A Short Manual for the Upen2DTool Software

V. Bortolotti[a], R.J.S. Brown, P. Fantazzini[b], G. Landi[c], F. Zama[c]

[a]*Department of Civil, Chemical, Environmental, and Materials Engineering, University of Bologna*
[b]*Department of Physics and Astronomy, University of Bologna, Italy*
[d]*Department of Mathematics, University of Bologna*

version 1.2 – October 2023

This 2-dimensional (2D) inversion program developed at the University of Bologna (Bologna, Italy), named Upen2DTool, processes 2D ASCII Nuclear Magnetic Relaxation (NMR) data to produce distributions in the two NMR dimensions (specifically longitudinal T_1 and transverse T_2 relaxation times). Upen2DTool implements the Upen2D algorithm, an 2D extension of the Uniform PENalty principle, to compute locally adapted regularization parameters and approximate solutions by solving a sequence of regularized least squares problems. The software is written in Matlab® script language and it comes with a version compiled for Windows. After a brief introduction of the theory, we present the algorithm used in the program. Then we describe the user interface and how to use the program. Finally, some examples of computation of the accompanied data are presented.

1   INTRODUCTION AND GENERAL THEORY

The structure of different types of porous media can be analyzed through measurements of the NMR relaxation of ^1H nuclei. Parameters like the longitudinal (T_1) and the transverse (T_2) relaxation times, as well as the self-diffusion molecular coefficient, can be determined and associated to properties of the ^1H fluids and saturated porous media.

In case of heterogeneous porous media samples, one expects to measure a multi-exponential decay signal from the relaxation measurements (but also for the diffusion measurements). So, for examples in case of 1D signal, one expects to find:

$$S(t_i) = g_i = \sum_{k=1}^{M} A(T_k) e^{-\frac{t_i}{T_k}} + \varepsilon_i \quad i=1, \dots, N \qquad 1)$$

Where, T_k are the relaxation times, $A(T_k)$ the unknown system relaxation times and ε_i represent the noise of the signal. N is the number of acquired points of the relaxation curve and M the number of relaxation components of the underlying physical model.

The problem of determining the distribution of relaxation times from NMR data is an inverse ill-posed problem modeled by a first kind Fredholm Integral Equation having separable exponential kernel, $k_1(t_1, T_1)$ and $k_2(t_2, T_2)$, that in a 2-dimensional, 2D, case has the following form:

$$S(\tau_1, \tau_2) = \int\int_0^\infty k_1(t_1, T_1) k_2(t_2, T_2) F(T_1, T_2) \, dT_1 \, dT_2 + e(t_1, t_2) \qquad \text{2)}$$

where $e(t_1, t_2)$ represents additive Gaussian noise. The evolution times t_1 and t_2 are independent variables in the two dimensions: t_1 is the evolution time of the experiment in the first dimension (IR or CPMG experiment) and t_2 the evolution time in the second dimension (CPMG experiment).

The kernels are decaying exponential functions whose expression depends on the specific NMR experiment and the unknown distribution $F(T_1, T_2)$ is supposed to be greater or equal to a constant $\ell \in \mathbb{R}$, not necessarily positive.

There are several codes, usually developed in Matlab®, that solve this ill-posed problem using different methods. We have developed UPEN2D, a Matlab script, that solves the 2D inverse ill-posed problem implementing the UPEN (Uniform PENalty) principle [1]. Therefore, UPEN2D computes locally adapted regularization parameters and approximate solutions by solving a sequence of regularized least squares problems.

Considering the discretized form of the integral equation (2), in order to avoid artifact peaks in the computed distribution, UPEN2D applies a multiple-parameter Tikhonov regularization and solves the following minimization problem:

$$\min_{\mathbf{f} \geq \ell} \left\{ \| (\mathbf{K}_2 \otimes \mathbf{K}_1)\mathbf{f} - \mathbf{s} \|^2 + \sum_{i=1}^{N} \lambda_i (\mathbf{Lf})_i^2 \right\}$$

where $\mathbf{K}_1 \in \mathbb{R}^{M_1 \times N_1}$, $\mathbf{K}_2 \in \mathbb{R}^{M_2 \times N_2}$ are the discretized exponential kernels, $\mathbf{s} \in \mathbb{R}^M, M = M_1 \cdot M_2$, is the discrete vector of the measured noisy data, $\mathbf{f} \in \mathbb{R}^N$, $N = N_1 \cdot N_2$, is the vector reordering of the distribution, $\| \cdot \|$ is the L_2 norm, $\mathbf{L} \in \mathbb{R}^{N \times N}$ is the discrete Laplacian operator and λ_i are the locally adapted regularization parameters, $i = 1, \dots, N$. The present version of the code implements the kernels for IR-CPMG (T_1-T_2 model) and CPMG-CPMG (T_2-T_2 model) experiments.

In the T_1-T_2 case, the two kernels have the following expressions:

$$K_1(t_1, T_1) \ = \ 1 - 2\, e^{-\frac{t_1}{T_1}}, \qquad\qquad K_2(t_2, T_2) \ = \ e^{-\frac{t_2}{T_2}}$$

while in the T_2-T_2 case, the two kernels have the following expressions:

$$K_1(t_1, T_{21}) \ = \ e^{-\frac{t_1}{T_{21}}}, \qquad\qquad K_2(t_2, T_{22}) \ = \ e^{-\frac{t_2}{T_{22}}}$$

In order to improve the quality and the efficiency of the reconstructions, Upen2D script makes it possible to apply Singular Value Decomposition (SVD) filtering. In fact, the SVD filter allows the reduction of the data size by projecting the data vector $\mathbf{s}$ onto a lower-dimensional subspace spanned by the first left singular vectors of the kernel matrices $\mathbf{K_1}$ and $\mathbf{K_2}$.

2   ALGORITHM

The numerical method implemented in Upen2D is an iterative procedure where, at each iteration, suitable values for the λ_i's are determined by imposing that all the non-null products $\lambda_i(\mathbf{Lf})_i^2$ have the same constant value (UPEN principle) and an approximate distribution is obtained by solving a problem with Newton's Projection method. See reference [1] for detailed explanation.

Briefly, let $p_\mu^{(k)}$ and $c_\mu^{(k)}$ denote the values of the gradient and Laplacian of the reconstructed map $\mathbf{f}^{(k)}$ at index μ, then the steps of the Upen2D algorithm can be summarized as follows:

1- Using the SVD threshold parameter (> 0) compute the truncated SVDs $\hat{\mathbf{K}}_1$ and $\hat{\mathbf{K}}_2$ matrices of the matrices $\mathbf{K_1}$ and $\mathbf{K_2}$;

2- Compute $\hat{\mathbf{s}}$ the SVD projection of $\mathbf{s}$;

3- Set k = 0. Using the tolerance parameter Tol$_{GP}$ and running few iterations the of Gradient Projection method, compute a staring guess $\mathbf{f}^{(0)}$ for the distribution $\mathbf{f}$:

$$\mathbf{f}^{(0)} = \arg\min_{\mathbf{f}\geq\ell} \| (\mathbf{K}_2 \otimes \mathbf{K}_1)\mathbf{f} - \hat{\mathbf{s}} \|_2^2$$

4- repeat

    a) compute

$$\lambda_i^{(k)} = \frac{\| (\mathbf{K}_2 \otimes \mathbf{K}_1)\mathbf{f}^{(k)} - \hat{\mathbf{s}} \|_2^2}{N\left(\beta_0 + \beta_p \max_{\mu\in I_i}(p_\mu^{(k)})^2 + \beta_c \max_{\mu\in I_i}(c_\mu^{(k)})^2\right)} \quad i = 1,\ldots,N$$

    b) using the tolerance parameters Tol$_{NP}$, Tol$_{CG}$ and the NP and CG algorithms, compute

$$\mathbf{f}^{(k+1)} = \arg\min_{\mathbf{f}\geq\ell} \| (\mathbf{K}_2 \otimes \mathbf{K}_1)\mathbf{f} - \hat{\mathbf{s}} \|_2^2 + \sum_{i=1}^{N} \lambda_i^{(k)}(\mathbf{Lf})_i^2$$

  until $\| \mathbf{f}^{(k+1)} - \mathbf{f}^{(k)} \| \leq Tol_{UPEN} \| \mathbf{f}^{(k)} \|$

5- Save computed distribution on file

The algorithm depends on several parameters that can be grouped as follows:

• Parameters: β_{00}, β_0, β_p, β_c used to control the proper choice of the local regularization parameters λ_i. See references [1], [8] and [7] for a detailed discussion.

• Tolerance parameters used to stop the iterative methods: Tol$_{CG}$, Tol$_{NP}$, Tol$_{GP}$, Tol$_{UPEN}$. The default values, that have proven to work properly for many kinds of different set of data, are reported in table 1.

• SVD threshold parameter used to filter the input data, the default value is set to 10^{-2}.

Tol_{CG}	Tol_{NP}	Tol_{GP}	Tol_{UPEN}
10^{-5}	10^{-5}	10^{-2}	10^{-3}

Table 1- Default Tolerance parameters.

3 DESCRIPTION OF THE PROGRAM AND INSTALLATION

These notes refer to the compiled version of Upen2DTool.

It is a command line program that computes 2D multi-exponential inversion of data that must be stored in an ASCII format file.

The Upen2D package consists of a folder, named "Upen2DTool_ver.1.1", which contains the following folders and files:

• "Upen2DTool.exe" is the executable program. Double-click to launch the program;

• "DATA" a folder used to store the input data files, and parameter files with the parameters used to drive Upen2D during the inversion process (see next paragraph for a detailed description of these files). The DATA folder has three embedded examples whose data are stored in the "T1-T2", "T2-T2" and "T1_T2_Synth" subfolders;

• "output files" a folder that contains the following output files created runtime by the computation:

- "2D_Distribution.txt", the computed 2D map of relaxation times;

- "Parameters.txt", with the relevant output parameters such as computation time, iteration numbers, value of the output residual norm;

- "T1.txt" and "T2.txt, with the values of the vectors T_1 and T_2 in eq. (1);

- "t1.txt" and "t2.txt", with the values of the vectors t_1 and t_2 in (1);

- "Residual.txt", with the residual of the last computed 2D map is saved in the file.

 • "DOC" folder with the manuals;

• "splash.png" the splash screen image (the image that appears while the program is launching).

Installation

To install the package just copy the Upen2DTool_ver.1.1 folder to your HD.

The compiled version needs the appropriate runtime library. Verify that MATLAB Runtime(R2023a) is installed. If not, you can download (and then install) the Windows version of the free MATLAB Runtime for R2023a from the following link on the MathWorks website:

   https://www.mathworks.com/products/compiler/mcr/index.html

For more information about the MATLAB Runtime and the MATLAB Runtime installer, see "Distribute Applications" in the MATLAB Compiler documentation in the MathWorks Documentation Center.

Program run and data loading

To run the program double click on the file "Upen2DTool.exe" contained in the Upen2DTool_ver.1.1 folder. Once the program is launched, 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 1.

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.
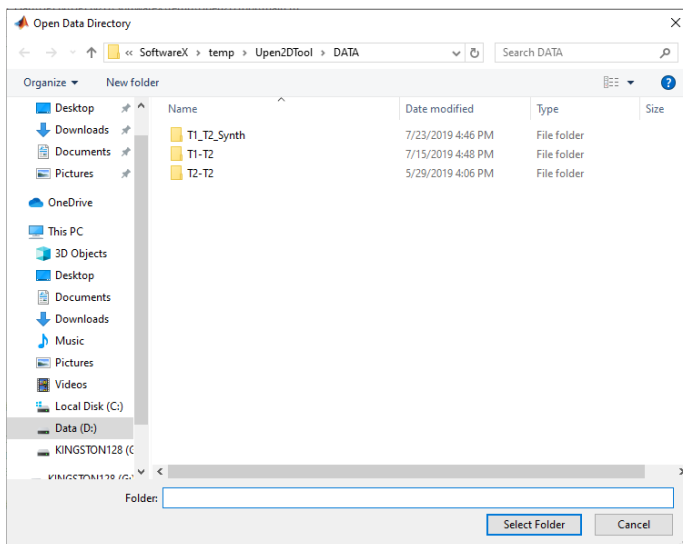

Figure 1- Input dialog to select the data directory. Single-click to select the sub-folder of the data folder DATA.

A message dialog box (see Figure 2) will appear and will remain on until the computation is finished.
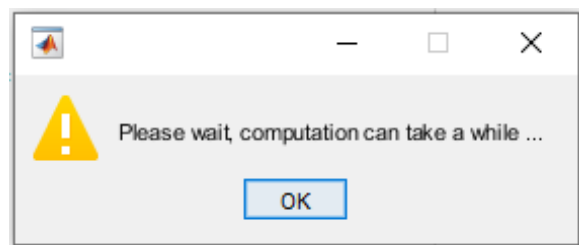

Figure 2- Message dialog box.

Input data, parameter files and keywords

A data folder must contain 6 files: three data files and three parameter files.

Data files

- a column ASCII data files with the list of the M times used to acquire the relaxation curve related to the first dimension. Usually the first dimension is the Inversion times of a IR sequence or the echo times of a CPMG sequence;

- a column ASCII data files with the list of the N times used to acquire the relaxation curve of the second dimension. Usually it contains the Echo times of a CPMG sequence;

- a 2D M×N matrix data file with the acquired signal. For example, in case of a IR-CPMG acquisition, the M rows of the matrix are the CPMGs acquired with N echoes at M different IR inversion times.

Parameter files with the keywords

The three (3) parameter files 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.

- FileFlag.par, contains a list of keywords that modify the functionality of Upen2D:

    -FL_typeKernel, 1 or 4, 1 to select the T_1-T_2 kernel model or 4 for the T_2-T_2 kernel model;

    -FL_InversionTimeLimits, 0 or 1, if the value is 0 then Upen2D automatically computes the left and right extreme limits of the relaxation times ($T_1(1)$, $T_1(M)$, $T_2(1)$ and $T_2(N)$) used in the inversion process using 4 times the last measured data times ($t_1(M)$, $t_2(N)$) and ¼ of the first measured data times ($t_1(1)$, $t_2(1)$), respectively. Otherwise if the value is 1 then Upen2D will use the limits as reported in the FileSetInput.par file (see below);

    -FL_OutputData, 0 or 1. 1 to create output files;

    -FL_NoContour, Number of Contour lines to obtain 2D contour maps. Set 0 to obtain the 2D map without contour lines;

    - FL_Verbose, 0 or 1. 1 to display extended information.

- FilePar.par

    - par.gpnr.tol, convergence tolerance of the Projected Gradient algorithm;

    - par.gpnr.maxiter, maximum number of iterations of the Projected Gradient algorithm;

    - par.nwtp.maxiter, maximum number of iterations of the Projected Newton algorithm;

    - par.nwtp.tolrho, convergence tolerance of the Projected Newton algorithm;

    - par.cgn2d.tol, convergence tolerance of the Conjugate Gradient algorithm;

    - par.cgn2d.maxiter, maximum number of iterations of the Conjugate Gradient algorithm;

    - par.svd.svd, 0 or 1, 1 enable the use of the SVD algorithm;

    - par.svd.soglia, tolerance of the SVD algorithm;

    - par.upen.tol, convergence tolerance of UPEN algorithm;

- par.upen.iter, maximum number of iterations of the UPEN algorithm;

- par.upen.beta00, is a scale parameter that depends on the specific material, see references [1], [4] and [3] for a detailed discussion;

- par.upen.beta0, is a compliance floor that prevents division by zero, it should be small enough to prevent undersmoothing, and large enough to avoid oversmoothing;

- par.upen.beta_p, the compliance parameter for the local slope smoothing of the 2D distribution;

- par.upen.beta_c, the compliance parameter for the local curvature smoothing of the 2D distribution.

- FileSetInput.par

-Filenamedata, the name of the 2D data files;

-filenameTimeX, the filename of vectors of times of the first dimension;

-filenameTimeY, the filename of vector of times of the second dimension;

-nx, number of relaxations time bins along the first dimension used by Upen2D;

-ny, number of relaxations time bins along the second dimension used by Upen2D;

-T1min, minimum inversion time limit for the second inversion dimension, used if FL_InversionTimeLimits = 0;

-T1max, maximum inversion time limit for the second dimension, used if FL_InversionTimeLimits = 0;

-T2min, minimum inversion time limit for the first inversion dimension, used if FL_InversionTimeLimits = 0;

-T2max, maximum inversion time limit for the first inversion dimension, used if FL_InversionTimeLimits = 0.

Output files

When the computation terminates, many files are saved in the folder "\DATA\output_files\". The 2D matrix of the computed distribution is automatically saved in an ASCII file with the name "2D_Distribution.txt". The files with the coordinates (relaxation times) of the distribution in the two dimensions are saved under the folder "\DATA\output_files\" with the names "T1.txt" and "T2.txt. Also the times of the acquired data are saved, "t_1.txt" and "t_2.txt". At last, the list of parameters used and the positions coordinates of peaks are saved in the file "Parameters.txt" and the residual of the last computed 2D map is saved in the file "Residual.txt". In addition, at the end a number of plots are shown (they can be saved as ".fig" files as well as ".jpg" files). In particular,

the software shows: the projection of the 2D map along the two dimensions (Fig. 3a and Fig. 3b), the 2D distribution map (Fig. 3c) and the 3D distribution map (Fig. 3d).
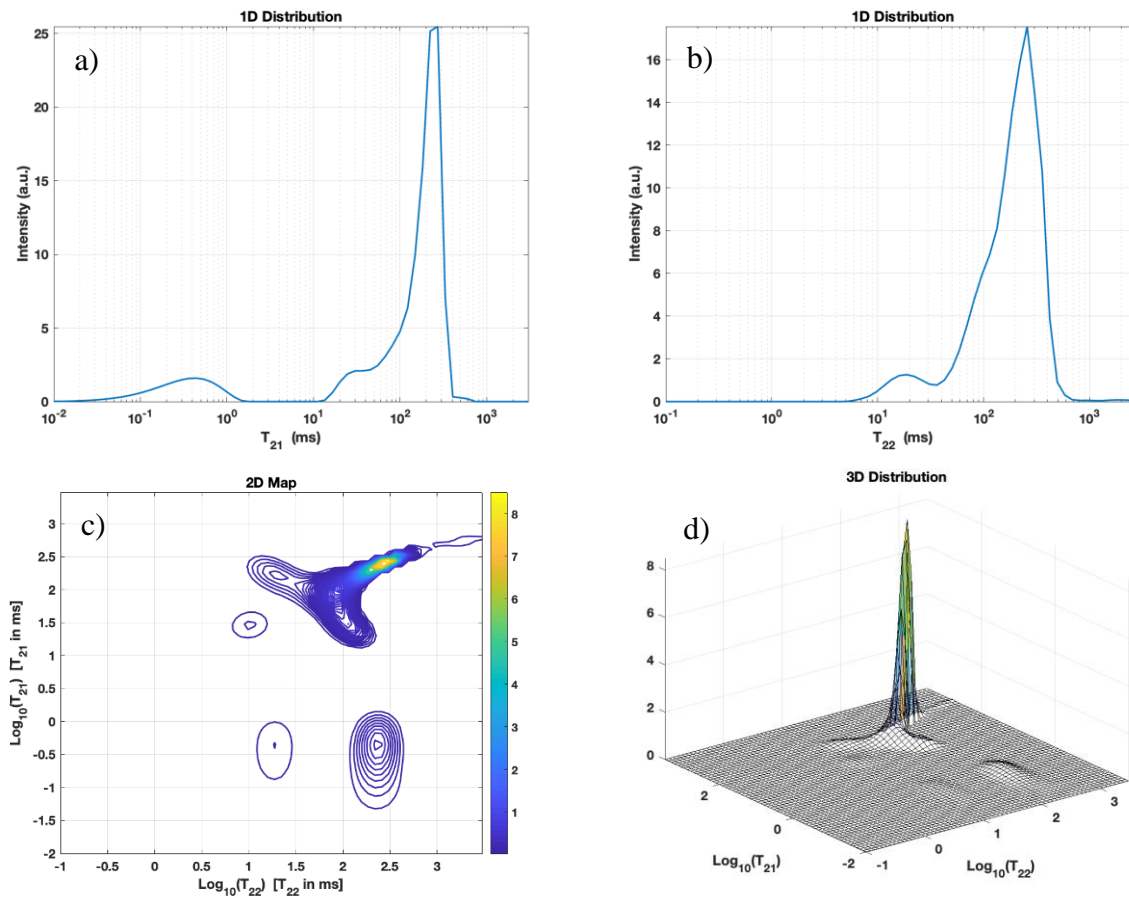

Figure 3 - Pictures of the plots automatically created by Upen2D.

4   EXAMPLES

In the DATA folder, there are three examples, one inside the sub-folder "T1-T2", a second one inside the subfolder "T2-T2" and a third in the subfolder "T1_T2_Synth".

Example1 folder "T1-T2"

In this example a sample of fresh cement paste was measured with an IR-CPMG 2D sequence. The number of the inversion points is 48, computed in geometric scale with a common ratio of 1.1896 in the range between 0.8 ms to 2800 ms. During the detection period, the TE was of 108 μs and the number of echoes was 1000. The repetition time was 3 s and number of scans 4.
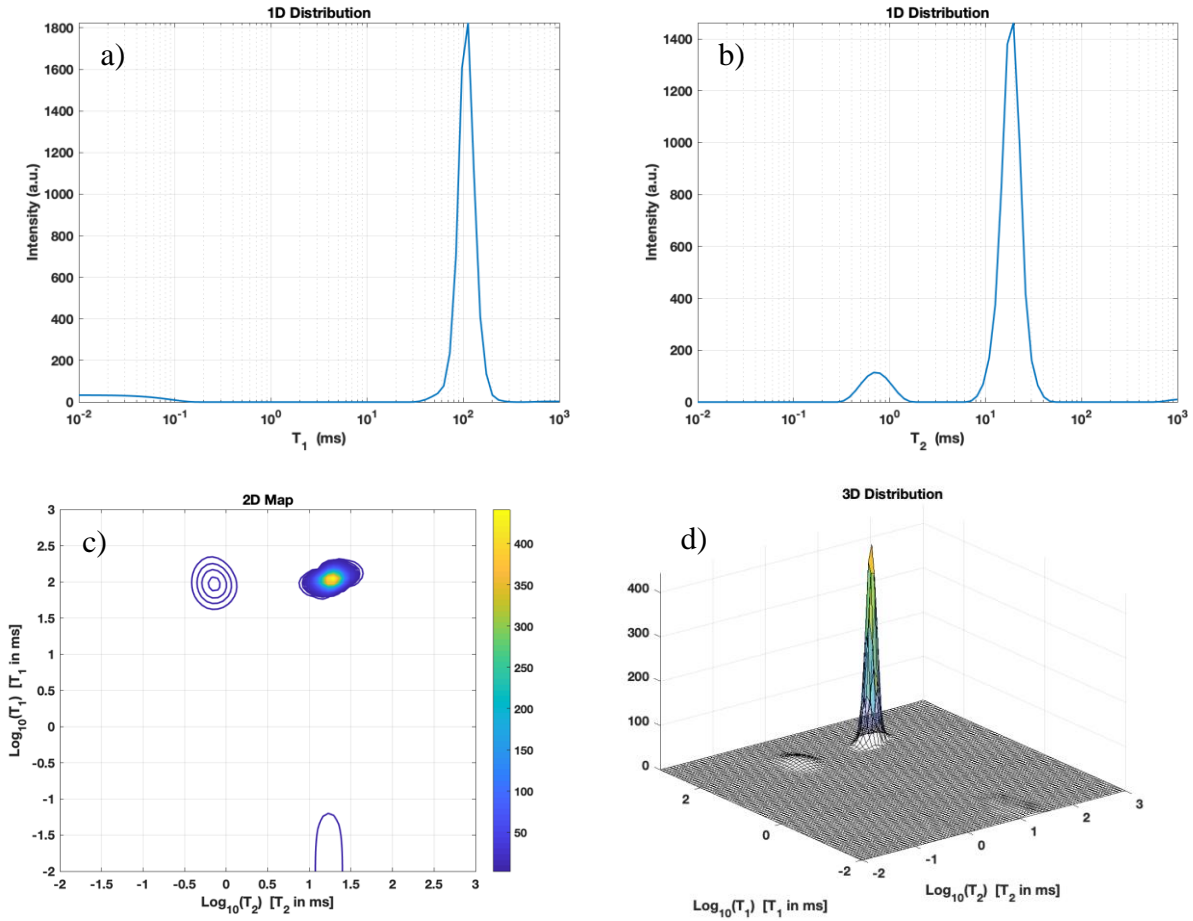
In Figure 4 the plots computed by Upen2DTool.

Figure 4 – "T1-T2" experiment. Projection along the T_1 (a) and T_2 (b) dimension, (c) Reconstructed 2D Contour relaxation map, (d) 3D map.

Example2 folder "T2-T2"

In this example a sample of Maastricht stone saturated with water was measured with a CPMG-CPMG 2D sequence. During the preparation period, the echo time (TE) was 6.61 ms and the number of echoes was 128. The mixing time lasts 100 s. During the detection period, the TE was of 300 μs and the number of echoes was 2800. The repetition time was 3 s and number of scans 16. The output files plots are those reported in the Figure 3 of the paragraph "Output files".

Example3 folder "T2-T2_Synth"

This is a T1-T2 synthetic case. We consider a reference map distribution of size 64x64 and use it to synthesize the simulated IR-CPMG data sequence with 128×2048 non uniformly distributed times in the interval [10^{-3}, 3 10^3] ms. The reference map distribution has the following three peaks:

	Height	T_1 (ms)	T_2 (ms)
P1	184.07	807.91	24.95
P2	227.55	6.56	2.48
P3	48.87	572.83	24.95

The noisy data s are obtained by adding Gaussian white noise of level 10^{-2} (SNR 20 db) to the synthetic data.

The peaks positions and height, computed by 2DUpenTool, are reported in the following table:

	Height	T_1 (ms)	T_2 (ms)
P1	176.84	959.47	24.95
P2	219.6	6.56	2.46
P3	42.17	572.83	24.95

The Relative Error (Err) and Root Mean Squared Error (RMSE) have the following values:

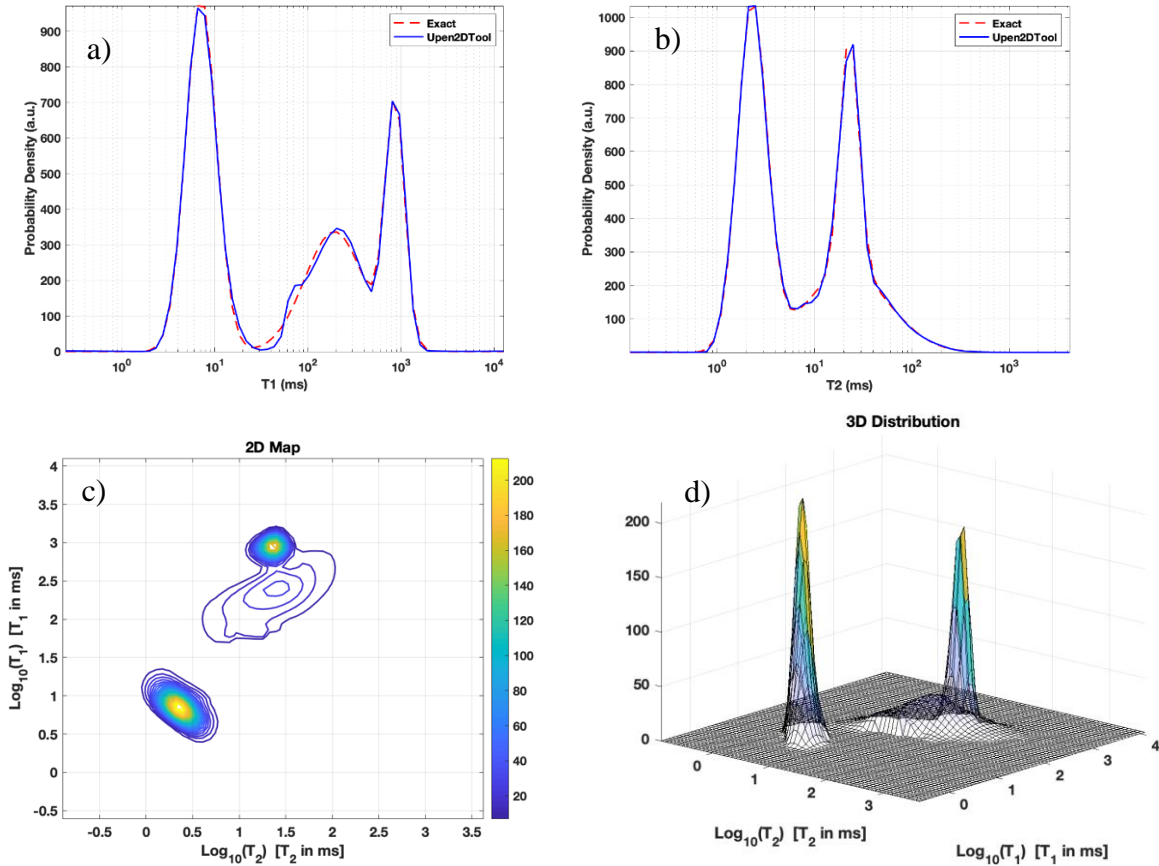Err = 0.113, RMSE= 1.720. The computed plots are shown in Figure 5.

Figure 5 - "T1-T2_synth" example. Projection along the (a) T_1 and (b) T_2 dimension, red line (true relaxation time distribution), blue line (computed relaxation time distribution), (c) Reconstructed 2D Contour relaxation map, (d) 3D map.

5   REFERENCES

[1] V. Bortolotti, R. J. S. Brown, P. Fantazzini, G. Landi, F. Zama, Inverse Problems 33 (1) (2016).

[2] B. Blüemich, Essential NMR, Springer-Verlag, 2005.

[3] V. Bortolotti, L. Brizi, P. Fantazzini, G. Landi, F. Zama, Filtering techniques for efficient inversion of two-dimensional nuclear magnetic resonance data, Journal of Physics: Conference Series 904 (1) (2017).

[4] V. Bortolotti, R.J.S. Brown, P. Fantazzini, G. Landi, F. Zama, UPEN2D: Improved 2DUPEN algorithm for inversion of two-dimensional NMR data, Microporous and Mesoporous Materials 269 195 – 198 (2018).