

Given the function R_1 defined by the experimental data set, the problem reduces to retrieve an inverse integral transform. From a mathematical standpoint, this is considered an "ill-posed problem", hard to

solve analytically. The Matlab function **l1_ls_nneg** (l1-Regularized Least Squares Problem Solver, implemented in the "SolverL1Ls.m" script) solves problems of the following form:

$$\min_{x_i>0} \|Ax - y\|^2 + \lambda \sum_{i=1}^{n} x_i$$

Where in our analysis A = K (see (1)), y = $R_1 - R_0$, and x = f_{num} (the normalized distribution function $f(\tau_C)$).

$$K = \left(\frac{\tau_{c_j}}{1+(\omega_{L_i}\tau_{c_j})^2} + \frac{4\tau_{c_j}}{1+4(\omega_{L_i}\tau_{c_j})^2} \right)_{i,j} \quad (1)$$

So, the normalized distribution function $f(\tau_C)$ and the offset value R_0 are determined as a solution of the minimum problem:

$$min_{f_{num},R_0 \geq 0} \|Kf_{num} + R_0 - R_1\|^2 + \lambda \|f_{num}\|_1 \quad (2)$$

The regularization parameter λ is determined automatically. In the iterative process to resolve (see (2)), R (0) = 0, f_{num} (0) = 0 were chosen as initial values.
Finally, R_{1_num} is computed as follows

$$R_{1\_num} = K * f_{num} + R_0$$

Then it is possible to make statistics on residuals, where residuals are given by the function R_{1_num} obtained numerically and the difference between the function R_1 defined by the experimental data set.

3. Inversion Algorithm

After launching the software and choosing the folder with the data you want to work with, the steps of the ModelFreeFFC algorithm can be summarized as follows:

1) Import **R_1** values (longitudinal relaxation rate defined by the experimental data) in s^{-1} unit, **f** field points in MHz unit and **Int_err** (error of R_1 data) from the ".xlsx" file into the chosen Data subfolder.
2) Compute the vector ω_L (nuclear Larmor angular frequency computed with $\omega_L = 2*\pi*f$).
3) Compute the logarithmically spaced vector τ_c in microsecond unit (computed with *logspace* Matlab command: *Tauc=logspace(log10(parFile.T1min), log10(parFile.T1max), parFile.nbin*) where *T1min, T1max* and *nbin* are parameters inputted by the user, see explanation in SOFTWARE INSTALLATION paragraph.
4) Compute the matrix **K** with the Matlab function: *MF_Kernel*.
5) Compute **f_{num}** (distribution function $f(\tau_C)$), the offset value **R_0** , **R_{1_num}** (longitudinal relaxation rate R_1 computed with $R_{1\_num} = K * f_{num} + R_0$ command) and **hist** (a structured variable with a list of computed parameters) with the Matlab function: *SolverL1Ls*. The inversion algorithm applies the *l1_ls_nneg* function [2] to solve an L_1 regularized least square problem with positive constraints. The regularization parameter is automatically computed by balancing the contributions of the regularization term and the data fit term.
6) Compute residual vector (=$R_{1\_num} - R_1$) and statistics on residuals with the Matlab function: *Residual_Analysis_main*.
7) Plot Distribution with the Matlab function: *grafico_Regressed_Data*.
8) Plot NMRD curve, comparison between **R_1** and **R_{1_num}** with also the **R_1** error data (*errorbar* command), with the Matlab function: *grafico_Computed_Data.*
9) Printout computed distribution and parameters (see OUTPUT FILES section, **Figure 9.a**).
10) Create the following output files: ***1D_Distribution.txt*** and ***Statistics.txt*** (that contains the arguments of the previous point).

4. Installation and Program description

SOFTWARE INSTALLATION

This SW was written under Matlab 2020b, but it does not need to have Matlab installed on the user PC. However, in order to used it, you need to download the appropriate free Matlab runtime library. For this FFCNMR Solver release the runtime library is named R2020b (9.9).

At present there are compiled versions for Windows OS (tested under Windows 10) and Linux (tested under Ubuntu 20.04).

Under Windows OS, the installation consists of two steps: 1) install the free Matlab runtime library and 2) unzip the zipped folder with the ModelFreeFFC set of files and subfolders.

In the following, the full procedure is described.

Follow the link:

https://www.mathworks.com/products/compiler/matlab-runtime.html and download the file "R2020b" (see red arrow below in **Figure 1**).

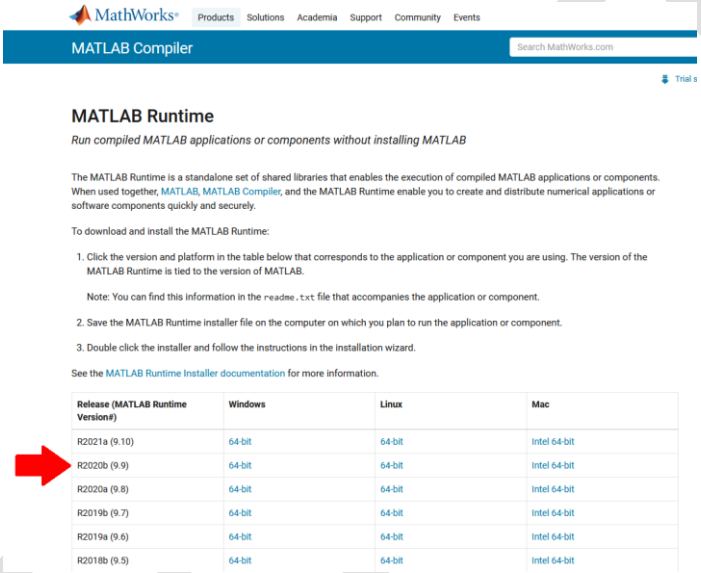

Figure 1-Mathworks webpage where to download the Matlab runtime library.

After a while you will have downloaded the following file "MATLAB_Runtime_R2020b_Update_5_win64.zip" (**Figure 1**).

It might take some time because the file is very big, about 3GB. The name of the zip file could be a little different owing to new library update. Once it's been downloaded, unzip the folder into a temporary folder and click on the executable file "setup.exe" (see the red arrow in **Figure 2**) to install the Matlab library.

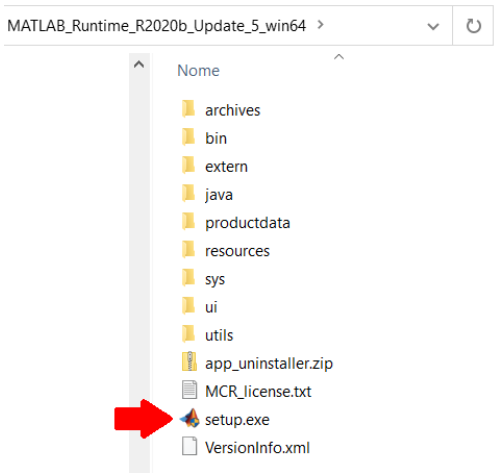

Figure 2 – Matlab runtime library installation.

After having installed the Matlab library, unzip the zipped distributed SW “ModelFreeFFC.zip” to create the working folder “ModelFreeFFC” (see **Figure 3.a**).

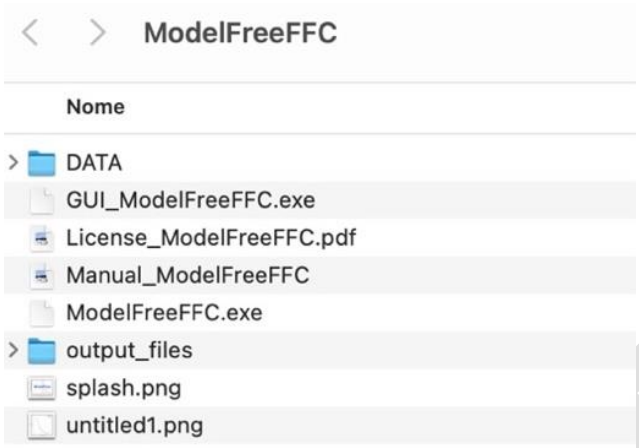

Figure 3.a – Contents of the “ModelFreeFFC” folder.

The ModelFreeFFC package consists of a folder, named “ModelFreeFFC” which contains the following folders and files (see **Figure 3.a**):

- “ModelFreeFFC.exe” is the standalone command line program version. Double-click to launch the application;
- “GUI_ModelFreeFFC.exe” is the standalone GUI program version. Double-click to launch the application;
- “DATA” a folder used to store the input data files and parameter files with the parameters used to drive ModelFreeFFC during the inversion process (see next paragraph for a detailed description of these files). The DATA folder comes with four examples whose data are stored in the “Anidra”, “Latte”, “Pioppo” and “Vinacce” subfolders; (see **Figure 3.b**)
- “output_files” a folder that contains the following output files created runtime by the computation:
  - “1D_Distribution.txt”, the computed distribution of correlation times;
  - “Statistics.txt”, postprocessing computation performed on residuals to evaluate the regression quality. (see **Figure 3.c**)
- “splash.png” the splash screen image (the image that appears while the program is launching);
- the “License_ModelFreeFFC.pdf“ the pdf file with the license terms;
- the “Manual_ModelFreeFFC” this manual.

Figure 3.b – Contents of the “DATA” folder.

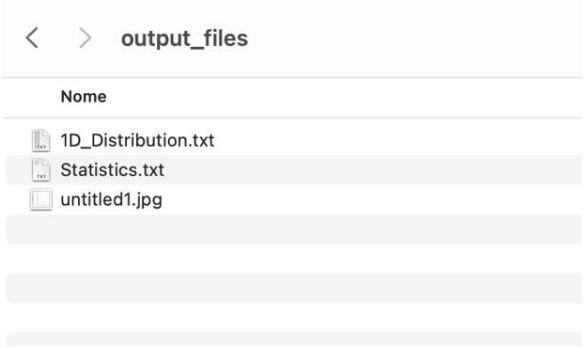

Figure 3.c – Contents of the “output_files” folder**.**

Under Linux the installation follows almost the same steps as under Windows, although it is, as usual, more complex and can depend on the different distribution/version of Linux.

COMMAND LINE VERSION RUN AND DATA LOADING

To run the program under Windows, double click on the file “ModelFreeFFC.exe” contained in the ModelFreeFFC folder. Once the program is launched (it can takes few seconds before to start), an input data dialog window (“Open Data Directory”) will appear asking for the directory where the datafiles and parameter files are stored. The input dialog is shown in **Figure 4.a**. It is also automatically open a terminal where computation information is printed out (see **Figure 4.b**).

To select data and parameter files select the desired folder present in DATA by clicking on the “Folder Selection” button. Then, the data and parameters inside the DATA folder are automatically loaded by the program and the computation starts.

Under Linux OS, the Matlab compiler creates a script (in this specific case named run_ModelFreeFFC.sh) that allows the launch of the program under Linux.

To run the command line version, use the following command:

“sudo ./run_ ModelFreeFFC.sh /usr/local/MATLAB/MATLAB_Runtime/v99”

where “/usr/local/MATLAB/MATLAB_Runtime/v99” is the location where has been installed the runtime library of Matlab 2020b.

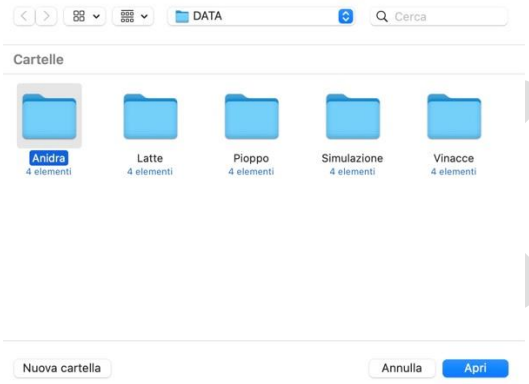

Figure 4.a – Input dialog to select the data directory. Single-click to select the sub-folder of DATA with the data to be processed.

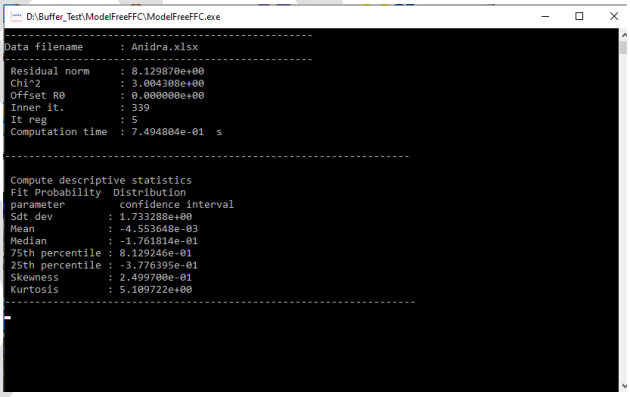

Figure 4.b – Windows terminal with computational information printed out.

GUI INTERFACE

Under Windows, double clicking on the executable file 'GUI_ ModelFreeFFC.exe' and after a few seconds, required for the initialization of the SW environment, the interface shown in **Figure 4.c** will be displayed. To do this under Linux, it is necessary to run the following terminal command:

“sudo ./run_GUI_ ModelFreeFFC.sh /usr/local/MATLAB/MATLAB_Runtime/v99”

where “/usr/local/MATLAB/MATLAB_Runtime/v99” is the location where has been installed the runtime library of Matlab 2020b.

With the GUI interface the user can test the effects on the computed map resulting from parameter changes. The parameter meanings are the same as described in the previous paragraphs.

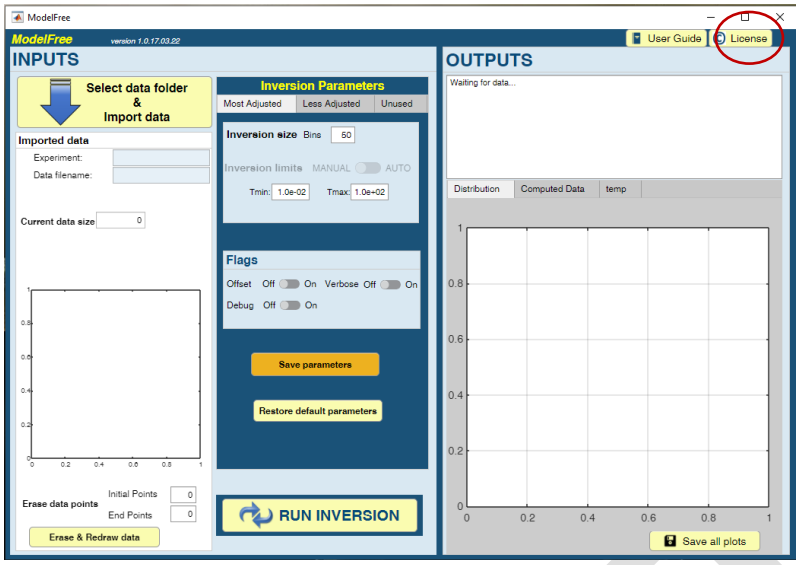

Figure 4.c – GUI of ModelFreeFFC.

SOFTWARE USAGE: INPUT DATA, PARAMETER FILES AND KEYWORDS

Data files

The folder "ModelFreeFFC" (see **Figure 5.a**) contains a subfolder called "DATA" in which you can find subfolders ("Anidra", "Latte", "Pioppo", "Vinacce") with datafile and parameter files (see **Figure 5.b**).

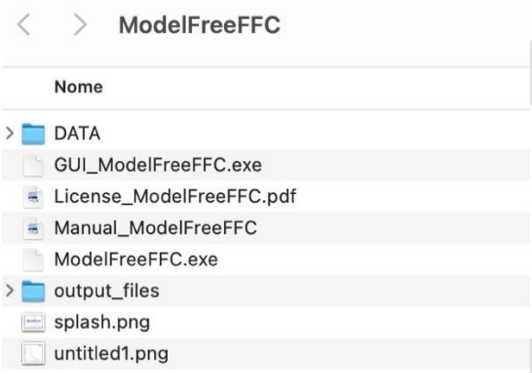

Figure 5.a – Subfolders and files of the ModelFreeFFC folder.

Figure 5.b – Datafile and parameter file of the Anidra folder, one of the DATA's subfolders .

In each of these subfolders you can find a data files in the ".xlsx" format.
The structure of any ".xlsx" file (see **Figure 6**) is made up of 3 columns, the first one reports the field points (in MHz unit), the second one for the corresponding R_1 values (in s^{-1} unit) and the third the error of R_1 data (in s^{-1} unit).

	Freq. (MHz) A	R_1 (s^{-1}) B	Error (s^{-1}) C
1	35.0	3.1	0.1
2	30.0	3.6	0.1
3	25.0	4.0	0.1
4	20.0	4.6	0.1
5	17.0	4.9	0.2
6	15.0	5.7	0.3

Figure 6 – xlsx datafile structure example (Anidra).

Parameter files with the keywords

The three parameter files with extension ".par" (see **Figure 5.b**) contain the keywords that drive the inversion computation. The files are in "fixed format", therefore the position (the column of the "=" symbol) of the value of keyword cannot be changed. Keyword value in a wrong position crashes the program. Rows that do not contain a keyword are interpreted as comments and ignored.
Changing the value of a parameter can improve or worsen the quality of the distribution computed and can also substantially increase/decrease the computation time in an unpredictable manner.
This because of the parameter changes also have effects on the matrix structure involved in the inversion procedure.

- **FilePar.par**, contains a list of keywords used by the inversion algorithm of ModelFreeFFC_ver0.2 (See **Figure 7.b**)
  - par.l1ls.tol, convergence tolerance of the l1ls algorithm (default: 1.0E-02);
  - par.l1ls.duality, relative target duality gap, internal parameter (default: 1.0E-05);
  - par.Solver.maxiter, maximum number of iterations of the solver algorithm (default: 1.0E+03);
  - par.Solver.tol, convergence tolerance of the solver algorithm (default: 1.0E-02).

  The ultimate version of the code can fit dispersion curves with quadrupole dips. The corresponding parameters are not described in this draft version.

- **FileFlag.par**, contains a list of keywords that modify the functionality of ModelFreeFFC_ver0.2 (See **Figure 7.a**)
  - FL_typeKernel, not used (kept for compatibility reasons);
  - FL_InversionTimeLimits, not used (kept for compatibility reasons);
  - FL_OutputData, (default value=1);
  - FL_NoContour, (default value=0);
  - FL_Verbose, 0 or 1. 1 to display additional information (default value=0);
  - FL_Debug, 0 or 1. 1 to add further output, useful for diagnostic analysis of the code. Its activation can have effect on the computation time and on the quality/shape of the computed distribution (default value=0);
  - FL_Amp_scale, multiplication factor of the data amplitude (default value =1);
  - FL_Scale_fact, multiplication factor of the frequency intensity (default value=1);
  - FL_T1T2Filter, (default value=0);
  - FL_EraseCol, exclude from the computation the last "FL_EraseCol" elements of the data vector (default value=0);
  - FL_EraseRow, exclude from the computation the first "FL_EraseRow" elements of the data vector (default value =0);
  - FL_Offset, enable the regression also of a constant term, the R_0 term of equation 1) (can be 0 or 1, default value =0);
  - FL_Stat, enable computation of postprocessing statistics performed on the residual, the difference between measured and reconstructed data. 0 to disable statistics, a positive value is used as threshold to highlight residual differences (default value=5).

- **FileSetInput.par**, contains the data filename, and other parameters used during the inversion (See **Figure 7.c**)
  - Filenamedata, the name of the xlsx data file;
  - nbin, number of correlation time bins used for the linear inversion, increasing this number will smooth the plotted curve but at the same time it will increase the computation time;
  - T1min, minimum inversion correlation time used for inversion;
  - T1max, maximum inversion correlation time used for inversion;

It is worth noting that the tau ranges determined by the limits can have strong effect on the computed tau-distribution, indeed they act on the matrix used by the inversion algorithm.

FileFlag.par

```
Flags for experiment
%
  FL_typeKernel          =2
  FL_InversionTimeLimits=0
  FL_OutputData          =1
  FL_NoContour           =0
  FL_Verbose             =0
  FL_Debug               =0
  FL_Amp_scale           =1
  FL_Scale_fact          =1
  FL_T1T2Filter          =0
  FL_EraseCol            =0
  FL_EraseRow            =0
  FL_Offset              =1
  FL_Stat                =5
END
```

Figure 7.a – FileFlag.par file

FilePar.par

```
Inversion Parameters for experiment
%
%[L1Ls parameters]
  par.l1ls.tol           =1.0E-02
  par.l1ls.duality       =1.0E-05
%
%[SolverL1Ls parameters]
  par.Solver.maxiter     =1.0E+03
  par.Solver.tol         =1.0E-02
%
END
```

Figure 7.b – FilePar.par file.

FileSetInput.par

```
Filenames e parameters for experiment
%
% [File Data]
  filenamedata           =Anidra.xlsx
%
% [Inversion Points]
  nbin                   =100
% [Inversion Time limits]
  T1min                  =1.0E-4
  T1max                  =1.0E+1
%
END
```

Figure 7.c – FileSetInput.par file.

How to upload the data: Click on the "Import Data" button (blue arrow at the top left in **Figure 8.a**, highlighted in red) to upload the data in the display window (see **Figure 4.a**).

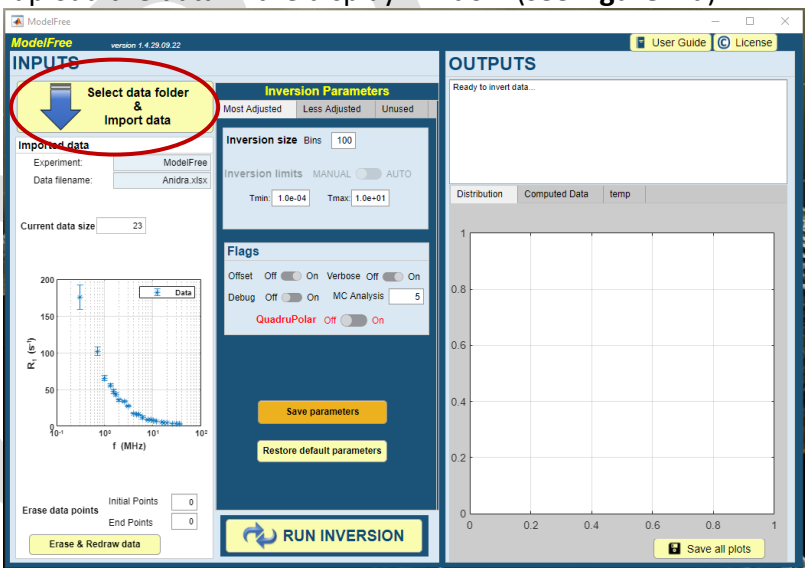

Figure 8.a – Interface with model selection buttons highlighted and imported data plotted.

The SW will provide default initial values of parameters red from the parameters files used then for the fitting, however, if you prefer, you can insert the parameters by yourself (highlighted in green in **Figure 8.b**) and you can either enable or not an offset on the baseline (highlighted in orange in **Figure 8.b**) for improving the fitting.

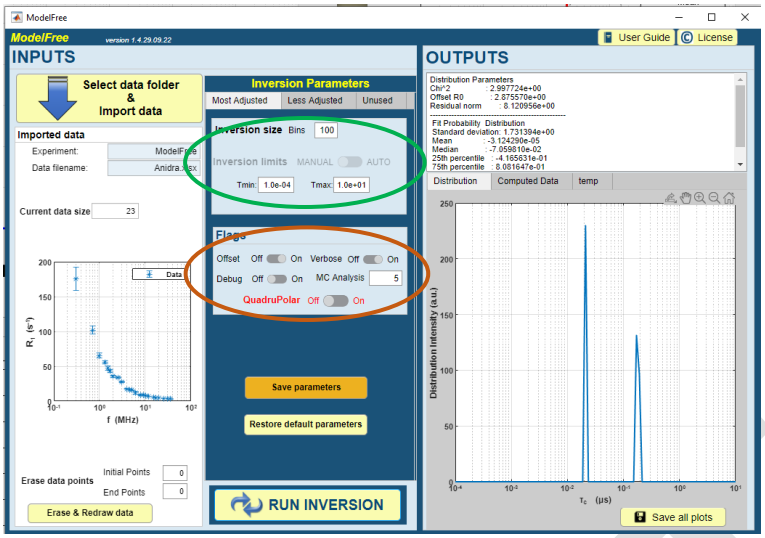

Figure 8.b – Interface with parameters and flags buttons highlighted.

If you click on "**Run Inversion**" button (bottom left in **Figure 8.c**, highlighted in red) the fitting results is displayed ("Distribution" plot in **Figure 8.c**), and the fitted values are reported below the plot ("Computed Data" plot in **Figure 8.d**).

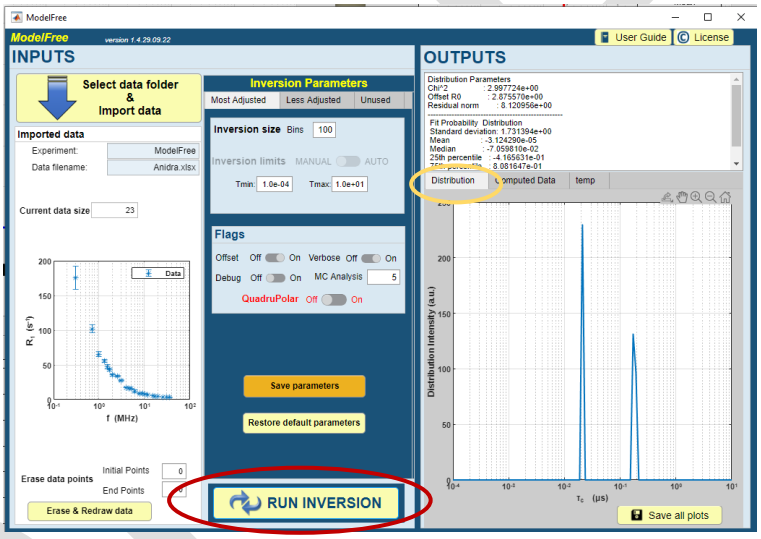

Figure 8.c – Interface with fit distribution showed.

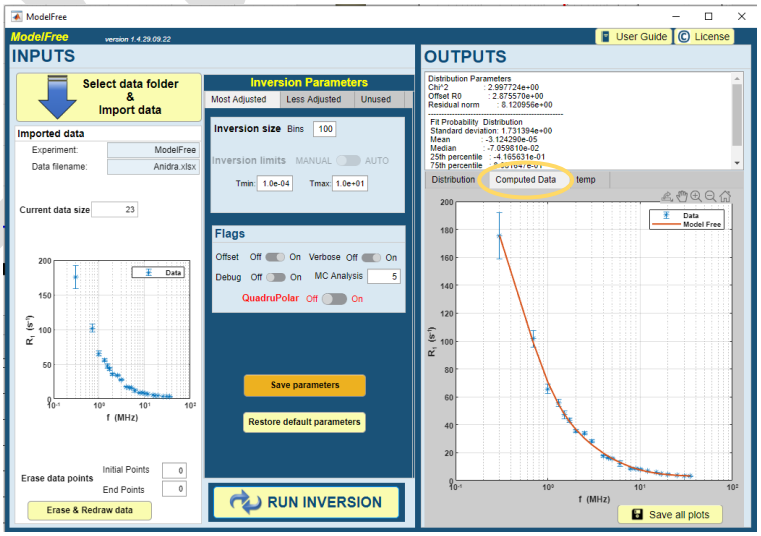

Figure 8.d – Interface with used and computed data are shown.

OUTPUT FILES

When the computation terminates, many files are saved in the folder "\ModelFreeFFC\output_files\" (see **Figure 3.c**). A few of them are saved only if the "FL_verbose" flag is on. The text file with the statistics is saved with the name "Statistics.txt" (see **Figure 9.a**). The text file with the matrix of the computed distribution (the first column reports τ_c, the second one instead shows f_{num}) is saved in an ASCII file with the name "1D_Distribution.txt" (see **Figure 9.b**).

In particular, the software shows: the distribution intensity map (see **Figure 10.a**) and the R1 distribution (where you can see the comparison between R_{1_num} and R_1) (see **Figure 10.b**).

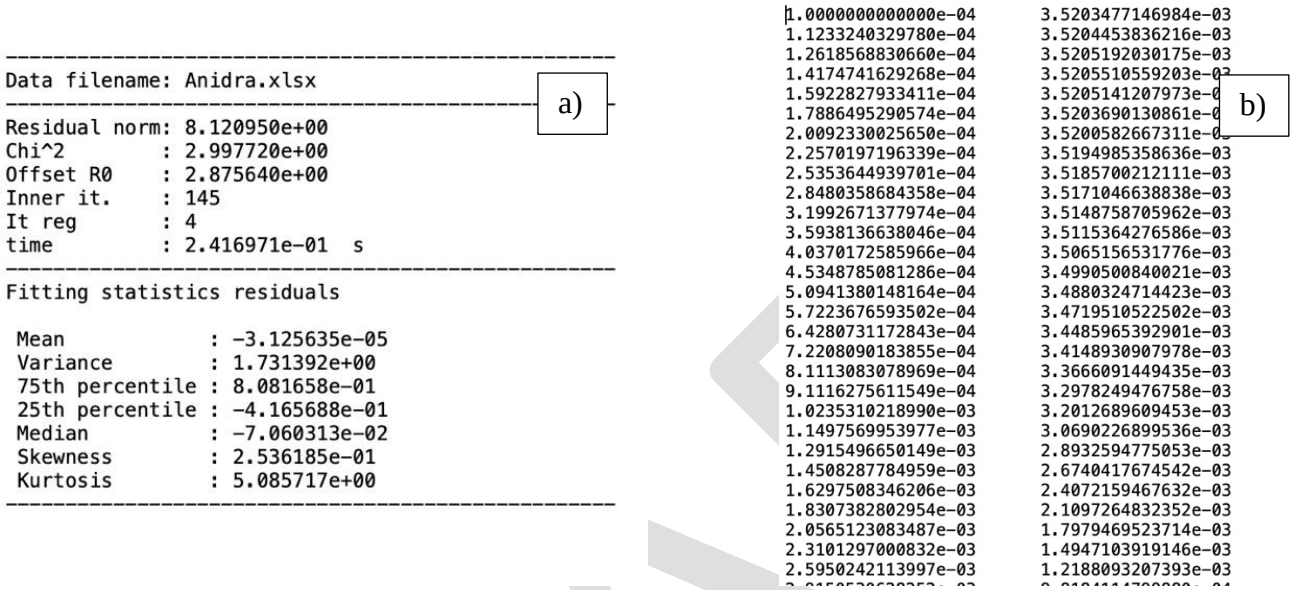

Figure 9 – Text files in output_files folder (Anidra), a) "Statistics.txt", b) 1D_Distribution.txt.

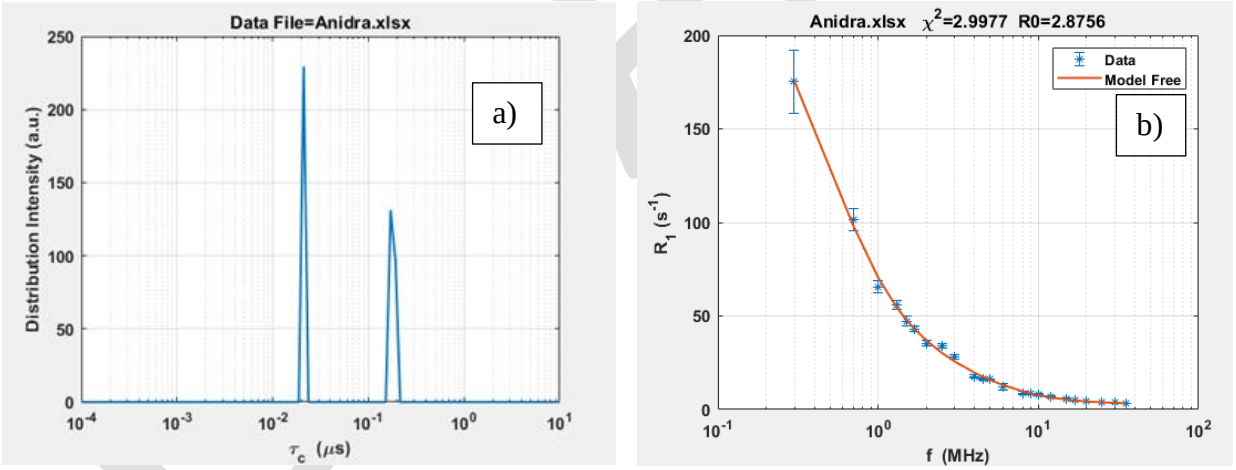

Figure 10 – Pictures of the plots automatically created by ModelFreeFFC tool.

STATISTICS

If the parameter flag FL_Stat is enabled (a value >0), then some statistics on residuals (calculated as: Data- ComputedData) are computed (see **Figure 9.a**), and the corresponding plots are shown.

a) The residual Box (**see Figure 11**), where on each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers. The outliers are plotted individually using the '+' symbol.

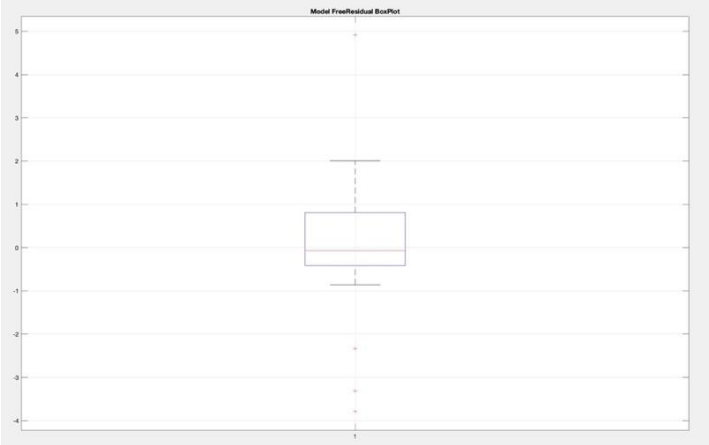

Figure 11 – The residual Box.

b) The histogram of residuals (see Figure 12) with the corresponding Normal distribution (red curve) computed using the Matlab function Residual_Analysis_main.

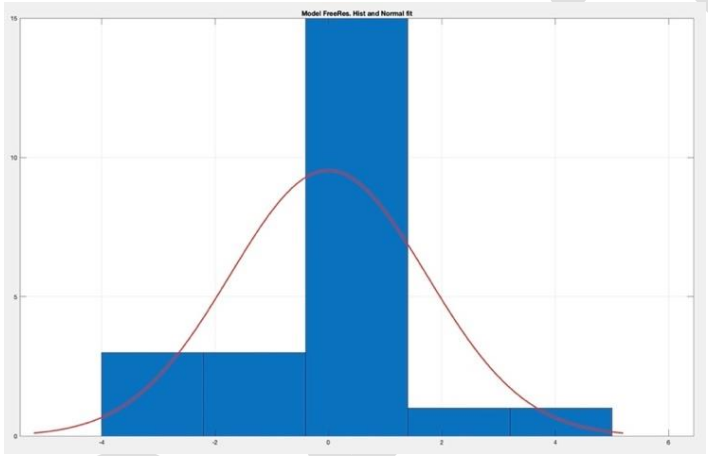

Figure 12 – The histogram of residuals.

5. References

[1] P. Lo Meo, S. Terranova, A. Di Vincenzo, D. Chillura Martino and P. Conte. Heuristic Algorithm for the Analysis of Fast Field Cycling (FFC) NMR Dispersion Curves. Anal. Chem. 2021, 93, 24, 8553–8558. https://doi.org/10.1021/acs.analchem.1c01264.

[2] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale l_1-regularized least squares. IEEE Journal on Selected Topics in Signal Processing, 1(4):606–617, 2007.