AMSC 664 Final Report

Introduction of Regularization into Bi-exponential Magnetic Resonance Relaxometry Modeling to Break the CRLB Barrier in Parameter Estimation

Zezheng Song[1] and Advisor: Richard G. Spencer[2]

[1] *zsong001@umd.edu*

[1] *Department of Mathematics, University of Maryland, College Park, MD 20742, U.S.A*

[2] *spencerri@mail.nih.gov*

[2] *National Institute on Aging, National Institutes of Health, Baltimore, MD 21224, U.S.A*

In this project, we consider inverse problems and study the stability and regularization in the process of parameter estimation in important magnetic resonance (MR) models. In the literature of MR and nonlinear least squares analysis, the Cramér–Rao lower bound (CRLB) provides a bound for the variance of parameters to be estimated. However, controlling variance to be small while making a huge bias is not desirable, and vice versa. Therefore, it is preferable to introduce the mean squared error (MSE), which is a metrics combining the variance and bias, to study the performance of the estimators. We will calculate the bias, variance and MSE by solving a regularized non-linear least squares problem using Monte Carlo simulations, and check if it is possible to adjust the range of regularization parameter to reduce MSE below the theoretical CRLB. We then propose a strategy to provide optimal regularization parameter to reduce the MSE below CRLB for a prior range of model parameters.

Contents

1 Introduction

1.1 Background

Magnetic resonance imaging (MRI) is widely used to map physical characteristics of a suitable sample. MRI machines use strong magnetic fields, magnetic field gradients and radio waves to produce images about the internal situation of the body. The MR images contain important information about many parameters, such as the nuclear spin density, the spin-lattice relaxation time T_1, the spin-spin relaxation time T_2, and molecular motions, such as diffusion and perfusion[5]. We can change the imaging effects by suppressing or enhancing a certain set of parameters in an experiment, so the MR images can look very different with different data acquisition methods. In general, an MRI imaging can be made to a spatial map of the density of stationary spins or moving spins, or of relaxation times, or of the water diffusion coefficients. These are the subjects of study for subareas known as spectroscopic imaging, diffusion-weighted imaging, angiographic imaging, and functional imaging. In biophysics, nuclei have their own magnetic moments, so they will interact with magnetic fields that applied externally. Therefore, when nuclei are exposed to a static magnetic field, their proton spins will be aligned either parallel or antiparallel. If we apply radiosfrequency (RF) pulse to the nuclei, then it will tip the net magnetization away from the static magnetic field, into the transverse plane.

Then, one will observe a relaxation of the signal intensity within the transverse plane, after excited in the above manner. The transverse T_2 relaxation is the loss of transverse magnetization, which is a time constant characterizing the signal decay. To be more exact, T_2 denotes the amount of time it takes to remain 63% of the excited transverse magnetization. It turns out that T_2 provides important biophysical information. Because of this, obtaining or estimating these parameters is of great clinical importance, such as diagnosis of cancer and many other diseases.

In this project, we specifically consider the T_2-weighted imaging, also known as transverse decay time weighted imaging, where we assume that tissues can be approximated as having two dominant components. The bi-exponential model for this is stated explicitly as,

$$S(\mathbf{TE}; c_1, c_2, T_{21}, T_{22}) = c_1 \exp(-\mathbf{TE}/T_{21}) + c_2 \exp(-\mathbf{TE}/T_{22})$$

where T_{21}, T_{22} represent transverse relaxation time constants, and c_1, c_2 are component fractions. **TE** are the echo times when we make the measurements, and S denotes the signal intensity.

The bi-exponential model is used for assessment of cartilage degeneration in osteoarthritis, where the rapidly relaxing component is proteoglycan and slowly relaxing compoent is less-bound water. For the project, we assume we have information about c_1, c_2 and study the tissue characteristics by estimating the two relaxation times, T_{21} and T_{22}, which relate to tissue hydration and microscopic organization.

For example, in brain, the myelin corresponds to a shorter relaxing component T_{21}, while extracellular and interstitial water corresponds to longer relaxing component T_{22}. In many clinical situations, we wish to know how much myelin is in the brain. Therefore, at a certain location, we do an "imaging" version of the MRI models we are considering, where the model applies separately at every imaging pixel. Longer T_2 means "more fluid-like, and less solid-like". Therefore, the fluidity or mobility of a component can be judged by its T_2 value. That is, we have to be able to distinguish the two relaxing components based on their relaxation times in order to make statements about c_1 as belonging to the shorter T_2 component, and c_2 as belonging

to the longer T_2 component.

In the procedure of parameter estimation, an important aspect is to find the limiting performance of estimators, and solve for efficient estimators that achieve this performance. The Cramér–Rao lower bound (CRLB) gives an explicit lower bound on the variance of any unbiased estimators for fixed but unknown parameters, and hence a benchmark against which we could compare the performance of any estimators[8]. In practice, we also compare with the nonlinear least squares (NLLS) analysis, since CRLB will generally be found to be close to NLLS, and NLLS can have little bias when non-regularized. Here, we state the definition of CRLB since it is a benchmark against which we could compare the performance of estimators.

Definition 1.1 (Cramér–Rao lower bound)**.** Consider the parameter vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_d]^T \in \mathbb{R}^d$, with probability density function $f(x; \boldsymbol{\theta})$, and the Fisher information matrix is a $d \times d$ matrix given by $I_{m,k} = -\mathrm{E}\left[\frac{\partial^2}{\partial\theta_m \partial\theta_k} \log f(x; \boldsymbol{\theta})\right]$. Let $\boldsymbol{T}(X)$ be an estimator, and denote its expectation vector $\mathrm{E}[\boldsymbol{T}(X)]$ by $\boldsymbol{\psi}(\boldsymbol{\theta})$, then the Cramér–Rao bound states that the covariance matrix of $\boldsymbol{T}(X)$ satisfies

$$\mathrm{cov}_{\boldsymbol{\theta}}(\boldsymbol{T}(X)) \geq \frac{\partial \boldsymbol{\psi}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} [I(\boldsymbol{\theta})]^{-1} \left(\frac{\partial \boldsymbol{\psi}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \tag{1.1}$$

The calculation of CRLB has been done for different noise distributions. For example, an explicit calculation of CRLB of noncentral χ-distribution has been done by Bouhrara and Spencer[1].

Although the CRLB is a popular performance benchmark, the limitation is also obvious: estimators with small variance usually have larger bias, i.e., far away from the true parameters. Therefore, we use mean squared error (MSE) as the metrics, which measures both the variance and bias of estimators. By definition, the MSE of an estimator $\hat{\theta}$ with respect to an unknown parameter θ is

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{E}_\theta\left[(\hat{\theta} - \theta)^2\right] = \mathrm{Var}_\theta(\hat{\theta}) + \mathrm{Bias}_\theta(\hat{\theta}, \theta)^2$$

In addition, Eldar, et al. (2006)[3] showed that MSE could be reduced below the CRLB if we relax the requirement of unbiasedness of estimators. Therefore, we artificially introduce bias by adding regularization to the estimation procedure, which can decrease the MSE below the conventional lower limit of CRLB, provided that we use appropriate amount of regularization. Mathematically, we formulate the following nonlinear least squares optimization problem

$$\hat{\mathbf{p}}_\lambda = \underset{\mathbf{p}}{\mathrm{argmin}} \left\{ \|\mathbf{G}(\mathbf{p}) - \mathbf{d}\|_2^2 + \lambda^2 \|\mathbf{L}(\mathbf{p} - \mathbf{p_0})\|_2^2 \right\}, \tag{1.2}$$

or equivalently,

$$\hat{\mathbf{p}}_\lambda = \underset{\mathbf{p}}{\mathrm{argmin}} \left\| \begin{pmatrix} \mathbf{G}(\mathbf{p}) \\ \lambda \mathbf{L}\mathbf{p} \end{pmatrix} - \begin{pmatrix} \mathbf{d} \\ \lambda \mathbf{L}\mathbf{p_0} \end{pmatrix} \right\|_2^2 \equiv \underset{\mathbf{p}}{\mathrm{argmin}} \left\| \tilde{\mathbf{G}}_\lambda(\mathbf{p}) - \tilde{\mathbf{d}} \right\|_2^2$$

where $\mathbf{G}(\mathbf{p})$ is a known model nonlinear in parameter $\mathbf{p} \in \mathbb{R}^s$, $\mathbf{p}_0$ a prior estimated parameter of the problem, $\mathbf{d} \in \mathbb{R}^n$ is the signal vector ($\mathbf{d} = \mathbf{G}(\mathbf{p}) + \nu$, where the noise ν is assumed to follow the normal distribution $\mathcal{N}(0, \sigma^2)$). Therefore, $\widetilde{\mathbf{G}}_\lambda(\mathbf{p})$ and $\widetilde{\mathbf{d}}$ are of size $(n + s) \times 1$. $\mathbf{L}$ is the weighting diagonal matrix which serves to scale the order of elements of the estimators, since different model parameters have different units and order of magnitude.

To find the estimators that fit the biomedical MRI data best, we need to solve the regularized nonlinear least squares problem (1.2). The introduction of Tikhonov regularization will improve the stability of the kernel, but the solution it produces will be distant from the solution of the original problem. In other words,

even without the presence of noises, it is inevitable to introduce bias by regularization.

1.2 Project Objectives

The goal of the project is to provide a method of analysis that yields, on average, a lower MSE than the conventional method of CRLB and NLLS. Therefore, the goal is very practical, that is to provide better parameter estimates than CRLB and NLLS does. In particular, we study the bias-variance trade-off of estimators in mono-exponential and bi-exponential model, with additive Gaussian noise. By comparing the MSE with CRLB, we could evaluate a suitable degree of regularization, through the value of λ, that we should introduce to achieve MSE less than CRLB. At the same time, we evaluate its robustness with respect to range of possible model parameter values. In practical applications, we will assume the realistic case of having some a priori knowledge of plausible parameter ranges.

2 Approach

2.1 General Idea

The general idea to achieve our goal is stated as following: based on the combination of each MRI model and each noise distribution, we generate synthetic data. For each combination of MRI model and noise distribution, we first compute the CRLB, which is the theoretical lower bound of the variance of the estimators for model parameters. To obtain the estimators, we formulate a constrained optimization problem (1.2), and the solution to which will be the estimators in that particular noise realization.

Then, for each regularization parameter λ in a specific range, we run a number of noise realizations to solve for the estimators (solving the constrained optimization problem (1.2)), then calculate the mean and variance of desired estimators within each run. Then, we calculate MSE by its relation to variance and bias. Finally, we obtain a set of MSE corresponding to different λ's, so we are able to compare the MSE with the CRLB, and observe which range of λ makes the MSE lower than the CRLB, i.e., regularized least squares estimators perform better than the theoretical lower bound. A detailed description of this is stated in Algorithm 1.

As stated, the key problem is solving the regularized least squares estimators. This ends up to solve a constrained nonconvex optimization problem, due to the fact that we have the model function nonlinear in the parameters. Since this problem is extremely case-dependent, and nonconvex optimization is a broad topic, we will present here several nonconvex optimization algorithms that we implement.

2.2 Algorithms and Guarantee Analysis

Algorithm 1: Monte Carlo Simulation

Input: Sequence of regularization parameters $\{\lambda_i\}_{i=1}^{M}$, number of noise realizations N, weighting matrix $\mathbf{L}$, MRI model $\mathbf{G}$;

Output: Estimators of parameters sol_vec;

Data: MRI model measurements $\mathbf{d}$;

for $i = 1$ **to** M **do**

  for $j = 1$ **to** N **do**

    Using optimization solvers to solve $\hat{\mathbf{p}}_\lambda = \underset{\mathbf{p}}{\operatorname{argmin}} \left\{ \|\mathbf{G}(\mathbf{p}) - \mathbf{d}(:, j)\|_2^2 + \lambda_i^2 \|\mathbf{L}\mathbf{p}\|_2^2 \right\}$;

    $\text{sol\_vec}_i(j, :) = (\hat{\mathbf{p}}_\lambda)^\top$;

  end

end

The point of Algorithm 1 is to solve for the synthetic data on various noise realizations, and obtain estimators of the model, with respect to different noise realizations. Next, we compute the bias, variance and MSE from all estimators to proceed on our analysis. Since our MSE is obtained using Monte Carlo methods, which contain only finitely many samples, it is important to perform a guarantee analysis to understand how many noise realizations are enough for our purpose.

By definition, the MSE of an estimator $\hat{\theta}$ with respect to an unknown parameter θ is

$$\text{MSE}(\hat{\theta}) = \text{E}_\theta \left[(\hat{\theta} - \theta)^2 \right]$$

and we let X denote the random variable $(\hat{\theta} - \theta)^2$. Then, by the generalized finite-sample Chebyshev's inequality by Kaban, et al.(2012)[4], we have

$$P(|X - m| \geq ks) \leq \frac{1}{N+1} \left\lfloor \frac{N+1}{N} \left(\frac{N-1}{k^2} + 1 \right) \right\rfloor$$

where m is the sample mean, s is the sample standard deviation, k is an arbitrary constant, and N is the number of times we sample X. For $N = 100$, the 95% confidence interval is approximately ± 4.9595 standard deviations[4], which is acceptable to us. Unless otherwise specified, we will implement 100 noise realizations to approximate MSE for the rest of the report.

2.2.1 Grid Search

For this algorithm, we use the bi-exponential model to illustrate the idea. First, assume true parameters to be $\mathbf{p} = (c_1, c_2, T_{21}, T_{22}) = (0.7, 0.3, 50, 90)$, echo time $\mathbf{TE} = 8 : 8 : 256$ milli-seconds , and the signal-to-noise ratio (SNR) to be 1000 and, say, we implement 100 noise realizations. This step is to generate the synthetic data $\mathbf{d}$ of size 32×100.

For the grid search, we build a meshgrid in T_{21} and T_{22}, i.e., a square domain $[0, 200] \times [0, 200]$ with sufficiently fine grid. We choose 60 different regularization parameter λ's that lie uniformly within $[10^{-4}, 10^2]$. Then, for each λ, each noise realization, and for each pair of grid points of (T_{21}, T_{22}), we have a linear problem in c_1 and c_2, so we use the MATLAB optimization solver `quadprog` to find c_1 and c_2 that minimizes Eq. (1.2). Finally, we calculate the bias, variance and MSE of estimators, for each λ.

2.2.2 Gradient Descent

The method of gradient descent is a common iterative, first-order gradient based method. It is based on the idea that the loss function $F(x)$ decreases most quickly when moving against the direction of gradient of F at a, i.e., $-\nabla F(a)$. The update formula of parameters is given by:

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

In addition, if the loss function F is convex and its gradient ∇F is Lipschitz continuous, then we are guaranteed to have a local minimum, provided the learning rate γ satisfies the Wolfe conditions. In order to have optimal step length γ, we could combine this method with the line search method. We also use backtracking approach to ensure that the candidate step lengths are chosen appropriately.

2.2.3 VARPRO

The VARPRO (variable projection) algorithm is a good option for solving nonlinear least squares problems, where many parameters are linear. The idea behind this algorithm is to separate the nonlinear and linear variables in the curve fitting loss function, and use finite differences to approximate the Jacobian of loss function. O'Leary, et al. (2013)[7] contributed to this algorithm a lot, and provided a robust implementation of this algorithm written in MATLAB. We implement the VARPRO algorithm developed by O'Leary referenced at `http://www.cs.umd.edu/~oleary/software/varpro/` and compare it to other numerical algorithms. The regularized version of VARPRO is non-trivial and takes a lot of extra effort, so this will be a potential future work.

2.2.4 Mesh Adaptive Direct Search (MADS)

In pattern-search methods, we choose a set of search directions at each iterate and evaluate the objective function at a given step length along each of these directions. The potential candidate points form a "stencil" around the current iterate. When one of the points in the frame gives a significant decrease in the objective function, we move to that point. Otherwise, if no points on the stencil gives huge decrease in value of objective function, we stay at the current iterate and reduce the step length. An advantage of this type of method is that it is possible to show global convergence results.

In particular, in our example, in the T_{21} and T_{22} grid, we use the estimate obtained from section 2.2.1, then starting from that estimate, we implement our MADS method. That is, beginning from the warm start obtained from 2.2.1, we only need two or three steps to achieve the desired accuracy. A detailed description

of the algorithm is in algorithm (2)[6].

Algorithm 2: Mesh Adaptive Direct Search (MADS)

Given convergence tolerance γ_{tol}, contraction parameter θ_{max},
sufficient decrease function $\rho : [0, \infty) \to \mathbb{R}$ with $\rho(t)$ an increasing function
of t and $\rho(t)/t \to 0$ as $t \downarrow 0$;
Choose initial point x_0, initial step length $\gamma_0 > \gamma_{tol}$, initial direction set $\mathcal{D}_0$
for $k = 1, 2, \ldots$ **do**
  if $\gamma_k \leq \gamma_{tol}$ **then**
    stop;
  end
  if $f(x_k + \gamma_k p_k) < f(x_k) - \rho(\gamma_k)$ *for some* $p_k \in \mathcal{D}_0$ **then**
    Set $x_{k+1} \leftarrow x_k + \gamma_k p_k$ for some such p_k;
    Set $\gamma_{k+1} \leftarrow \varphi_k \gamma_k$ for some such $\varphi_k \geq 1$;
  else
    Set $x_{k+1} \leftarrow x_k$;
    Set $\gamma_{k+1} \leftarrow \theta_k \gamma_k$, where $0 < \theta_k \leq \theta_{\max} < 1$
  end
end

For the implementation, we choose $\gamma_{tol} = 1e{-}4$, $\gamma_0 = 1$, $\theta = 0.5$, $\mathcal{D}_0 = \{[0, 0.5], [0, -0.5], [-0.5, 0], [0.5, 0]\}$, $\phi = 1.5$, and $\rho(t) = t^{(3/2)}$. However, since this algorithm is activated by grid search, it is very time consuming (solving hundreds and thousands of least squares problems, depending on the density of $T_{21} - T_{22}$ grid).

2.2.5 Gauss-Newton Method and Levenberg-Marquardt Method

Gauss-Newton method can be regarded as the Newton's method with line search, which uses the following approximation for the Hessian matrix:

$$\nabla^2 f_k \approx J_k^T J_k, \tag{2.1}$$

where J_k is the Jacobian matrix. The search direction p_k then satisfies that

$$J_k^T J_k p_k = -J_k^T r_k \tag{2.2}$$

Then, it is easily shown that when the Jacobian J_k is of full rank, and gradient ∇f_k is nonzero, then p_k is indeed a descent direction for f, namely,

$$(p_k)^T \nabla f_k = (p_k)^T J_k^T r_k = -(p_k)^T J_k^T J_k p_k = -\|J_k p_k\|_2^2 \leq 0 \tag{2.3}$$

where r_k is the residual. To implement Gauss-Newton, similarly to gradient descent, we also need to perform a line search in the direction p_k for step length α, which needs to satisfy the Wolfe conditions.

Levenberg-Marquardt method is a very famous nonlinear least squares method. The Levenberg-Marquardt method uses the same approximation of the Hessian matrix as in Eq.(2.1), but replaces the line search method with a trust-region method.

3 Mono-exponential Model

3.1 Nonlinear Least Squares Problem and Analytic Computation

To establish methodology with a simpler but still-relevant problem, we first analyze the simpler but still-important mono-exponential model

$$S(\mathbf{TE}; c, T) = c\exp(-\mathbf{TE}/T) \tag{3.1}$$

where $\mathbf{TE} = \{TE_k\}_{k=1}^n$ is the echo time. In the actual signal measured, we have additive white Gaussian noise, namely,

$$S(\mathbf{TE}; c, T) = c\exp(-\mathbf{TE}/T) + \nu \tag{3.2}$$

where ν is the noise with normal distribution, i.e., $\nu \sim \mathcal{N}(0, \sigma^2)$.
Now, we set up the constrained optimization problem as following

$$\begin{aligned} \underset{c,T}{\text{argmin}} \quad & \left\| c\exp(-\mathbf{TE}/T) - \mathbf{d} \right\|_2^2 \\ \text{s.t.} \quad & c \geq 0, T \geq 0 \end{aligned} \tag{3.3}$$

where $\mathbf{TE} = \{TE_k\}_{k=1}^n$ is the echo time, $\mathbf{d} = \{d_k\}_{k=1}^n$ is the signal data of one noise realization, i.e., d_k is the datum we collect at time TE_k, for $k = 1, 2, \ldots, n$. The objective function in (3.3) can be expanded as

$$\begin{aligned} L &= \left\| c\exp(-\mathbf{TE}/T) - \mathbf{d} \right\|_2^2 \\ &= \sum_{k=1}^n \left[c\exp(-TE_k/T) - d_k \right]^2 \\ &= \sum_{k=1}^n \left[c^2\exp(-2TE_k/T) - 2c\exp(-TE_k/T)d_k + d_k^2 \right] \\ &= \left[\sum_{k=1}^n \exp(-2TE_k/T) \right] c^2 - \left[\sum_{k=1}^n 2\exp(-TE_k/T)d_k \right] c + \sum_{k=1}^n d_k^2 \end{aligned} \tag{3.4}$$

Assuming linear progression of $\mathbf{TE} = \{kTE\}_{k=1}^n$, we do a change of variable, i.e., let $Q = \exp(-TE/T)$, then we have

$$\begin{aligned} L &= \left\| c\exp(-\mathbf{TE}/T) - \mathbf{d} \right\|_2^2 \\ &= \left[Q^2 + (Q^2)^2 + \ldots + (Q^n)^2 \right] c^2 - 2\left[\sum_{k=1}^n Q^k d_k \right] c + \sum_{k=1}^n d_k^2 \\ &= \frac{Q^2(1-Q^{2n})}{1-Q^2} c^2 - 2\left[\sum_{k=1}^n Q^k d_k \right] c + \sum_{k=1}^n d_k^2 \end{aligned} \tag{3.5}$$

Now, to minimize the loss function L, we set $\frac{dL}{dQ} = 0$, and obtain the following quadratic equation in c,

$$\left[2nQ^{2n+3} - \left(2n+2\right)Q^{2n+1} + 2Q \right] c^2 - 2\left[\sum_{k=1}^n kQ^{k-1} d_k \right] \left(1-Q^2\right)^2 c = 0 \tag{3.6}$$

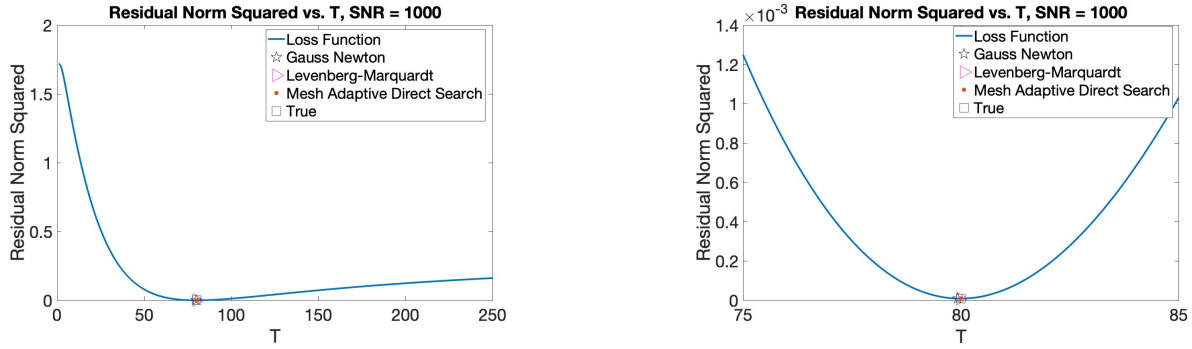
Eq.(3.6) has solution of

$$c_1 = 0, \qquad c_2 = \frac{2\Big[\sum_{k=1}^{n} kQ^{k-1}d_k\Big]\Big(1-Q^2\Big)^2}{\Big[2nQ^{2n+3} - \Big(2n+2\Big)Q^{2n+1} + 2Q\Big]} \tag{3.7}$$

Numerically, we use the solvers to solve for c and T in (3.3). To generate the synthetic data, we use (4.2), where we choose $c = 0.7$, $T = 80$, $\mathbf{TE} = 8:8:128$ milliseconds, $S = c\exp(-\mathbf{TE}/T)$ and $SNR = 1000$, so the standard deviation of noise is

$$\sigma = \frac{1}{SNR}$$

To compare the performance of each solver, we use parameters set up as above. For the specific noise realization, we make the following plots of loss function (3.3) value vs. T, and label the estimators returned by different solvers.



(a) (b)

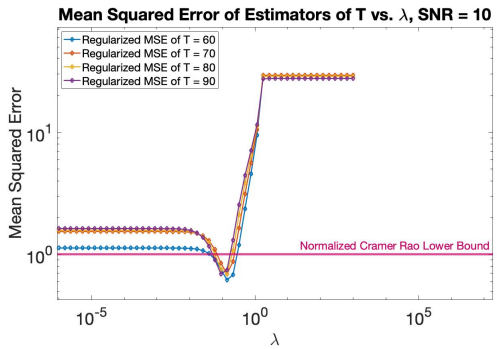
Figure 1: Landscape of $L = \left\|c\exp(-\mathbf{TE}/T) - \mathbf{d}\right\|_2^2, c = 0.7, T = 80, \mathbf{TE} = 8:8:128$ ms. Left Figure: $T \in [0, 250]$ with 100000 points uniformly distributed. Right Figure: Zoom in view, $T \in [75, 85]$ with 1000 points uniformly distributed.

3.2 Regularized Nonlinear Least Squares Estimation

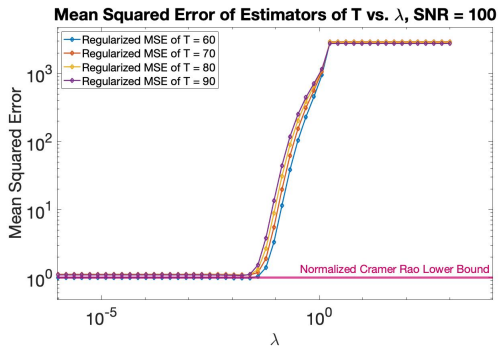
In order to add robustness to the estimators and lower the MSE below the conventional CRLB, we introduce a regularization term and obtain the following optimization problem,

$$\begin{aligned} \underset{c,T}{\operatorname{argmin}} \quad & \left\| c\exp(-\mathbf{TE}/T) - \mathbf{d} \right\|_2^2 + \lambda^2 \left\| (c,T)^\top \right\|_2^2 \\ \text{s.t.} \quad & c \geq 0, T \geq 0 \end{aligned} \tag{3.8}$$

where **TE** is the echo time, **d** is the signal data, λ is the regularization parameter. To choose the desired range of λ which makes MSE below CRLB, we plot the regularized MSE of estimators of T vs. λ.



(a) $SNR = 10$



(b) $SNR = 100$

Figure 2: Mean squared error of estimators of T vs. λ, where $c = 0.7$, $T = 60:10:90$, $\mathbf{TE} = 8:8:128$ ms. Left: $SNR = 10$. Right: $SNR = 100$.

As we see from Figure 2, we observe a consistent range of values of λ's that make MSE of T below the corresponding CRLB, in the $SNR = 10$ case. On the other hand, we do not observe MSE below CRLB for any regularization parameter λ in the case of $SNR = 100$. The explanation for this is when the noise is small, then the Levenberg-Marquardt returns the optimal estimator, so the variance of the estimators are already very close to CRLB even without the introduction of regularization. Plus, by adding regularization, we will introduce more bias into the estimators.

4 Bi-exponential Model

Now, bi-exponential analysis proceeds analogously. We consider the following model

$$S(\mathbf{TE}; c_1, c_2, T_{21}, T_{22}) = c_1 \exp(-\mathbf{TE}/T_{21}) + c_2 \exp(-\mathbf{TE}/T_{22}) \tag{4.1}$$

where **TE** is the echo time, and c_1, c_2, T_{21}, T_{22} are parameters to be estimated. In spectroscopy, which we consider here, the noise model is additive white Gaussian noise, namely,

$$S(\mathbf{TE}; c_1, c_2, T_{21}, T_{22}) = c_1 \exp(-\mathbf{TE}/T_{21}) + c_2 \exp(-\mathbf{TE}/T_{22}) + \nu \tag{4.2}$$

where ν is the noise with the normal distribution $\nu \sim \mathcal{N}(0, \sigma^2)$. Now, we set up the unregularized optimization problem as follows

$$\begin{aligned} \underset{c_1, c_2, T_{21}, T_{22}}{\operatorname{argmin}} \quad & \left\| c_1 \exp\left(-\mathbf{TE}/T_{21}\right) + c_2 \exp\left(-\mathbf{TE}/T_{22}\right) - \mathbf{d} \right\|_2^2 \\ \text{s.t.} \quad & c_1 \geq 0 \\ & c_2 \geq 0 \\ & T_{21} \geq 0 \\ & T_{22} \geq 0 \end{aligned} \tag{4.3}$$

where **TE** is the echo time, **d** is the signal data of one noise realization. To recover the parameters of the underlying bi-exponential model accurately, we need to study and compare the performance of these solvers in Section 2.2. The criteria that we use to compare these solvers are

- If the solver achieves a local, global minimum of loss function or not, and
- At the same condition number of the Jacobian matrix **J** of bi-exponential model, which solver returns estimators that have small variance, and
- At the same SNR in the synthetic data, which solver returns estimators that have small variance, and
- The processing time of the solvers.

Remark: Different from the goal of our project, which is to estimate T_{21}, T_{22} given known c_1, c_2, this section tests the accuracy and robustness of each optimization solver, so the tests in this section are implemented for four-parameter estimation (Estimating for c_1, c_2, T_{21}, T_{22} simultaneously).

4.1 Global Optimum Check

In principle, we prefer solvers that could find a global optimum of the loss function, and that are more stable and fast. By Figure 3, we find that the loss function (4.3) has a relatively smooth landscape, but there is a huge area near the global optimum where the landscape is almost flat. For this specific noise realization, all solvers could reach the global optimum well. We plot the projection of estimators by individual solvers on

the $T_{21} - T_{22}$ plane, where the green point is the true parameter $(50, 90)$, the closest to it is the estimator by Levenberg-Marquardt, then followed by MADS and VARPRO, respectively.

Here, we consider the model that $\mathbf{G}(\mathbf{p}) = 0.7\exp(-\mathbf{TE}/50) + 0.3\exp(-\mathbf{TE}/90)$, where $\mathbf{TE}$ is the echo time, which is a vector $8:8:256$ in milliseconds. To generate the data $\mathbf{d}$ for one noise realization, we set $SNR = 1000$. In the Table 1 below, we enumerate the estimator returned by individual optimization solver, and the corresponding value of loss function. To compare, we also generate the noiseless data $\mathbf{d}$, i.e.,

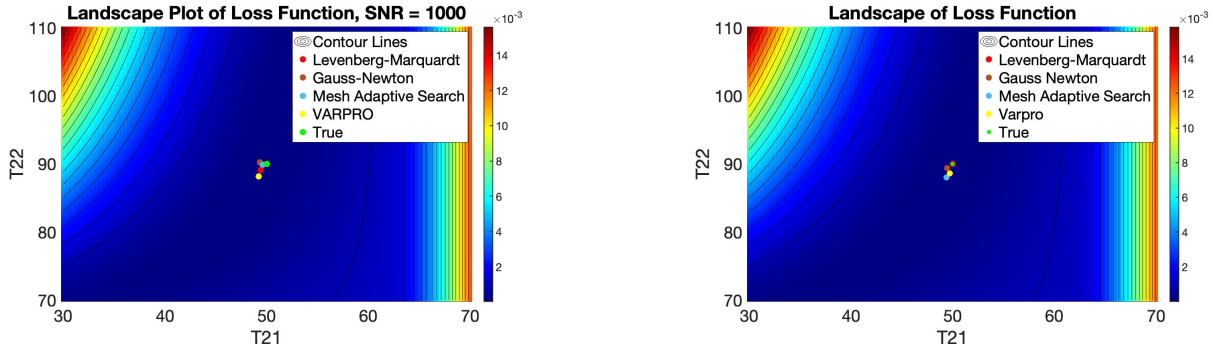
Table 1: Estimators with corresponding value of loss function, $SNR = 1000$

Solver	Estimator	Value of Loss Function
Gauss Newton	(0.6940,0.3063,49.3361,90.2190)	5.3124e-5
Levenberg-Marquardt	(0.6841, 0.3163, 49.4100, 89.1342)	3.6034e-05
MADS	(0.6934, 0.3068,49.6449, 89.8818)	3.6052e-5
VARPRO	(0.6932, 0.3171, 49.2124, 89.1562)	3.6169e-5
True	(0.7, 0.3, 50, 90)	3.7924e-5

$\mathbf{d} = 0.7\exp(-\mathbf{TE}/50) + 0.3\exp(-\mathbf{TE}/90)$ to compare the performance of individual solvers. In this noiseless case, we could offset the influence of the realization of a specific noise, and observe which solver could return a loss function value as close to zero as possible. From Figure 3b, we observe that the estimates returned by Levenberg-Marquardt coincide with the true parameters $T_{21} = 50, T_{22} = 90$.

Table 2: Estimators with corresponding value of loss function, noiseless data

Solver	Estimator	Value of Loss Function
Gauss Newton	(0.6908, 0.3092, 49.4642, 89.3857)	2.4257e-5
Levenberg-Marquardt	(0.7000, 0.3000, 50.0000, 90.0000)	9.4217e-25
MADS	(0.6925, 0.3050,49.4024, 89.0207)	1.1302e-7
Varpro	(0.6950, 0.3048, 49.7285, 88.6095)	2.1828e-07
True	(0.7, 0.3, 50, 90)	0



(a) Data with $SNR = 1000$ (b) Noiseless data

Figure 3: Landscape of loss function and estimators by various solvers.

To check if estimators are at least critical points, we do a sanity check to see if the Karush-Kuhn-Tucker conditions are met. To do so, we set up the following optimization problem

$$
\begin{aligned}
\underset{c_1,c_2,T_{21},T_{22}}{\operatorname{argmin}} \quad & \left\| c_1 \exp\left(-\mathbf{TE}/T_{21}\right) + c_2 \exp\left(-\mathbf{TE}/T_{22}\right) - \mathbf{d} \right\|_2^2 \\
\text{s.t.} \quad & -c_1 \leq 0 \\
& -c_2 \leq 0 \\
& c_1 - 1 \leq 0 \\
& c_2 - 1 \leq 0 \\
& -T_{21} \leq 0 \\
& -T_{22} \leq 0 \\
& T_{21} - T_{22} \leq 0 \\
& T_{21} - 200 \leq 0 \\
& T_{22} - 200 \leq 0
\end{aligned}
$$

We now define the corresponding Lagrangian function. First, since we do not have any equality constraints, then there are no equality terms in the Lagrangian function. Second, the complementary slackness property holds, so the inequality constraints are inactive and the optimum values are within the interior of feasible region. Hence, the Lagrangian function is the original loss function. This coincides with Figure 3, since the optimum values are indeed not achieved on the boundary of domain. Therefore, the Lagrangian function degenerates to the loss function, so it suffices to check the gradient of Lagrange function with respect to the parameters, and we list them below as

Table 3: Gradient of Lagrangian function evaluated at estimates ($SNR = 1000$)

Solver	Gradient of Lagrangian Function at Estimator
Gauss Newton	(-1.0692e-2, -1.0173e-2, -2.0430e-5, -1.2058e-5)
Levenberg-Marquardt	(2.4983e-4, 3.7184e-4, 3.9016e-6, 2.4022e-6)
MADS	(-2.2528e-4, -3.3072e-4, -3.1190e-6, -2.0495e-6)
VARPRO	(-3.0673e-4, -5.4640e-4,4.7554e-6, 4.1402e-6)
True	(0.0031, 0.0042, 4.5848e-5, 1.4530e-5)

Table 4: Gradient of Lagrangian function evaluated at estimates (Noiseless landscape)

Solver	Gradient of Lagrangian Function at Estimator
Gauss Newton	(-2.7845e-2, -1.0548e-2, -1.4572e-5, -1.1307e-5)
Levenberg-Marquardt	(0, 0, 0, 0)
MADS	(-1.6358e-4, -1.4972e-4, -2.6832e-6, -1.4820e-6)
VARPRO	(-2.8575e-4, -3.8534e-4,3.5712e-6, 3.8962e-6)
True	(0, 0, 0, 0)

Indeed, we find that the estimator returned by Levenberg-Marquardt has the minimum norm of gradient in the noiseless landscape, so according to the sanity check it is a local critical point, and from the landscape plot, a global minimizer.

4.2 Performance Regarding the Conditioning of the Model

The second criterion to compare the performance of these solvers is to study their stability of the estimators with respect to the conditioning of the model $\mathbf{G} = c_1 \exp(-\mathbf{TE}/T_{21}) + c_2 \exp(-\mathbf{TE}/T_{22})$. Then under this criterion, for a fixed conditioning of the model, the optimal solver should be able to return estimators with relatively small variance. In other words, both in the well-conditioned and ill-conditioned case of model $\mathbf{G}$, the optimal solver is supposed to show better stability.
First, we will study the conditioning of model $\mathbf{G}(\mathbf{p})$, where $\mathbf{p} = (c_1, c_2, T_{21}, T_{22})$.

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmin}} \left\| \mathbf{G}(\mathbf{p}) - \mathbf{d} \right\|_2^2 \tag{4.4}$$

where we have $\mathbf{G} : \mathbb{R}^N \to \mathbb{R}^M$. However, since $\mathbf{G}$ is a nonlinear model with respect to the parameters, we perform a first-order approximation of the $j-$th element of $\mathbf{G}(\mathbf{p})$ about a vector $\mathbf{p_0}$

$$\mathbf{G}_j(\mathbf{p}) \approx \mathbf{G}_j(\mathbf{p}_0) + \sum_l \left. \frac{\delta G_j}{\delta p_l} \right|_{\mathbf{p}_0} \cdot (p_l - p_{0,l}) \tag{4.5}$$

Therefore, we denote by $\mathbf{J}$ be the Jacobian matrix of $\mathbf{G}$ evaluated at $\mathbf{p_0}$, namely

$$J_{jl} = \left. \frac{\delta G_j}{\delta p_l} \right|_{\mathbf{p}_0}$$

Then, Eq.(4.4) becomes

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmin}} \left\| \mathbf{J}\mathbf{p} + \left(\mathbf{G}(\mathbf{p}_0) - \mathbf{J}\mathbf{p}_0 - \mathbf{d} \right) \right\|_2^2 \tag{4.6}$$

where $\mathbf{G}(\mathbf{p_0})$ and $\mathbf{J}\mathbf{p_0}$ are constants. Therefore, we could study the conditioning of the original model $\mathbf{G}$ by studying its Jacobian matrix $\mathbf{J}$. For a particular example, we take

$$\mathbf{G}(\mathbf{p}) = 0.7 \exp(-\mathbf{TE}/50) + 0.3 \exp(-\mathbf{TE}/T_{22}),$$

where we fix $c_1 = 0.7, c_2 = 0.3, T_{21} = 50$, and vary T_{22} within the range $[30, 70]$. We set the echo time $\mathbf{TE} = 8 : 8 : 256$ ms, and data $\mathbf{d}$ is obtained with $SNR = 1000$. Then, for this model, we obtain the Jacobian matrix $\mathbf{J}$ of size 32×4, and we plot its condition numbeer $\kappa(\mathbf{J})$ vs. T_{22}. As expected, from Figure 4, we observe that as T_{22} approaches T_{21}, the condition number of Jacobian matrix $\mathbf{J}$ becomes larger, so the problem is more ill-conditioned, and this is the region where our approach of introducing regularization will work better. On the other hand, as T_{22} deviates away from T_{21}, we see that we have a more stable model.

Then, we explore the stability of the problem with respect to the T_{22}/T_{21} ratio and the SNR. Now, keep all the parameters of the bi-exponential model before, except we let $T_{21} = 60$, and $T_{22} = 70 : 20 : 150$, so the T_{22}/T_{21} ratio is greater than 1. From the Figure 5 below, we also observe that the T_{22}/T_{21} ratio could be used to measure the stability of model $\mathbf{G}$. Then, we perform 100 noise realizations and thus obtain 100 estimators of c_1, c_2, T_{21}, T_{22}. Hence, we could plot the variance of the resulting estimators vs. T_{22}/T_{21} ratio. We observe that for all solvers, their stability is the worst when T_{21} equals T_{22}. Levenberg-Marquardt's and MADS's performance are close, with Levenberg-Marquardt's slightly better. VARPRO is the best since it guarantees great stability in a wide range of T_{22}/T_{21} ratio. In practice, this robustness is very important since we need the stability to deal with different set of T_{22} values.

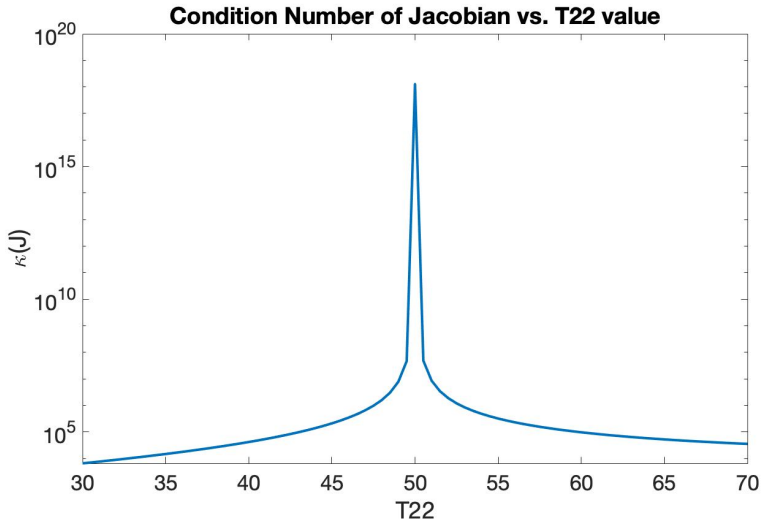
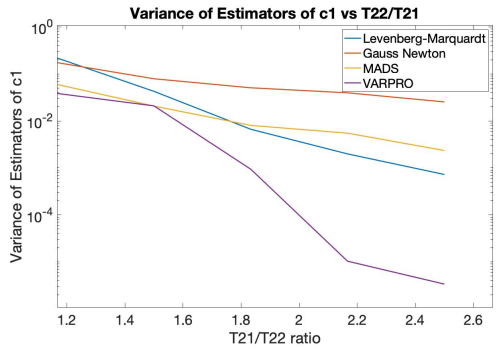
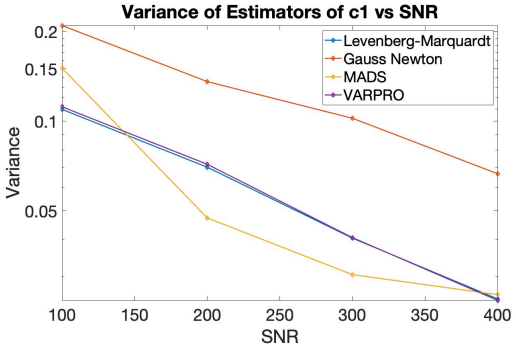


Figure 4: Condition number of the Jacobian matrix $\mathbf{J}$ of model $\mathbf{G}$ vs. T_{22}, where $\mathbf{G}(\mathbf{p}) = 0.7\exp(-\mathbf{TE}/50) + 0.3\exp(-\mathbf{TE}/T_{22})$. The condition number blows up when $T_{21} = T_{22} = 50$, as $\mathbf{J}$ becomes singular.



(a) Variance vs. T_{22}/T_{21} ratio



(b) Variance vs. SNR

Figure 5: Performance of solvers regarding to conditioning and SNR. Left: $c_1 = 0.7, c_2 = 0.3, T_{21} = 60, T_{22} = 70 : 20 : 150, SNR = 1000$. Right: $c_1 = 0.7, c_2 = 0.3, T_{21} = 50, T_{22} = 90, SNR = 100 : 100 : 400$.

4.3 Performance Regarding the Signal-to-Noise Ratio (SNR)

In practice, we will observe signal-to-noise ratio (SNR) in a wide range. SNR is defined as the ratio of the magnitude of signal to the magnitude of background noise, i.e.,

$$SNR = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

Therefore, the noise of data decreases as SNR increases. To compare the performance of solvers, we could study the variance of estimators by individual solvers, in a range of SNR. As expected, in Figure 5, variances of estimators decrease as SNR increases for all solvers.

4.4 Processing Time

Another important criterion to consider is the processing time. In Table 5 below, we compare the processing time for each solver for 100 noise realizations. The processing time is obtained by MATLAB command `tic toc`. From Table 5, we observe that Levenberg-Marquardt has the highest speed, while Gauss Newton and VARPRO also relatively fast.

Table 5: Processing time for different solvers

Solvers	Processing Time (sec)
Gauss Newton	5.6103
Levenberg-Marquardt	1.4235
MADS	29.0450
VARPRO	4.4697

4.5 Conclusion

According to the results in each criterion in this section, we conclude MADS, Levenberg-Marquardt and VARPRO have better performance than Gauss Newton. In our problem, the landscape of loss function is well-behaved and smooth, so all solvers could achieve a global minimum reasonably well. We choose Levenberg-Marquardt method as the one which we use for finding the desired range of Tikhonov regularization parameter λ, since it returns a global optimum of the loss function, relatively stable on different conditioning and SNR of the problem, and relatively fast processing speed. In the case of multiple existences of local optimum and a more complex geometry of the landscape of loss function, there are multi-scale optimization algorithms available in the literature, that could jump out of the local trap and find the global optimum in the feasible region.

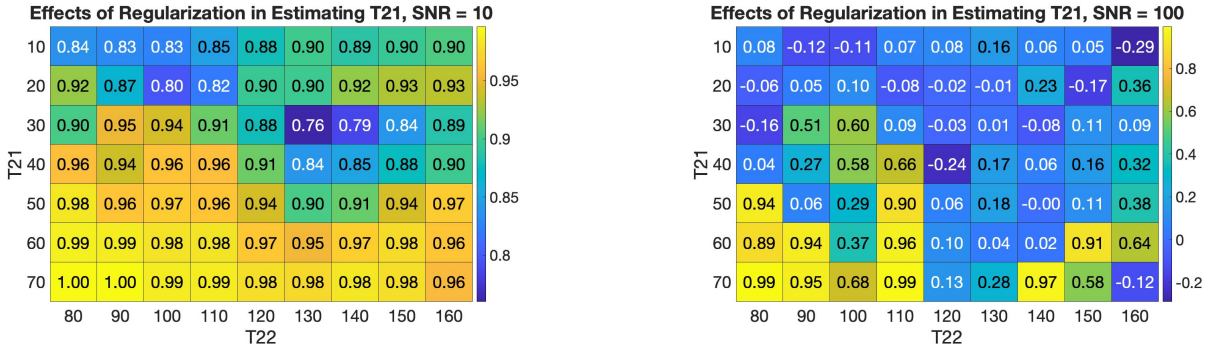
5 Results

5.1 Choosing Optimal λ I: For each Combination of Parameters

In practice, given a prior knowledge of range of parameters, one needs to figure out the value of regularization parameter λ that possesses the power to improve the average estimation accuracy. Since we choose the MSE as the metrics to measure the estimators, the goal is to find λ that gives the average MSE less than CRLB. In other words, instead of trying to find the optimum λ for one possible combination of parameters, we need to find a λ that gives the best improvement of MSE over CRLB, for all possible combinations of parameters in a particular range. A preliminary goal is that, to visualize the effects of regularization parameter λ on each possible combination of T_{21}, T_{22}. This means that, for each combination of T_{21}, T_{22}, we calculate the regularized MSE of estimators for T_{21}, T_{22}, and find the greatest improvement of MSE over CRLB,

$$\text{Improvement} = \frac{CRLB - MSE}{CRLB} \tag{5.1}$$

One case that interests an experimentalist is assuming we are given c_1 and c_2, and we look for the optimal λ that works for T_{21} and T_{22} in some ranges, which we will focus on. Particularly, in our experiment, we set $c_1 = 0.5, c_2 = 0.5$, T_{21} in the range of $10:10:70$, T_{22} in the range of $70:10:160$. Although in theory, T_{22} could achieve 1500, but the ranges we choose here are broad enough for practical purposes. Then, for each combination of T_{21}, T_{22}, we calculate the regularized MSE of estimators for T_{21}, T_{22}, and then for each λ, we calculate the corresponding MSE, and select the λ that gives the best improvement. Therefore, in this case, the λ that corresponds to each combination of parameters is optimal and thus different. In Figure 6, each entry in the table is the improvement we defined in (5.1). Therefore, positive, larger values indicate significant improvment of MSE over CRLB, while negative number means there is no improvement, i.e., the MSE is alway above the CRLB.



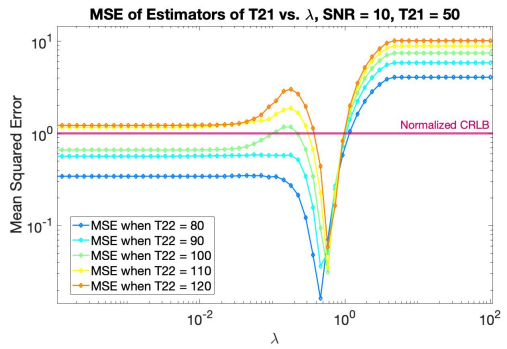
(a) $SNR = 10$ (b) $SNR = 100$

Figure 6: Improvements of MSE over CRLB by optimal λ. Positive numbers indicate there is improvement of MSE over CRLB, and negative numbers indicate there is no improvement. An optimal value of λ was used for each entry in the figure.

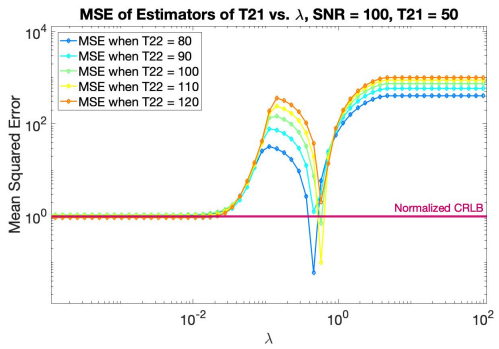
Firstly, in Figure 6, we observe that for $SNR = 10$, we see the effects of regularization are more significant than that of $SNR = 100$, where a large portion of the map shows a huge improvement of MSE over CRLB. This is due to the fact that in the high noise case, by introducing regularization parameter λ, MSE can break the theoretical limit of CRLB, since we artificially introduce more bias, and our CRLB only works for unbiased estimators. Secondly, we observe for both cases of $SNR = 10$ and $SNR = 100$, the lower left corner of the table shows the effects of regularization more obviously, and mitigates when we move to upper right of the table. This result coincides very well to our conditioning analysis to the bi-exponential model, since when T_{22}/T_{21} ratio is close to 1, we have a more ill-conditioned system, and that is the case when introduction of regularization is more helpful.

Next, we plot regularized MSE vs. λ for different combinations of T_{21} and T_{22}, for different SNRs. In Figure 7 , we see when we fix $T_{21} = 50$, and vary T_{22} in the range of $80:10:120$, we observe a relatively consistent range of λ's that make MSE less than CRLB. To be more specific, we choose 60 λ's uniformly in the $\log\text{space}(-4, 2)$, and for each T_{21}, T_{22} combinations, we compute the corresponding regularized MSE of estimators of T_{21}, the corresponding CRLB, and the relative improvement of MSE over CRLB, with that particular λ, which we show in the figure below. The connection between Figure 6 and Figure 7 is that, the values in tables of Figure 6 measure how much MSE dips below the CRLB in Figure 7, and that's why a larger and positive value in Figure 6 means more significant improvement, while negative values would mean MSE always sits above CRLB.

Then, to understand why we have MSE dipping below CRLB, we implement a similar visualization of



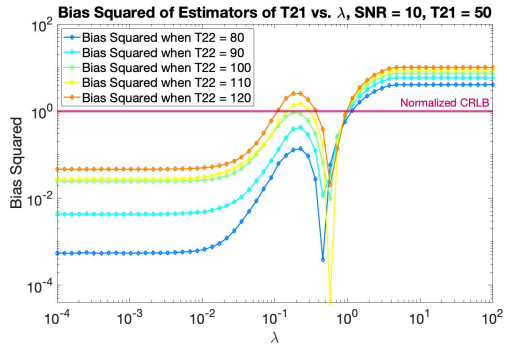
(a) $SNR = 10$



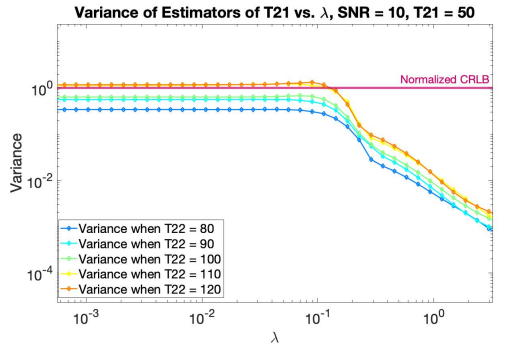
(b) $SNR = 100$

Figure 7: MSE vs. λ

bias squared vs. λ and variance vs. λ. For the MSE in Figure 7a, we decompose it into the bias squared and variance of estimators of T_{21}, and obtain Figure 8. From Figure 8, we conclude that the MSE is mostly contributed by bias we artificially introduced through regularization parameter λ, where the MSE dipping below CRLB is caused by the bias squared dipping below CRLB, and we see that variance of estimators of T_{21} is decreasing as regularization parameter λ increases, which is what we expect since regularization should improve the stability of problem.



(a) Bias squared of estimators of T_{21} vs. λ.



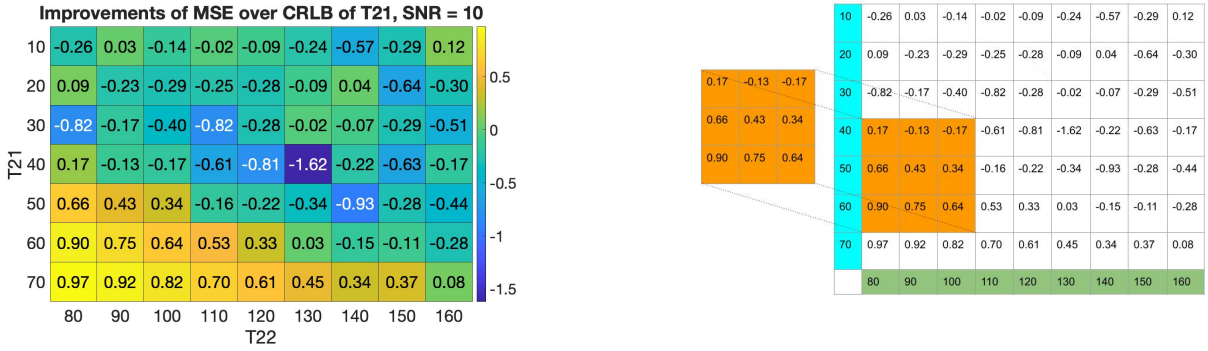
(b) Variance of estimators of T_{21} vs. λ.

Figure 8: Bias squared, variance of estimators of T_{21} vs. λ, $SNR = 10$.

5.2 Choosing Optimal λ II: A Strategy When Given a Prior Range of Parameters

In practice, one will be given a range of possible values of T_{21} and T_{22}, and need to give an optimal λ that works well for all parameters on average. Based on the result that even different combinations of parameters shall have a relatively consistent range of optimal value of λ's, we are now finally proposing a strategy of how to give the optimal λ for a prior range of $T_{21} - T_{22}$ parameters. Given a prior range of $T_{21} - T_{22}$, we could compute the average of relative improvement of MSE over CRLB, for all λ's in the range, and find the λ that gives the highest possible average improvement we defined in (5.1). In another word, this heat map now corresponds to a single λ, and we obtain as many heat maps as many λ's we consider. Then, we repeat the above procedure for all λ's in some particular ranges, then we could average the improvement of MSE

over CRLB, and then chooses the λ that gives the best average improvement. In Figure 9, all entries denote the improvement of MSE over CRLB we defined in (5.1), and corresponds to the same $\lambda = 1e-4$.



(a) Improvements of MSE with $\lambda = 1e-4$.

(b) An example of prior range of $T_{21} - T_{22}$.

Figure 9: Improvements of MSE over CRLB by λ. Positive numbers indicate there is improvement of MSE over CRLB, and negative numbers indicate there is no improvement. A single $\lambda = 1e-4$ over the entire diagram has been used, which is the main difference with Figure 6. Left: Improvements of MSE over CRLB with $\lambda = 1e-4$. Right: A particular example of a prior range of $T_{21} - T_{22}$, with $\lambda = 1e-4$.

In Figure 9b, we take the prior range of T_{21} in the range of $[40, 60]$, T_{22} in the range of $[80, 100]$ as labeled in the orange area. Then, for this particular heat map with $\lambda = 1e-4$, we compute the average improvement in this prior range, which is 0.3988. Then, we repeat this calculation for all the corresponding region in heat maps, with respect to different λ's. Finally, we choose the λ that gives us the best average improvement of MSE over CRLB within the specific region of parameters. In this particular example, we find the optimal λ that works best on average is 0.7318, with a average improvement of MSE over CRLB to be 0.5382.

5.3 Practical Applications

To illustrate the practical application of these results, we envision the following scenario: A magnetic resonance experimentalist will be making relaxometry measurements on a specific sample. Based on the construction of the sample, it is comprised of two components of equal amount. The signal model will be bi-exponential, with equal values for $c_1 = c_2 = 0.5$.

Given the size and chemical construction of the sample, and its chemical stability and the amount of time available for study, the SNR will be around 30. Based on the hydration and chemical make-up of the sample, a plausible range of T_{21} is $20 - 35$ ms, while a plausible range of T_{22} is $75 - 160$ ms. The experimentalist has no prior expectation of the likelihood, within these ranges, of the true values of T_{21} and T_{22}; In other words, the priors are flat over the given ranges. (Note that there are other possibilities, e.g., T_{21} may be much more likely to have a value in the center of the given range than it is have a value towards the edges of the range, but this can be incorporated by performing a weighted average over prior expected values.)

Based on the given values of c_1, c_2, T_{21}, T_{22}, and SNR, we will first pick a λ using L-curve method, generalized cross validation, or the discrepancy principle, which are standard approaches and can be found in Spencer, et al. (2020)[9]. This will get us a reasonable value of λ to start with.

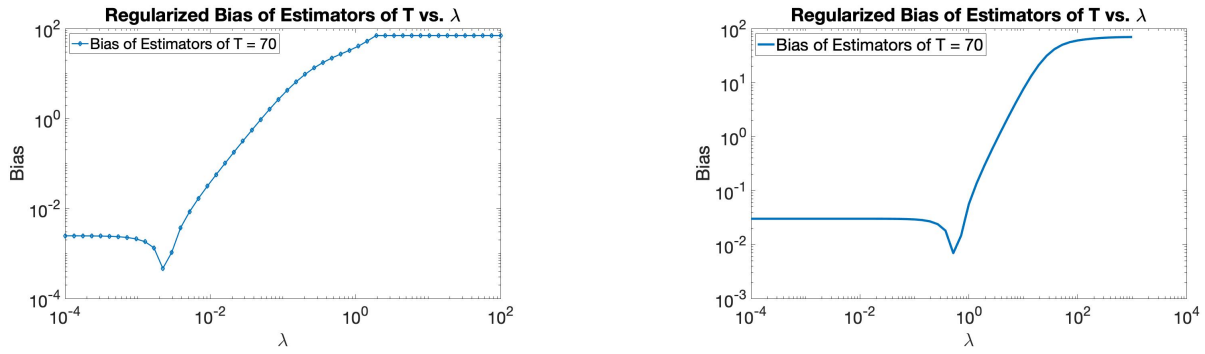
Next, we fix c_1, c_2, and for the given SNR, we calculate the MSE for every pair of T_{21}, T_{22} over many

noise realizations. We will also calculate the corresponding CRLB. Then, we can calcuate the ratios of MSE/CRLB for every pair of T_{21}, T_{22}.

After this, we will average MSE/CRLB within the plausible ranges of T_{21} and T_{22}. Now, we repeat this for a range of λ's, which will be guided by the general shape of plots shown in this report. We find some values of λ giving MSE/CRLB less than 1 and we select the optimal λ which achieves the greatest improvement.

6 Validation

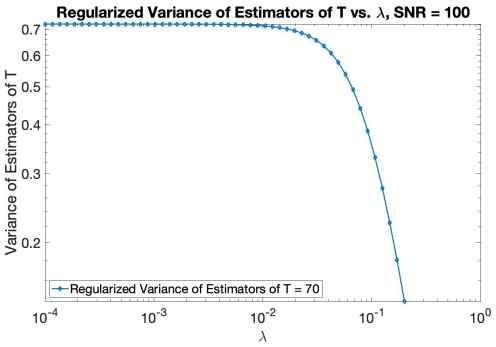
To test our results, we use mono-exponential model as an example to illustrate the similarity and differences between numerical and theoretical analysis. By the work by Box, et al. (1971)[2], we are able to explicitly approximate the bias of the estimators of the nonlinear parameter estimation model. Also, by the work by Spencer, et al. (2020)[9], we could compute the covariance matrix of the estimators, then obtain the variance. Therefore, we are able to calculate the MSE directly by the relationship among them. In Figure 10 and Figure 11 we compare the results obtained numerically and theoretically. Regarding the bias of estimators, we observe from Figure 10a and Figure 10b that the general shape of both results coincide very well, while the theoretical result deviates from the numerical result by a scaling factor that depends on the particular parameters, and this is because that the work by Box, et al. (1971)[2] is only a local first order approximation of the bias in the nonlinear parameter estimation, so there will be differences to numerical solutions, and in real application, we should follow the results obtained numerically. On the other hand, regarding variance of estimators, from Figure 11, the variance of estimators obtained numerically and theoretically coincide very well.



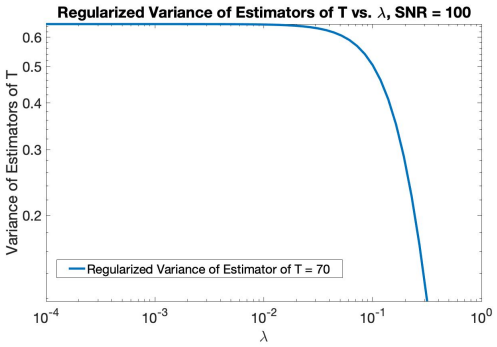
(a) Numerical Results

(b) Theoretical Calculation

Figure 10: Bias of estimators of T vs. λ, where $c = 0.5$, $T = 70$, and $\mathbf{TE} = 8 : 8 : 128$ ms, $SNR = 100$. Left: Numerical results using Levenberg-Marquardt. Right: Theoretical calculation using results from Box, et al. (1971)[2].



(a) Numerical Results



(b) Theoretical Calculation

Figure 11: Variance of estimators of T vs. λ, where $c = 0.5$, $T = 70$, and $\mathbf{TE} = 8 : 8 : 128$ ms. Left: Numerical results using Levenberg-Marquardt. Right: Theoretical calculation using results from Spencer, et al. (2020)[9].

7 Project Schedule and Milestones

- September - October 2020: Define the project. Set up goals for the project. (finished)
- November 2020: Write and implement codes for solving the bi-exponential model data, including Levenberg-Marquardt, VARPRO, MADS, etc. (finished)
- December 2020: Validation and interpretation of results. (finished)
- February 2021: Write and implement codes for solving the mono-exponential model data, and compare results to those of bi-exponential model. (finished)
- March - May 2021: Propose strategy to select optimal regularization parameter, visualization of results, wrap up the project. (finished)

8 Future Work

The methods and strategy we proposed in this report could be easily generalized to other common biomedical MRI models. Originally, we planned to study some of these models within the scope of this project, but later we decided it is more important to study bi-exponential model thoroughly and completely, and provide deeper insights into the project. Having said that, we still list them here as a reference for possible future work, and each model itself could be expanded to a independent project.

- Stretched-exponential model[10]

$$S(t;\alpha,D) = S_0 \exp\left(-(tD)^{\alpha}\right)$$

  where α is the stretching constant ($0 \leq \alpha \leq 1$), and D denotes the distributed diffusion parameter. $S(t;\alpha,D)$ and S_0 denote signal intensities with and without diffusion weighting, respectively. t represents the measurement times at which we measure the signal intensities. This model arises from some models of restricted diffusion.

- Diffusion kurtosis model[10]

$$S(t; D, K) = S_0 \exp\left(-tD + t^2 D^2 K/6\right)$$

where D represents mean diffusivity, and K represents mean kurtosis. $S(t; \alpha, D)$ and S_0 denote signal intensities with and without diffusion weighting, respectively. t denotes the measurement times at which we measure the signal intensities. This model is a first-order approximation to non-Gaussian diffusion.

9 Implementation and Deliverables

9.1 Hardware and Software

We implement the algorithm on the personal laptop with

- CPU: 2.3GHz Dual-Core Intel Core i5
- Memory: 8GB 2133 MHz LPDDR3
- OS: OS Catalina Version 10.15.6

The software we use is MATLAB R2020b.

9.2 Deliverables

- Datasets: All mono-exponential and bi-exponential model signals are synthetic data, generated by the model and additive white Gaussian noise.
- MATLAB codes: The folder "ns_biexp" contains the driver codes for visualization of final results on strategy of choosing optimal regularization parameter. The folder "biexp" and "monoexp" contain codes of optimization solvers, validation and testing results for bi-exponential and mono-exponential model. The link to MATLAB codes is referenced here `https://github.com/zsong2019/AMSC664`.
- Trained estimators of models: We have many saved data of estimators of bi-exponential model, from which we perform statistical analysis, including bias, variance, MSE, etc.
- Documents including the proposal, the presentation slides, the mid-year report and final report, etc.

10 Conclusion

In the final report, we analyze the mono-exponential model and bi-exponential model, and implement different solvers to fit for the optimal model parameters. We also improve the stability of estimators by introducing an regularization term to the nonlinear least squares loss function, which reduces the MSE over the CRLB of parameters. Most importantly, we answer the question that, instead of having the prior information about the accurate values of the parameters of the model, we are given a prior range of the possible values

of parameters, and we find out a method to give an optimal regularization parameter that on average, an experimentalist will get a better improvement of MSE over CRLB, which is of great practical value.

A possible future direction of the work is to assume we have information about T_{21}, T_{22} values, and estimate for the component fractions c_1, c_2, or we estimate the four parameters simultaneously. Our idea of introduction of regularization could then be applied to other common biomedical MRI models.

11 Bibliography

References

1. Mustapha Bouhrara and Richard G Spencer. Fisher information and cramér-rao lower bound for experimental design in parallel imaging. *Magnetic resonance in medicine*, 79(6):3249–3255, 2018.

2. MJ Box. Bias in nonlinear estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(2):171–190, 1971.

3. Yonina C Eldar. Uniformly improving the cramér-rao bound and maximum-likelihood estimation. *IEEE Transactions on Signal Processing*, 54(8):2943–2956, 2006.

4. Ata Kabán. Non-parametric detection of meaningless distances in high dimensional data. *Statistics and Computing*, 22(2):375–385, 2012.

5. Zhi-Pei Liang and Paul C Lauterbur. *Principles of magnetic resonance imaging: a signal processing perspective*. SPIE Optical Engineering Press, 2000.

6. Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

7. Dianne P O'leary and Bert W Rust. Variable projection for nonlinear least squares problems. *Computational Optimization and Applications*, 54(3):579–593, 2013.

8. Calyampudi Radhakrishna Rao, Calyampudi Radhakrishna Rao, Mathematischer Statistiker, Calyampudi Radhakrishna Rao, and Calyampudi Radhakrishna Rao. *Linear statistical inference and its applications*, volume 2. Wiley New York, 1973.

9. Richard G Spencer and Chuan Bi. A tutorial introduction to inverse problems in magnetic resonance. *NMR in Biomedicine*, page e4315.

10. Jianjian Zhang, Shiteng Suo, Guiqin Liu, Shan Zhang, Zizhou Zhao, Jianrong Xu, and Guangyu Wu. Comparison of monoexponential, biexponential, stretched-exponential, and kurtosis models of diffusion-weighted imaging in differentiation of renal solid masses. *Korean journal of radiology*, 20(5):791–800, 2019.