A short manual for 2d FLI - two-dimensional Fast Laplace Inversion

Schlumberger-Doll Research

April 16, 2003

2D FLI is an algorithm to perform two-dimensional numercial inverse Laplace transformation, developed at Schlumberger-Doll Research. This algorithm solves the integral equation:

$$M(\tau 1,\tau 2) = \iint dT1\,dT2 \cdot F(T1,T2)\exp(-\tau 1/T1)\exp(-\tau 2/T2) + Noise$$

where M is the signal as a function of $\tau 1$ and $\tau 2$ and the algorithm solves for F(T1,T2) subject to the non-negativity constraint, F >= 0.

Details were published in the following papers:

1. L. Venkataramanan et. al., IEEE Tran. SP. 50, 1017-1026 (May, 2002).
2. Y.-Q. Song, et al., J. Magn. Reson. 154, 261-268(2002).
3. M D Hurlimann and L Venkataramanan, J. Magn. Reson. 157, 31 (2002).
4. US Patent 6,462,542 B1, October 8, 2002

This document describes the use of the 2d FLI program.

How to use the program

Requirement:

Matlab5.3 and above,
optimization toolbox

Files in the distribution:

FLIv1.0:
- example:
  - example1.mat
  - example1.matRe.mat
- FLI.m
- FLIdrive.m
- FLIEstimate.m
- FLIminfun.m
- misc:
  - CalcT1T2r.m

ComputeProjections.m
FileRead.m
ReadT1T2.m

plotting:

PlotData2.m
PlotFEstimate.m
PlotFig.m
PlotT1T2.m

ReadME.pdf

Main files:

FLI

This is the main program for the algorithm.

1. read the data file with 2d data and parameters,
2. Set kernels, range of T1 and T2, digitization, choose regularization methods,
3. perform singular value decomposition (SVD) and compress the data,
4. optimize for the best F. The resulting density function (F) is stored in matlab variable FEst.

FLIEstimate.m

Core routine to optimize for F.

On the distribution CD, the compiled versions for Solaris (FLIEstimate.mexsol) and Window (FLIEstimate.dll) are included, instead of the source file. They are compiled for MATLAB 6 release 13.

FLIminfun.m

Minimization function, used in FLIEstimate.m and internally by matlab.

Example driver file:

FLIdrive.m

Set the path before running FLI.

Input data and parameters

Data:	A matrix of the 2 dimensional input data
NoiseStd:	A variable holding the variance of the noise
Tau_1:	A column vector holding the τ1 values, with the same length as width of the matrix **Data**.
Tau_2:	A column vector holding the t2 values, with the same length as the length of the matrix **Data**.

Example of the example1.mat (Output from matlab):

```
>> whos
  Name          Size                 Bytes  Class

  Data        2046x20               327360  double array
  NoiseStd       1x1                     8  double array
  Tau_1         20x1                   160  double array
  Tau_2       2046x1                 16368  double array
```

Execution of the program

In matlab, once the path is set to include the FLI.m file, command FLI will execute the program.

The command will first prompt for an input file. Pressing the return key without an input file will allow the program to analyze the data in the example folder.

The result of the inversion is stored in a file whose name is the input file name plus "Re.mat".

Main output variables

The resulting 2D spectrum

```
FEst          100x101          80800  double array
```

The resulting 2D fit to the data

```
Fitdata       2046x20          327360  double array
```

Kernel functions

```
K_1           20x101           16160  double array
K_2           2046x100         1636800  double array
Kernel_1      1x1              984  inline object
Kernel_2      1x1              974  inline object
```

The resulting T1/T2 ratio spectrum

```
T1T2r         1x100            800  double array
T1T2rdist     100x1            800  double array
```

The projected spectra of T1 and T2

```
T1            1x100            800  double array
```

T1_dist	1x100	800 double array
T2	1x100	800 double array
T2_dist	1x100	800 double array

The error in the 180 degree pulse

beta	1x1	8 double array

Plotting example

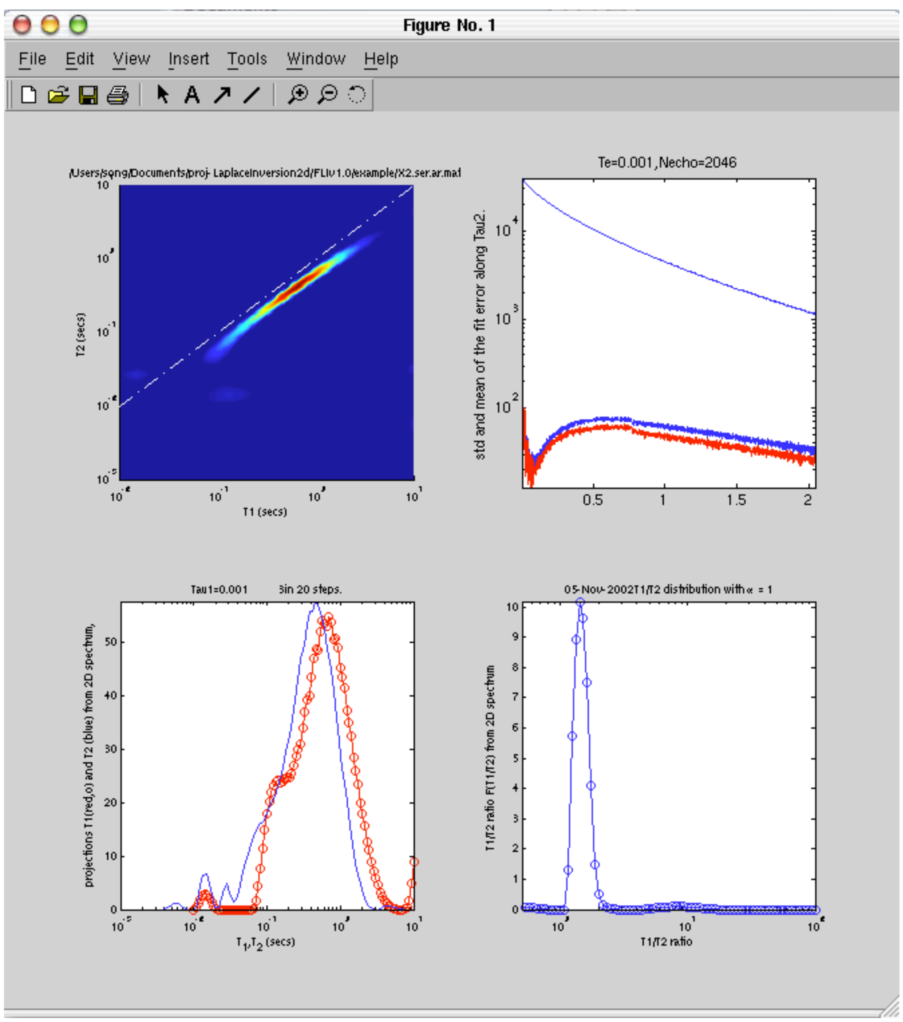
Command: PlotT1T2

Panel 1: the 2D spectrum, FEst, the line is for T1=T2.

Panel 2: fit error estimate: top blue line is the data for the shortest tau_1; the lower blue line variance of the fit error; the lower red line is the mean of the fit error.

Panel 3: projected spectra of T1 and T2. Title shows the range of Tau_1.

Panel 4: projected spectrum of T1/T2 ratio.



User Modifications of FLI

Modification to FLI.m

1. Modification for different regularization parameter and schemes

Regularization (alpha) can affect the final inversion result and its value depends on the signal-to-noise ratio of the data. For example, a small alpha can be used with data of high S/N to obtain details of the spectrum. However, when S/N is low, a larger alpha should be used often resulting in a smoother spectrum. In practice, one should perform the inversion with several values of alpha in order to gain confidence in the features observed in the inversion results.

The regularization is set by the following in FLI.m:

Alpha_Auto = 0:	fixed regularization, and the value is AlphaStart.
Alpha_Auto = 1:	automatic regularization using BRD method. For this method to work well, the Data should be free from systematic errors and NoiseStd reflect the variance of the random noise.
Alpha_Auto=2:	A series of regularization will be used to calculate the inversion. Then the heel of the fit error (c2) will be found to be the optimal regularization.

2. Modifications for different range of T1 and T2

These following variables should be set to reflect the range of T1 and T2 for the specific data set for optimal inversion result. For example, the minimum T1 (InitTime_T1) should not be smaller than the smallest Tau_1 and the maximum T1 (FinalTime_T1) should not be much larger than the maximum Tau_1. The similar rule should apply to T2.

The number of points for T1 and T2 should not be much smaller than 100. For example, Number_T1=20 might be too small to allow a good fit for some data.

The ranges of T1 and T2 are set by the following in FLI.m:

```
InitTime_T1 = .01;          % The initial value of T1 (in seconds)
FinalTime_T1 = 10;          % The final value of T1 (in seconds)
Number_T1 = 100;            % The number of T1's

InitTime_T2 = .001;         % The initial value of T2 (in seconds)
FinalTime_T2 = 10;          % The final value of T2(in seconds)
Number_T2 = 100;            % The number of T2's
```

3. Modifications for DC offset

The non-ideal inversion of the 180 degree pulse and the rf inhomogeneity would result in a DC offset. This offset can be estimated by adding a column in K_1. This feature is enabled by:

```
AllowDCOffset = 1;   % allow a dc offset
% AllowDCOffset = 0;          % no offset
```

4. Modifications for different kernels

The inversion kernels can be modified to suit the specific data. The kernels are defined as inline functions and calculated later.

```
%exponential decay
%Kernel_1 = inline('exp(- Tau * (1./ TimeConst))','Tau','TimeConst');

%inv recovery
Kernel_1 = inline('1- 2*exp( - Tau * (1./ TimeConst))','Tau','TimeConst');
```

5. Modification to neglect the non-negativity constraint

It is not implemented in the current software.

The principle of the modification is discussed in the reference 1 (equation 36).

Modification to FLIestimate.m, FLIminfun.m

User should not modify these files.

Modification to files in the directory misc

User should not modify these files.

Modification to files in the directory plotting

The files in those directories are provided as examples. Users may use them as is or as templates for their own routines to display the inversion results.

Contact information

L. Venkataramanan, lvenkataramanan@ridgefield.oilfield.slb.com
M. D. Hurlimann, mhurlimann@ridgefield.oilfield.slb.com
Y.-Q. Song, ysong@slb.com

Schlumberger-Doll Research, Old quarry road, Ridgefield, CT 06877