Inverse Laplace Transform of Real-valued Relaxation Data by a Regularized Non-negative Least Squares Method

Zhikun Cai*
Department of Nuclear, Plasma, and Radiological Engineering, University of Ilinois at Urbana-Champaign
(Dated: December 10, 2014)

I. PROBLEM DESCRIPTION

Relaxation, a common phenonemon in natural sciences, usually describes how an equilibrated system responses to some pertubation and returns to equilibrium again. For example, dielectric relaxation describes the delay in the dielectric constant of a material with respect to a changing electric field, while stress relaxation and structural relaxation provide information about the conformation change of material subject to applied force or environmental fluctuations. For some simple systems, the experimental or simulated relaxation data usually represents a simple exponential decay $\exp(-z_c t)$ in time domain, where z_c is interpreted as characteristic relaxation rate. However, for most of complex systems or simple systems under extreme environments, the relaxation phenomena are more complicated. One commonly accepted idea is to interpret the relaxation data as a superposition of unknown number of exponential decay modes with different decay rates. Mathematically, we can represent the relaxation data $F(t)$ as

$$F(t) = \sum_m f_m \exp(-z_m t), \tag{1}$$

where subscript m stands for different mode, f_m is physically nonnegative weighting factor, and z_m is the relaxation rate for each mode. It is the undetermined number of subscript m that makes it untable to fit data by choosing an arbitrary number of modes (unless physical argument can be provided).

However, if we look from another perspective and change subscript m from discrete values to continuous variable, we recover a familiar Laplace transform formula (except the conventional variable notation is reversed):

$$F(t) = \int_0^\infty dz\ f(z) \exp(-zt), \tag{2}$$

with constraint

$$f(z) \geq 0, \tag{3}$$

where $f(z)$ is interpreted as relaxation rate distribution function. Alternatively, if we define relaxation time as inverse relaxation rate, namely, $\tau = 1/z$, then we can rewrite Eq. (2) and (3) as

$$F(t) = \int_0^\infty d\tau\ \rho(\tau) \exp(-t/\tau), \quad \rho(\tau) \geq 0, \tag{4}$$

which is essentially the same as inverse Laplace transform. As a simple example, a single exponential decay in time domain is expected to give a delta function in z domain. Thus if there exists a reliable method to extract relaxation rate distribution function $f(z)$ by inverting relaxation data $F(t)$, then we are able to acquire sound information about relaxation behavior.

So the topic of this project is to develop a computer code to perform Inverse Laplace Transform (ILT) of discrete real-valued data sets, which are available from either simulations or experiments. Specifically, a regularized non-negative least squares method developed by Provencher [1] is implemented to address this problem.

II. METHOD

Inverse problems are generally ill-posed, and become ill-conditioned when translated to numerical representation. The solution may not be unique, and even the exact solution may not be desirable considering noises included in the data available. Thus special technique incorporating principle of parsimony or any statistical prior knowledge, and constraints introducing absolute prior knowledge becomes critical for seeking the optimal solution.

In the cases that analytical expression for the relaxation function is known or supposed, inverse Laplace transform can be accomplished by taking the Bromwich integral numerically over specific nodes. Among available methods, the Euler algorithm, Talbot algorithm and Gaver-Stehfest method have been proved effective in numerical inversion of Laplace transform of a given function [2]. Although these methods are sensitive to noise and are not suitable for inverting discrete data from simulation or experiment, they serve as useful tools for testing empirical model function and theoretical prediction.

When knowledge of analytical expression for the relaxation function acquired from either simulation or experiment is lacking, inverting tabular data sets becomes necessary. Under such circumstances, an effective method developed by Provencher [1] can be used. The method performs ILT by constructing a Regularized Nonnegative Least Squares (RNNLS) problem [1], which is further converted and solved by Least Distance Programming (LDP) algorithm [3]. Note that the method described in Ref. [1] can be used to address general inverse problems represented by linear algebraic equations besides ILT. In this project only what is essential to our topic is implemented. Below we outline the procedures.

* zcai12@illinois.edu

Algorithm Description:

1. Constructing a least squares problem: This algorithm begins with constructing a linear system by discretizing the integral in Eq. (2) according to some quadrature rule (like trapezoidal rule):

$$F(t_k) \approx \sum_{m=1}^{N_x} w_m e^{-t_k z_m} f(z_m), \quad k = 1, 2, \ldots, N_y \tag{5}$$

where w_m are the weights of the quadrature formula. The solution, $f(z_m)$, is then determined at the N_x grid points z_m ($N_x < N_y$). Eq. (5) can be written as a least squares problem:

$$||\boldsymbol{\eta} - \boldsymbol{C}\boldsymbol{x}||_2^2 = \text{mininum} \tag{6}$$

with constraint from Eq. (3)

$$\boldsymbol{x} \geq \boldsymbol{0} \tag{7}$$

where $N_y \times 1$ vector $\boldsymbol{\eta}$ represents data $F(t_k)$ to be inverted, and the $N_y \times N_x$ coefficient matrix $\boldsymbol{C}$ contains $w_m e^{-t_k z_m}$, and $N_x \times 1$ vector $\boldsymbol{x}$ contains all unknowns $f(z_m)$. The ill-posedness of this inverse problem is represented by the ill-conditioning of matrix $\boldsymbol{C}$.

2. Imposing regularization: To address the ill-conditioning of coefficient matrix $\boldsymbol{C}$, Ref. [1] proposes a regularization method so that the optimal solution satisfies

$$\begin{aligned} V(\alpha) &= ||\boldsymbol{\eta} - \boldsymbol{C}\boldsymbol{x}||_2^2 + \alpha^2 ||\boldsymbol{r} - \boldsymbol{R}\boldsymbol{x}||_2^2 \\ &= \text{mininum} \end{aligned} \tag{8}$$

where the second term on the right is called *regularizor*, and α is a regularization parameter determining the relative strength. The regularizor penalizes an $\boldsymbol{x}$ for deviations from behavior expected on the basis of statistical prior knowledge or parsimony. Several types of regularizors are outlined in Ref. [1]. In this project, the regularizor we choose to impose simplicity or smoothness to solution is

$$||\boldsymbol{r} - \boldsymbol{R}\boldsymbol{x}||_2^2 = \int_a^b [f''(z)]^2 \, dz. \tag{9}$$

When approximating the second difference by finite difference and using numerical integration, we reach $\boldsymbol{r} = 0$ and $\boldsymbol{R}$ is a $(N_{reg} = N_x + 2) \times N_x$ matrix of bandwidth three,

$$\boldsymbol{R} = \begin{bmatrix} 1 & & & & \\ -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \\ & & & & 1 \end{bmatrix} \tag{10}$$

where the first two rows are formed by enforcing extra two points below the interval to be zero. Similar implementation is also applied to the last two rows. Such extra requirement biases the solution to smoothly approach zero near the ends of the interval and is suitbale for our case based on physical prior knowledge.

By now the problem is set up as a RNNLS problem (Eq. (8)) subject to nonnegative constraint (Eq. (7)).

3. Converting RNNLS to LDP: The conversion from RNNLS to LDP can be achieved by a sequence of singular value decomposition and replacement of variable. First, the singular value decomposition of $\boldsymbol{R}$ is computed:

$$\boldsymbol{R} = \boldsymbol{U} \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} \boldsymbol{Z}^T \tag{11}$$

By changing variable (below subscript 3 is used just to catch up with derivation in Ref. [1])

$$\boldsymbol{x}_3 = \boldsymbol{Z}^T \boldsymbol{x}, \quad \boldsymbol{x} = \boldsymbol{Z}\boldsymbol{x}_3, \tag{12}$$

Eq. (8) becomes

$$\begin{aligned} &||\boldsymbol{\eta} - \boldsymbol{C}\boldsymbol{Z}\boldsymbol{x}_3||^2 + \alpha^2 ||\boldsymbol{r} - \boldsymbol{U} \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} \boldsymbol{Z}^T \boldsymbol{Z}\boldsymbol{x}_3||^2 \\ =&||\boldsymbol{\eta} - \boldsymbol{C}\boldsymbol{Z}\boldsymbol{x}_3||^2 + \alpha^2 ||\boldsymbol{r}_1 - \boldsymbol{H}_1\boldsymbol{x}_3||^2 + \alpha^2 ||\boldsymbol{r}_2||^2 = \text{mininum}. \end{aligned} \tag{13}$$

where the $N_x \times 1$ and $(N_{reg} - N_x) \times 1$ vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are defined by

$$\boldsymbol{U}^T \boldsymbol{r} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{bmatrix}. \tag{14}$$

And the constraint becomes

$$\boldsymbol{Z}\boldsymbol{x}_3 \geq \boldsymbol{0}. \tag{15}$$

The term $\alpha^2 ||\boldsymbol{r}_2||^2$ in Eq. (13) can be dropped because of independence of $\boldsymbol{x}_3$.

Next, we change variable $\boldsymbol{x}_3$ to $\boldsymbol{x}_4$ by

$$\boldsymbol{x}_4 = \boldsymbol{H}_1\boldsymbol{x}_3 - \boldsymbol{r}_1, \quad \boldsymbol{x}_3 = \boldsymbol{H}_1^{-1}(\boldsymbol{x}_4 + \boldsymbol{r}_1). \tag{16}$$

Eqs. (13) and (15) then become

$$||\boldsymbol{\eta} - \boldsymbol{C}\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{r}_1 - \boldsymbol{C}\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{x}_4||^2 + \alpha^2 ||\boldsymbol{x}_4||^2 = \text{minimum}, \tag{17}$$

$$\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{x}_4 \geq -\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{r}_1. \tag{18}$$

Then, we perform the second singular value decomposition

$$\boldsymbol{C}\boldsymbol{Z}\boldsymbol{H}_1^{-1} = \boldsymbol{Q}\boldsymbol{S}\boldsymbol{W}^T, \tag{19}$$

and make the change in variables

$$\boldsymbol{x}_5 = \boldsymbol{W}^T \boldsymbol{x}_4, \quad \boldsymbol{x}_4 = \boldsymbol{W}\boldsymbol{x}_5, \tag{20}$$

Eqs. (17) and (18) then become

$$||\boldsymbol{\gamma} - \boldsymbol{S}\boldsymbol{x}_5||^2 + \alpha^2||\boldsymbol{x}_5||^2 = \text{minimum}, \tag{21}$$

$$\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{W}\boldsymbol{x}_5 \geq -\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{r}_1, \tag{22}$$

where the $N_x \times 1$ vector $\boldsymbol{\gamma}$ is defined by

$$\boldsymbol{\gamma} = \boldsymbol{Q}^T(\boldsymbol{\eta} - \boldsymbol{C}\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{r}_1) \tag{23}$$

And Eqs. (21) is identical to the following equation except for some constants independent of $\boldsymbol{x}_5$:

$$||\tilde{\boldsymbol{\gamma}} - \tilde{\boldsymbol{S}}\boldsymbol{x}_5||^2 = \text{minimum}, \tag{24}$$

where $N_x \times 1$ vector $\tilde{\boldsymbol{\gamma}}$ and $N_x \times N_x$ diagonal matrix $\tilde{\boldsymbol{S}}$ are defined by

$$\tilde{\gamma}_j = \gamma_j S_{jj}(S_{jj} + \alpha^2)^{-1/2}, \quad \tilde{S}_{jj} = (S_{jj} + \alpha^2)^{1/2} \tag{25}$$

We obtain the final form of the problem by make the change of variables

$$\boldsymbol{\xi} = \tilde{\boldsymbol{S}}\boldsymbol{x}_5 - \tilde{\gamma}, \quad \boldsymbol{x}_5 = \tilde{\boldsymbol{S}}^{-1}(\boldsymbol{\xi} + \tilde{\gamma}) \tag{26}$$

whereby Eqs. (24) and (22) become

$$||\boldsymbol{\xi}||^2 = \text{minimum}, \tag{27}$$

$$\boldsymbol{Z}\boldsymbol{H}_1^{-1}\boldsymbol{W}\tilde{\boldsymbol{S}}^{-1}\boldsymbol{\xi} \geq -\boldsymbol{Z}\boldsymbol{H}_1^{-1}(\boldsymbol{r}_1 + \boldsymbol{W}\tilde{\boldsymbol{S}}^{-1}\tilde{\gamma}) \tag{28}$$

Thus an LDP probelm is finally constructed and is ready to be solved.

4. ***Solving LDP:*** The problem represented by Eqs. (27) and (28) can be solved by LDP algorithm developed by Lawson and Hanson (See Chap. 23 in Ref. [3]). Once the optimal solution $\boldsymbol{\xi}$ is obtained, the original solution is expressed as

$$\boldsymbol{x} = \boldsymbol{Z}\boldsymbol{H}_1^{-1}[\boldsymbol{W}\tilde{\boldsymbol{S}}^{-1}(\boldsymbol{\xi} + \tilde{\gamma}) + \boldsymbol{r}_1] \tag{29}$$

III. RESULTS

We implement the method described above in python code *ilt.py*, which calls support subroutine in *ldp.py*. And then two typical relaxation examples (multiple exponential decays and stretched exponential decay) as well as one real simulation case are demonstrated in *example.py*.

Note that from Eq (2) and (4), we obtain a relation

$$zf(z) = \tau\rho(\tau). \tag{30}$$

This relation gives a unified representation of information in $f(z)$ or $\rho(\tau)$, and more importantly, it implies more physical interpretations when plotted on logscale [4]. Thus, in this section, we will stick to ploting this quantity on logscale when presenting our results.

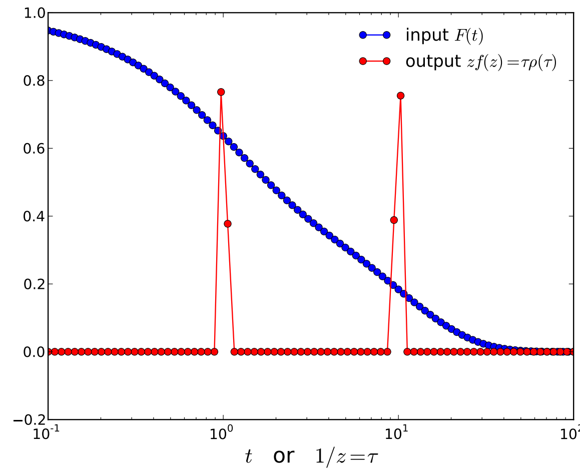

FIG. 1. Solution of ILT for multiple exponential decays. The input is 100 data points from $F(t) = 0.5\exp(-z_1 t) + 0.5\exp(-z_2 t)$, $z_1 = 0.1$, $z_2 = 1$, $10^{-1} < t < 10^2$, and regularization parameter $\alpha = 10^{-6}$. The output is divided by 5 for clarity when plotted with input.

Example 1: Multiple exponential decays

First, we apply the method to invert multiple exponential decay data. Physically assuming two relaxation modes, we choose the input as

$$F(t) = 0.5\exp(-z_1 t) + 0.5\exp(-z_2 t), \quad 10^{-1} < t < 10^2 \tag{31}$$

where $z1 = 0.1$, $z_2 = 1$. And the analytical solution of ILT is given by

$$f(z) = 0.5\delta(z - z_1) + 0.5\delta(z - z_2) \tag{32}$$

and

$$zf(z) = \tau\rho(\tau) = 0.5\tau\delta(\tau - \tau_1) + 0.5\tau\delta(\tau - \tau_2). \tag{33}$$

where $\tau_1 = 1/z_1 = 10$, $\tau_2 = 1/z_2 = 1$. In Figure 1, we plot the output solution together with input data points. As we can see, the numerical solution gives two peaks right at τ_1 and τ_2, and remains close to zero elsewhere. This is impressive since δ function is usually hard to resolve numerically. According to this demonstration, the method is able to resolve discrete exponential relaxation modes.

Example 2: Stretched exponential decay

Next, we consider another model widely used to analyze relaxation data, that is, stretched exponential decay

$$F(t) = \exp\left[-\left(\frac{t}{\tau_1}\right)^{\beta}\right], \quad 0 < \beta < 1. \tag{34}$$

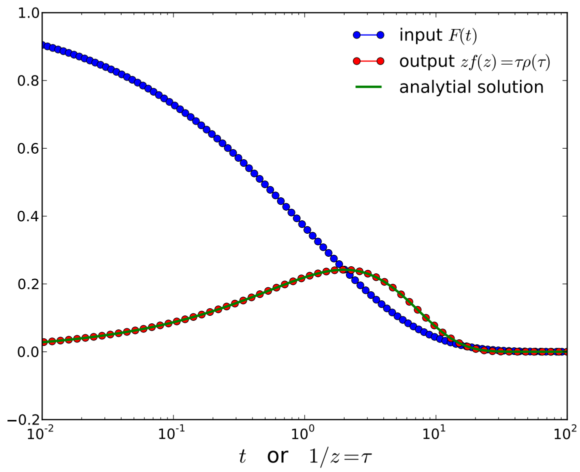

FIG. 2. Solution of ILT for stretched exponential decay. The input is 100 data points from $F(t) = \exp\left[-(t/\tau_1)^\beta\right]$, $\tau_1 = 0.1$, $\beta = 0.5$, $10^{-2} < t < 10^2$, and regularization parameter $\alpha = 10^{-6}$.

where τ_1 defines characteristic relaxation time. The analytical ILT of stretched exponential has no closed form except for $\beta = 0.5$. At $\beta = 0.5$, the analytical solution is given by

$$zf(z) = \tau\rho(\tau) = \frac{1}{2}\sqrt{\frac{\tau}{\pi\tau_1}}\exp\left(-\frac{\tau}{4\tau_1}\right) \tag{35}$$

So we choose parameter $\tau_1 = 1/z_1 = 1$, $\beta = 0.5$, and input 100 data points of $F(t)$ over time range $[10^{-2}, 10^2]$. The computed result is shown in Figure 2. As we we can see, the numerical result shows good agreement with analytical solution. Physically, stretched exponential is interpreted as superposition of relaxation modes over wide relaxation rate range. Thus there is a wide range of nonzero values in $zf(z) = \tau\rho(\tau)$.

Example 3: Simulation case (density-density correlation decay)

For the last example, we try to apply the method to real data from molecule dynamics (MD) simulation. In a simulation box, we simulate the newtonian dynamics of $N = 1700$ binary Lennard-Jones particles (A:B = 8:2) interacting with Lennard-Jones potential

$$V_{\alpha\beta}(r) = 4\epsilon_{\alpha\beta}\left[\left(\frac{\sigma_{\alpha\beta}}{r}\right)^{12} - \left(\frac{\sigma_{\alpha\beta}}{r}\right)^{6}\right], \quad \alpha, \beta \in A, B. \tag{36}$$

Then from the MD trajectories we compute the self density-density correlation function

$$F_s(\boldsymbol{k}, t) = \langle \frac{1}{N}\sum_{j=0}^{N} e^{i\boldsymbol{k}\cdot(\boldsymbol{r}_j(t)-\boldsymbol{r}_j(0))}\rangle \tag{37}$$

where $\boldsymbol{k}$ is wave vector and $\langle\cdots\rangle$ means ensemble average. The simulated system is nowadays a typical and simple enough benchmark system in the study of liquids and supercooled liquids [5]. And the computed correlation function is a good indicator of structure relaxation and is widely used to quantify characteristic relaxation time.

We intend not to include detailed description of simulation procedures here, but focus on the inversion of simulation data. Two input data files at two different simulated temperatures are already prepared, and they are named as *example_simu_highT.dat* and *example_simu_lowT.dat* respectively. At the high temperature the system exists as normal liquid with fast dynamics, while at the low temperature the system enter supercooled regime and dynamics is largely slowed down and heterogeneous.

The inversion results are shown tin Figure 3. At the high temperature panel of Figure 3, the input curve simply decay to zero very fast and only one single peak appears in the output. However, in the second panel, we see the input data shows feature of two-step decay and additional peaks emerges in the output. Such temperature dependence makes sense physically. Relaxation at high temperature is fast and almost homogeneous, so only one single peak representing one fast relaxation mode shows up in the relaxation rate/time distribution. As temperature goes down, dynamic heterogeneity starts to emerge. Beside the fast mode, additional slower relaxation modes appear and can be identified as extra peaks in the distribution of relaxation rate/time.

IV. SUMMARY

In this project, we implemented a regularized nonnegative least squares method developed by Provencher [1] to perform inverse Laplace transform of real-valued relaxation data. Specifically, we benchmark the method with two ideal data sets, including multiple exponential decays and stretched exponential, as well as one real data set from molecular dynamics simulation. Although some details remain to be improved, the good agreement of numerical results with analytical solutions or physical expectation is impressive and encouraging.

To further improve the code or the method, some aspects can be taken into account in later work. First, in this project we choose regularization parameter mainly based on physical understanding. But one should be aware that a too small regularization parameter doesn't help improve conditioning of inverse problem and smoothness is not sufficiently imposed, while a too large value tends to over-smooth the solution and suppress some detailed features. Thus, developing an effective strategy for choosing reasonable regularization parameter is critical to this method. Second, error bar estimation for the solution needs to be implemented and improved later. Third, besides using Eq (9) as regularizor, it is worth trying to develop new regularizors that are also able to effectively produce good solutions. Finally, it is

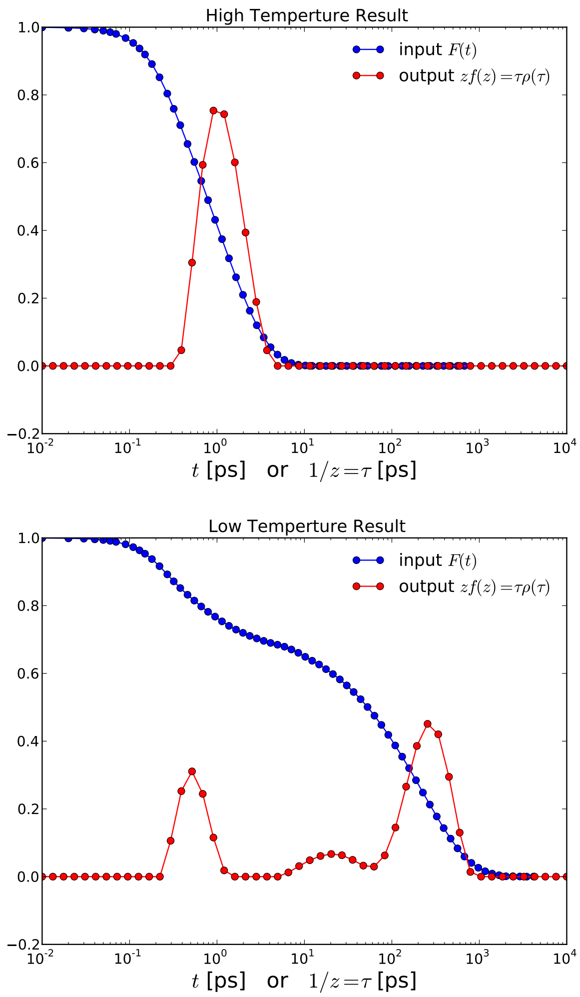

FIG. 3. Solutions of ILT for simulation data at high temperature (top panel) and low temperature (bottom panel). Due to data noise, the regularization parameter is increased to $\alpha = 4 \times 10^{-2}$.

also worth considering whether incorporating truncated SVD into this regularization method would absorb merits from both and lead to another effective method.

[1] S. Provencher, Computer Physics Communications **27**, 213 (1982).
[2] J. Abate and W. Whitt, INFORMS Journal on Computing **18**, 408 (2006).
[3] C. Lawson and R. Hanson, *Solving Least Squares Problems* (1974).
[4] Z. Cai and Y. Zhang, to be published.
[5] W. Kob and H. Andersen, Physical Review E **51** (1995).