Piecewise linear Model Fitting of DCE-MRI data

(13 May 2016 - 20 July 2016)

By

Anmol Nijhawan

Department of Electrical and Electronics Engineering

Indian Institute of Technology Guwahati.

Under the guidance of

Dr. Anup Singh

Assistant Professor

Indian Institute of Technology Delhi.

(Centre for Biomedical Engineering.)

Funding agency

National Network for Mathematical and Computational Biology (NNMCB)

(The Science and Engineering Research Board,

Government of India).

Indian Institute of Science, Bangalore 560012

Contents

Piecewise linear Model Fitting of DCE-MRI data

1. Purpose:

DCE-MRI is widely used technique for diagnosis of disease. Various models are available for quantitative analysis of DCE-MRI data. PL model is one such model for DCE-MRI data analysis. Purpose of this project was to implement piecewise linear model for DCE-MRI data analysis as a Mex File using C and compare results with MATLAB based implementation in term of accuracy and speed. For this purpose LM and trust regions based algorithms are implemented and evaluated in term of accuracy and performance using simulated as well as real human brain DCE-MRI data.

2. Introduction:

Magnetic Resonance Imaging (MRI) is a powerful and non-invasive medical imaging technique used for diagnosis of diseases and for studying anatomy and the physiological processes of the body.

2.1 ***How MRI works:-***

Most MRI systems use a superconducting magnet, which consists of many coils or windings of wire through which a current is passed, creating a magnetic field of upto 7.0 tesla. Maintaining such large magnetic field is accomplished using superconductivity which reduces resistance to almost zero. Superconductivity is achieved by using liquid helium at 452.4 degrees below zero.

There are also three gradient

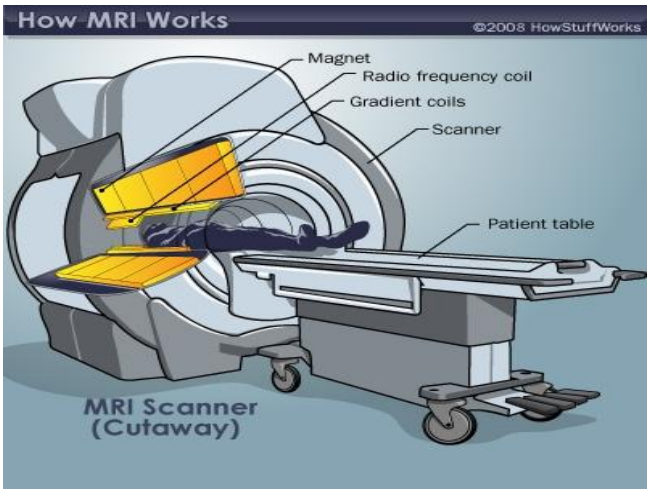
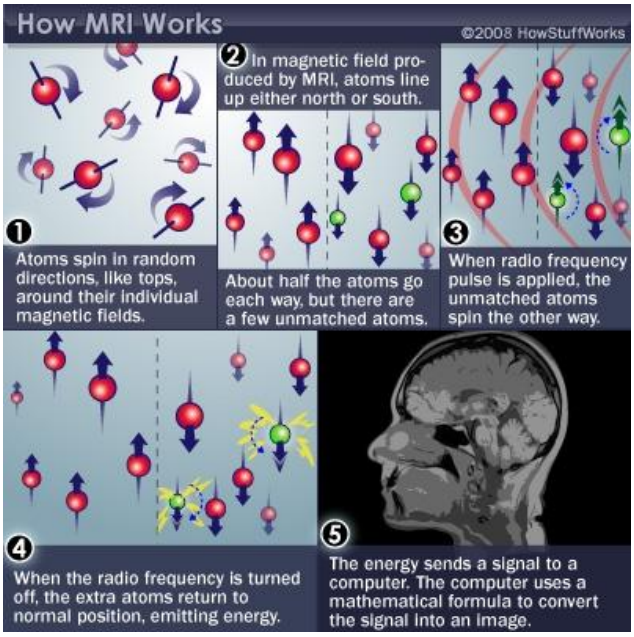


Figure 1: Cartoon of MRI.

Ref: ***http://science.howstuffworks.com/mri1.htm***

magnets which are much lower in magnetic strength compared to main magnet. While main magnet create an intense, stable magnetic field around the patient, the gradient magnets create a variable field, which allows different parts of the body to be scanned.

Radiofrequency coils transmit radiofrequency waves into patient's body. There are different coils for different body parts: knees, shoulders, wrists, heads, necks and so on. These coils usually conform to the contour of the body part being imaged, or atleast reside very close to it during the exam. Other parts of the machine include a very powerful computer system and a patient table, which slides the patient into the bore.



Brief explanation on the principle and working of MRI.

Ref: *http://science.howstuffworks.com/mri3.htm*

The MRI machine applies a ***radio frequency (RF) pulse*** that is specific only to particular nuclei i.e. hydrogen.

Basic Principle of MRI: The MRI machine applies a ***radio frequency (RF) pulse*** that is specific only to particular nuclei i.e. hydrogen.

The system directs the pulse toward the area of the body we want to examine. When the pulse is applied, the unmatched protons absorb the energy and spin in a different direction, this is the resonance part of MRI. The RF pulse forces them to spin at a particular frequency, in a particular direction. This specific frequency is called ***Larmour frequency.*** This frequency is calculated based on strength of main magnetic field and particular tissue being imaged. The gradient magnets turn on and off in a specific manner, altering the magnetic field on a local level and capturing the image of a particular region referred as "slice". When the RF pulse is turned off, the hydrogen protons slowly return to their natural alignment within the magnetic field and release the energy absorbed from RF pulses. When they do this, they give off a signal that the

coils pick up and send to the computer system. Computer system computes the ***Fourier transform*** of the captured signal so that it can be used for further processing and seeing the images of the region of interest.

2.2 ***Dynamic Contrast Enhanced (DCE) MRI: -*** **[1]**

Dynamic Contrast Enhanced (DCE) MRI is an advanced MRI technique. With recent advances the use of ***DCE MRI*** Imaging has become quite popular. In this novel technique an intravenously injected tracer (normally Gadolinium agent, used as contrast) increases the contrast between different tissues or between normal and abnormal tissue in region being imaged. Dynamic refers to serial acquisition of images of tissue or section under consideration before and after intravenously injection of tracer. This results in a time series data at each voxel which help in extracting useful information about tissue vasculature in region of interest and play dominant role in diagnosis, staging and follow up studies. In DCE-MRI, a paramagnetic substance is used as tracer. In DCE-MRI a linear relation has been observed to exist between concentration and change in relaxivity (inverse of relaxation time). Therefore, because of the dependence on the tissue parameters, the measured signal intensity in DCE-MRI data is spatially non-linearly related to contrast agent concentration. Contrast agents are classified into two types:

- Positive: Cause a reduction mainly in T_1 relaxation time. They appear as bright regions in MRI and are typically small molecular weight compounds such as Gadolinium. Dynamic MRI study using these contrast agents is called T_1 weighted DCE-MRI. **[2]**
- Negative: They are small particulate aggregates often termed super-paramagnetic iron oxide. These agents produce predominantly spin-spin relaxation effects (local field Inhomogeneities), which result in shorter T_2 or T_2^* relaxation times. They appear predominantly dark on MRI. Dynamic MRI based upon these contrast agents is called dynamic susceptibility contrast (DSC) MRI.

  Both DSC-MRI and DCE-MRI techniques measure the influence of the same tracer through its effect on relaxation times. Basically, in DSC the susceptibility effect or bulk magnetic susceptibility (BMS) dominates relaxivity effect while in DCE-MRI the T1 relaxivity effect dominates susceptibility effect.

2.3 ***Contrast insertion and DCE-MRI Data Acquisition***-

Tracer is injected intravenously and it travels to arteries via heart-lung system and distributes uniformly throughout the body. First circulation of tracer bolus is followed by its re-circulation till its complete removal from the body by the kidney action. In this study, the region of interest is brain. In this MRI technique, a tracer (like Gd-DTPA) is injected to the patient and multi-slice image sequences over time are acquired providing a time series of 4D DCE-MRI data. The data consists of a 2D image frame in the XY plane for each slice in the Z plane. This type of slice set is provided for each time frame (time points) in series (t). Each image frame of the data provides a cross- sectional view of the patient's brain.

2.4 ***Applications in medicine*** – DCE-MRI technique is used in the diagnosis of:

- Tumors/cancer
- Musculoskeletal disorders
- Strokes
- Alzheimer
- Cerebral Microvascular disease

2.5 **Data fitting for quantitative analysis of DCE-MRI data:**

Quantitative analysis of DCE-MRI data involves fitting using various mathematical models. Various algorithms for data fitting are available. Inbuilt routine of MATLAB can be used for data fitting; however, it is quite slow process. Alternative approaches are implementation in C/JAVA. Recently, Mex files are used to link C based codes with MATLAB. Mex files allow to retain speed of C codes and utilization of various features of MATLAB. Mex stands for MATLAB Executable. Mex Files are a way to call custom C or FORTRAN routines directly from MATLAB as if they were MATLAB built-in functions. MATLAB's Mex interface allows to speed up bottlenecks in a code (usually loop intensive computations) by writing them in low-level languages.

3. Material and Method:

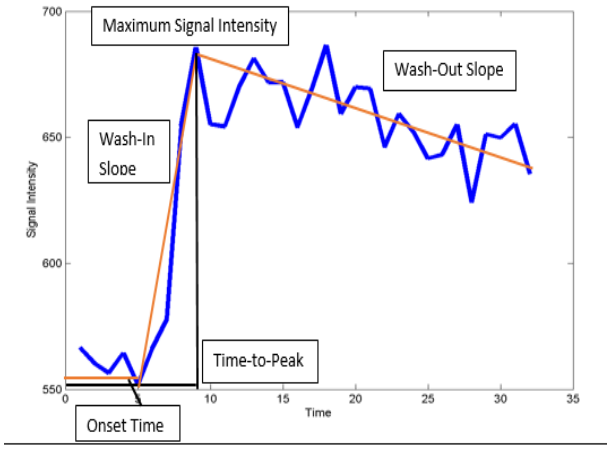
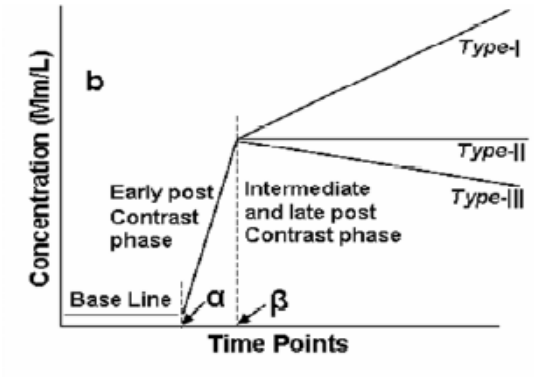
MRI data was acquired for brain tumor patients at 3T MRI scanner. Imaging started with a localizer followed by conventional T1, T2 and FLAIR weighted images. After this DCE-MRI data was acquired.

Imaging parameters: T2-weighted (TR/TE/ =3500ms/110ms, , slice thickness=4mm, acquisition matrix= 384×384, field of view (FOV) =230×230mm^2, reconstructed matrix=448×448), T1-weighted (TR/TE=2000ms/20ms, number of slices=28, slice thickness=5mm, FOV=230×184mm^2, acquisition matrix=400×272, reconstructed matrix=400×400, , inversion time (IR) =800ms), FLAIR (TR/ TE/ IR =4700ms/288ms/1650ms, NSA=2, slice thickness=0.56mm over contiguous, flip angle=90˚, acquisition matrix= 204×204, FOV=230×184mm^2, reconstructed matrix=256×256),

DCE-MRI was performed using the sequences described below. First, pre-contrast 2D T1-weighted, TSE (TR/TE 360/10 ms), and fast dual spin echo proton density (PD)-weighted and T2-weighted (TR/TE1/TE2 3500/23.2/90 ms) with slice thickness 6 mm; FOV= 240 × 240 mm^2; matrix size=256×256, fat suppressed images were acquired to quantify voxel-wise pre contrast tissue longitudinal relaxation time. Then, a DCE perfusion imaging dynamic series was performed using a T1 fast field echo (T1-FFE) sequence (TR /TE= 5.0/1.4 msec; flip angle 10; slice thickness 6 mm; FOV= 240 × 240 mm^2; matrix size=128×128; image size= 256×256, half scan option enabled with factor Y=0.625). At the fourth time point of the DCE-MRI data acquisition, 0.1 mmol/kg body weight of Gd-BOPTA (Multihance, Bracco, Italy) was administered intravenously with the help of a power injector at a rate of 3.5 ml/sec, followed by a bolus injection of a 30-ml saline flush (2). A series of 384 images at 32 dynamics for 12 slices were acquired with a temporal resolution of 3.9 sec.

3.1 **Approximation of DCE-MRI DATA Trend by a Continuous Piecewise Linear Function (PL Model) [3.]**

The signal intensity time curve obtained from Dynamic contrast enhanced (DCE) MRI data at individual voxels is converted into Concentration time curve (CTC). The different shapes of

	
Signal Intensity Time Curve.	Concentration Time Curve.

Concentration time curve forms an important criterion for distinguishing enhancing lesions (tumor) from normal tissue. Close observations suggest a Piecewise Linear Function to approximate the Concentration time curve. The piecewise linear function which models the CTC curve is as follows –

$$f(t) = \begin{cases} c; & t \leq \alpha \\ c + b_1(t-\alpha); & \alpha < t \leq \beta \\ c + b_1(\beta-\alpha) + b_2(t-\beta); & t > \beta \end{cases}$$

- Type-I curve models the enhancing lesions. The increase in concentration values continuously is due to leakage of contrast due to blood brain barrier breakdown.
- Type-II curve model the tissue part in brain. In this part the washout is slower.
- Type-III curve model the arteries in brain. In this part the washout is faster.

The five parameters which form the basis of the Piece-wise Linear model are-

- α- BAT (Bolus arrival Time). The time when contrast (gadolinium-based) reaches the brain.
- β- Time to peak. The time after BAT when concentration value is maximum in case of arteries and tissues.
- c-constant which accounts for initial noise.

- b_1-Slope 1
- b_2-Slope 2

3.2 **Data Fitting Methods:**

Three different algorithms were used for model fitting on CTC curves at individual voxels. They are-

- **<u>Ordinary-Least-Squares method</u> [3.]**

Goal is to minimize the differences between the observed responses in the dataset and the responses predicted by the linear approximation of the data.

$$R^2 = \sum_{t=1}^{N}(y(t) - f(t, \alpha, \beta, c, b_1, b_2))^2$$

The condition for the minimum is as follows:

$$\frac{\partial R^2}{\partial c} = 0;\ \frac{\partial R^2}{\partial b_1} = 0;\ \frac{\partial R^2}{\partial b_2} = 0;$$

α And β are found iteratively by moving through different time points and minimizing the error for each time point by finding the minimum of each parameters (c, b_1, b_2). The normal equations for finding the minimum of free parameters (c, b_1, b_2) are-

$$\boldsymbol{AX=B};$$

$$\text{A}=\begin{bmatrix} N & S_1 + (\beta-\alpha)(N-\beta) & S_2 \\ S_1 + (\beta-\alpha)(N-\beta) & S_{11} + (\beta-\alpha)^2(N-\beta) & S_2(\beta-\alpha) \\ S_2 & S_2(\beta-\alpha) & S_{22} \end{bmatrix};$$

$$\text{X}=\begin{bmatrix} c \\ b_1 \\ b_2 \end{bmatrix};\qquad \text{B}=\begin{bmatrix} S_{y1} \\ S_{y2} \\ S_{y3} \end{bmatrix};$$

$$S_1 = \sum_{\alpha+1}^{\beta}(t-\alpha);\ S_{11} = \sum_{\alpha+1}^{\beta}(t-\alpha)^2;\ S_2 = \sum_{\beta+1}^{N}(t-\beta);\ S_{22} = \sum_{\beta+1}^{N}(t-\beta)^2;$$

$$S_{y1} = \sum_{1}^{N} (y(t))\,;\; S_{y2} = \sum_{\alpha+1}^{\beta} y(t)(t-\alpha) + \sum_{\beta+1}^{N} y(t)(\beta-\alpha)\,;$$

$$S_{y3} == \sum_{\beta+1}^{N} y(t)(t-\beta)\,;$$

The system $\boldsymbol{AX=B}$ is solved by taking the inverse of matrix ($\boldsymbol{A^{-1}}$) and multiplying to the equation. Thus $\boldsymbol{X= A^{-1}B}$

The system of equations was solved for each choice of parameters α and β and the choice of parameters which resulted in a minimum R^2 was taken as a set of best fit parameters.

- **<u>Levenberg Marquardt algorithm (LM Algorithm)</u> [4.]**

This algorithm is based on non-linear least square analysis to fit a set of M unknown parameters over N observations (data points). For nonlinear dependencies the minimization proceeds iteratively, starting from an initial guess and updating the unknown parameters when the sum of squared error between the model function and observed values (data points) decreases from its previous value. The algorithm interpolates between the ***Gauss-Newton*** method and ***Gradient Descent*** method depending upon the current parameters. When the current parameters are far away from a local minimum then gradient descent method is applied to get closer to the solution and when current parameters is closer to solution then Gauss-Newton is applied.

Let f be an assumed functional relation which maps a parameter vector
p ε R^m to an estimated measurement vector $\hat{x}$ = f(p) , $\hat{x}$ ε R^n. An initial parameter estimate p_0 and a measured vector x are provided & it is desired to find vector p^+that best satisfies the functional relation f, i.e. minimizes the squared distance $\varepsilon^T\varepsilon$ with ε= x-$\hat{x}$. J is the Jacobin matrix.

$$J^T(J\delta p - \varepsilon) = 0 \qquad (1.1)$$

$$J^T J\delta p = J^T\varepsilon \qquad (2.1)$$

$$N\delta p = J^T\varepsilon \qquad (3.1)$$

Where off- diagonal elements of N are identical to the corresponding elements of J^TJ and diagonal elements are given by $N_{ii} = \lambda + [J^TJ]_{ii}$

For some λ>0. λ is called damping term.

The algorithm can be stated as follows-

1. Compute $\hat{x} = f(p_0)$ and calculate ε= x-$\hat{x}$.
2. Pick a modest value for lambda(λ) say λ=0.001
3. Solve the equation (3.1) for δp and evaluate $\widehat{x1} = f(p + \delta p)$ and calculate $\varepsilon 1$= x-$\widehat{x1}$
4. If $\varepsilon 1(p + \delta p) \geq \varepsilon(p)$, increase λ by a factor of 10 and go back to step 3.
5. $\varepsilon 1(p + \delta p) < \varepsilon(p)$, decrease λ by a factor of 10, update the trial solution p ← p + δp , and go back to step 3.

The process of repeatedly solving Equation 3 for different values of the damping term until an acceptable update to the parameter vector is found corresponds to one iteration of LM algorithm. The LM algorithm terminates when atleast one of the following conditions is satisfied:

- The magnitude of the gradient of $\varepsilon^T \varepsilon$ i.e. $J^T \varepsilon$ in the right hand side of Equation 2 drops below a threshold ε_1
- The relative change in the magnitude of δp drops below a threshold ε_2.
- The error $\varepsilon^T \varepsilon$ drops below a threshold ε_3.
- A maximum number of iterations K_{max} is completed.

In order to compute δp from ***equation (3)*** matrix inversion and solution of linear algebraic equations methods such as Gauss Jordan Elimination with back-substitution or Cholesky decomposition can be applied. Cholesky decomposition is applied in our case as it is much faster as well as the matrix to be solved is positive semi-definite and can be easily decomposed into a lower and upper triangular matrix. We solve **Ax=b** by first computing the Cholesky decomposition **A= *L. L*^{T}** , then solving **Ly=b** for **y** by **forward substitution** and finally solving $\boldsymbol{L^T x = y}$ for **x** by **back substitution.**

Thus **A= *L. L*^{T}** In order to construct L we proceed as follows:

$$L_{ii} = (a_{ii} - \sum_{k=1}^{i-1} L_{ik}^2)^{1/2}$$

$$L_{ji} = \frac{1}{L_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} L_{ik} L_{jk}\right) \qquad j = (i+1), (i+2) \ldots N$$

To ensure a proper and justifiable values of parameters, we impose some boundary constraint on all five parameters. The bounds are as follows:

α – [0.26 to 0.585];

β – [0.455 to 0.78];

c – [0 to 1];

b_1 – [0 to 1];

b_2 – [(-1) to 1];

The Levenberg Marquardt algorithm provide accurate results and greatly minimize the sum of squared error.

- **Trust Region Reflective algorithm [5.]**

The original problem of fitting Concentration Curve with Piecewise linear model is a bound constraint problem. So an interior trust region approach for non-linear minimization subject to bounds is the best algorithm for fitting the Piecewise linear model to Concentration Time Curve. Trust region method define a region around current iterate within which they trust the model $\psi_k(s)$ to be an adequate representation of the objective function. In practical algorithms, we choose the size of the region according to the performance of the algorithm during previous iterations. If the model is generally reliable, producing good steps and accurately predicting the behavior of objective function along these steps, the size of the trust region is steadily increased to allow longer, more ambitious steps to be taken, On the other hand, a failed step indicates that our model is an inadequate representation of the objective function over the current trust region, so we reduce the size of the region and try again. We define certain terminologies:

Let f(x) =$\sum_{t=1}^{N}(f(t,\alpha,\beta,c,b_1,b_2)-y(t))^2$

Let g(x) $\stackrel{\text{def}}{=} \nabla f(x)$. The vector v(x) $\in R^n$ is defined below.

For each 1≤i≤ n,

(i) If $g_i < 0$ and $u_i < \infty$ then $v_i \stackrel{\text{def}}{=} x_i - u_i$;

(ii) If $g_i \geq 0$ and $l_i > -\infty$ then $v_i \stackrel{\text{def}}{=} x_i - l_i$;

(iii) If $g_i \geq 0$ and $u_i = \infty$ then $v_i \stackrel{\text{def}}{=} -1$;

(iv) If $g_i \geq 0$ and $l_i = -\infty$ then $v_i \stackrel{\text{def}}{=} 1$;

For any s $\in R^n$, diag(s) denotes an n-by-n diagonal matrix with the vector s defining the diagonal entries in their natural order. Moreover, for any nonsingular matrix A $\in R^{n*n}$ and any l>0, A^{-l} denotes the inverse ofA^l, where A^l is the l^{th} power of A. Using this notation , we define:

D(x) $\stackrel{\text{def}}{=}$diag ($|v(x)|^{-1/2}$)

Matrix H is Hessian matrix which can be approximated by

H= $J^T J$ (J is the Jacobin matrix).

Matrix J^v is defined as:

$J^v \stackrel{\text{def}}{=}$ diag (sgn (g(x))) (sgn stands for signum function)

Matrix C is defined as:

C $\stackrel{\text{def}}{=}$ D(x)diag(g(x))$J^v D(x)$

The algorithm can be stated as follows where k stands for k^{th} iteration:

I. $Compute\ f_k, g_k, D_k, H_k\ and\ C_k\ ; define\ the\ quadratic\ model.$

$$\psi_k(s) \stackrel{\text{def}}{=} g_k^T s + \frac{1}{2} s^T (H_k + C_k) s$$

II. Compute a step s_k, with $x_k + s_k \in$ int (F), based on the Sub-problem. **[6.] [7.]**

$$\min_{s} \{ \psi_k(s) : ||D_k s|| \leq \Delta_k \}$$

III. Compute

$$\rho_k = \frac{f(s_k + x_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k}{\psi_k(s_k)}$$

IV. If $\rho_k > \mu$ then set $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$

V. Update Δ_k as specified below.

Updating Trust Region Size Δ_k

1) If $\rho_k \leq \mu$ then set $\Delta_{k+1} \in (0, \gamma_1 \Delta_k]$.

2) If $\rho_k \in (\mu, \eta)$ then set $\Delta_{k+1} \in [\gamma_1 \Delta_k, \Delta_k]$.

3) If $\rho_k \geq \eta$ then

If $\Delta_k > \Lambda_l$ then

Set $\Delta_{k+1} \in$ either $[\gamma_1 \Delta_k, \Delta_k]$ or $[\Delta_k, \gamma_2 \Delta_k]$,

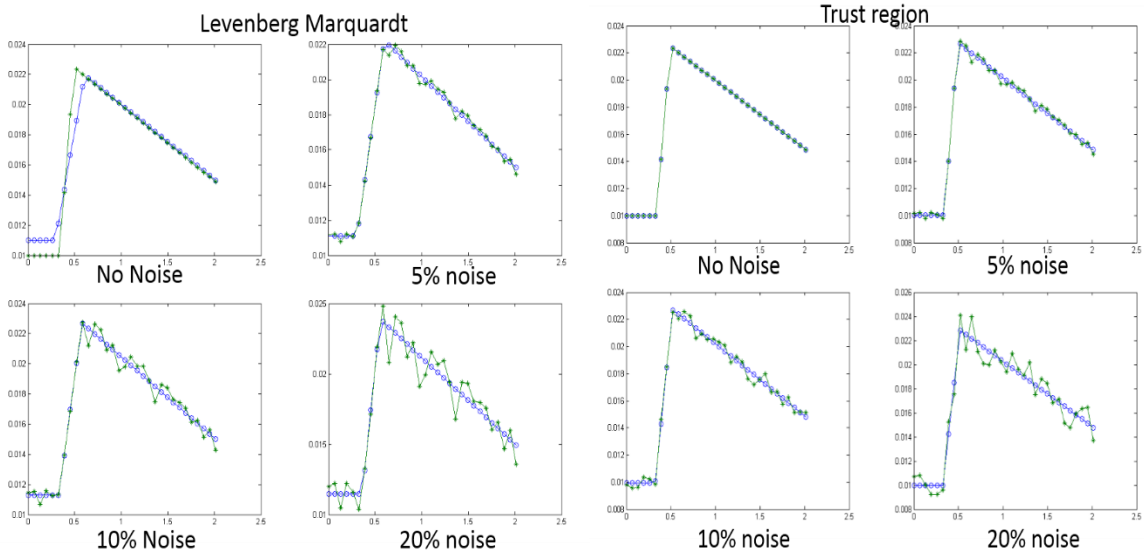
Otherwise

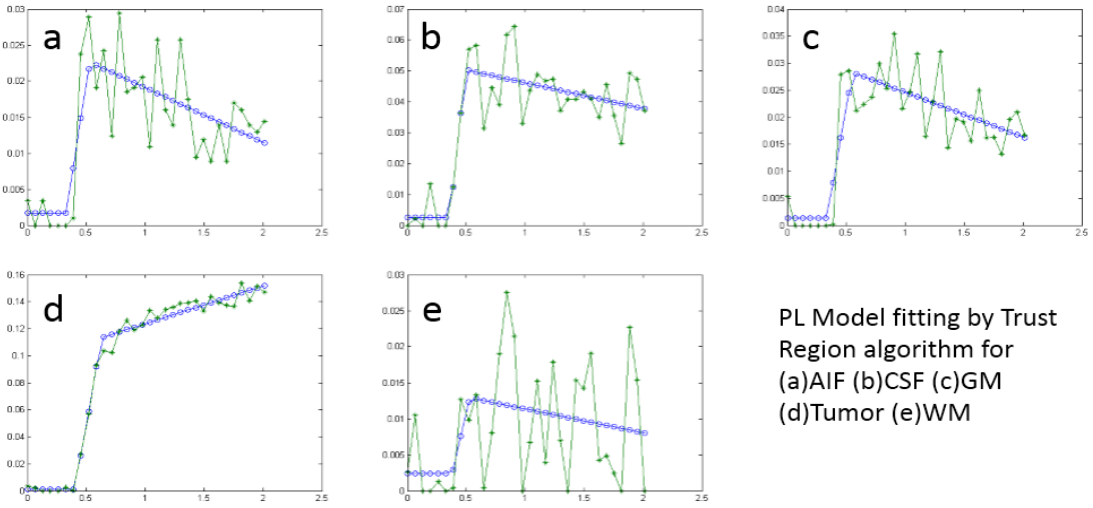
Set $\Delta_{k+1} \in [\Delta_k, \min(\gamma_2 \Delta_k, \Lambda_u)]$.

By following the algorithm in this manner and initialing μ=0.25, η=0.75, Λ_l=1, Λ_u=7.85, γ_1=0.5, γ_0=0.0625, γ_2=2 we can implement the trust region algorithm successfully.

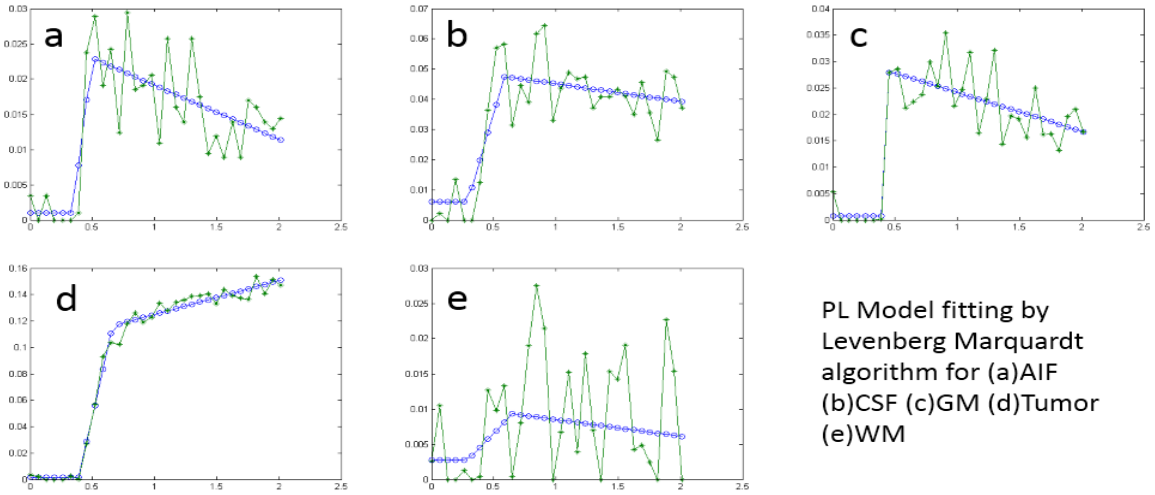
4) **Results and Discussion**

Simulated models to test the accuracy and robustness of the algorithm and code.



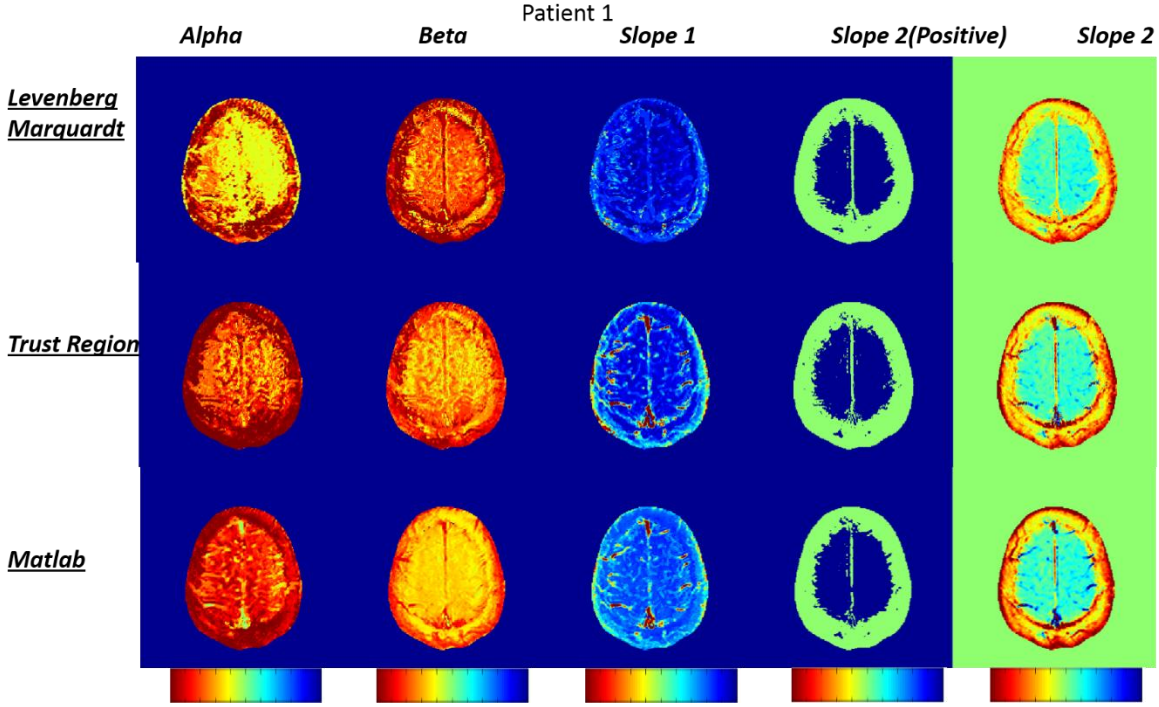


PL Model fitting by Trust Region algorithm for (a)AIF (b)CSF (c)GM (d)Tumor (e)WM

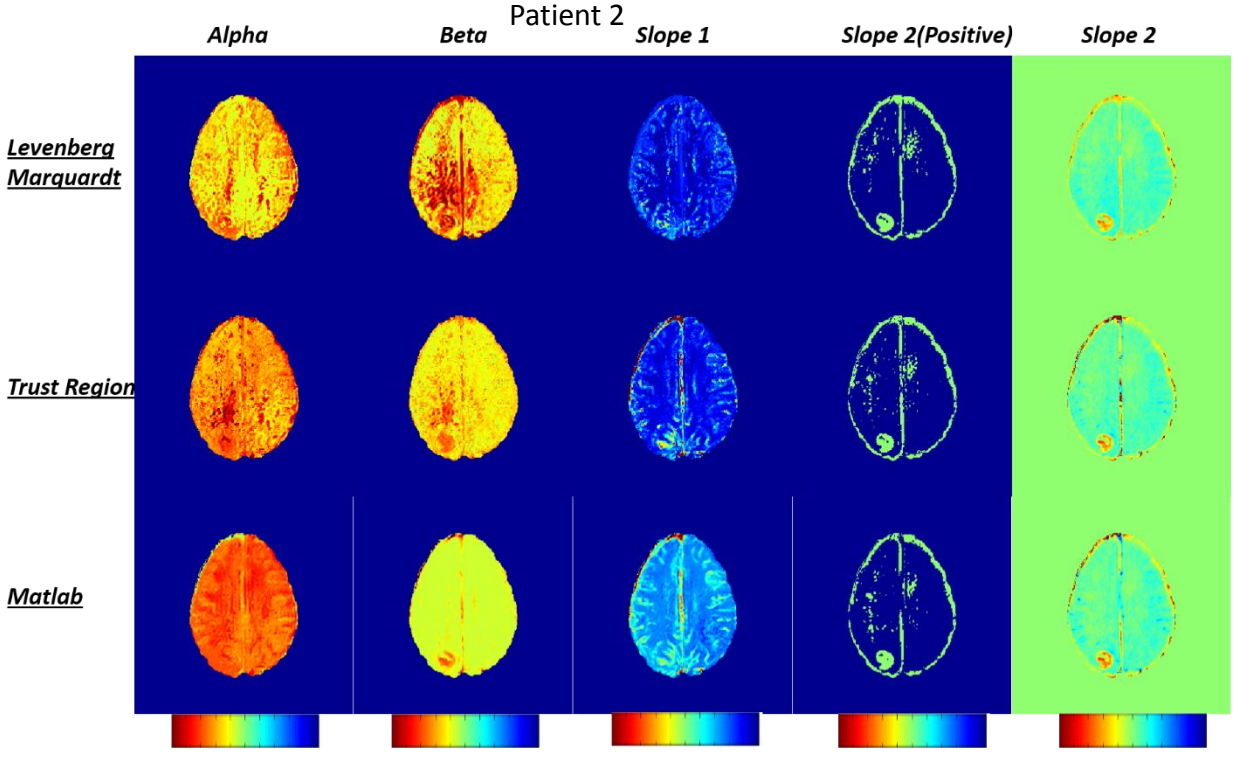


PL Model fitting by Levenberg Marquardt algorithm for (a)AIF (b)CSF (c)GM (d)Tumor (e)WM

Patient 1



Patient 2



5) Conclusions:

We have successfully implemented Trust Region and LM algorithms as a CMex file and linked it to Matlab. The Trust Region algorithm works more accurately than Levenberg-Marquardt algorithm for the bound constraint problems. The Piecewise Linear fitting on the Concentration Time Curve (CTC) of DCE-MRI data is more accurately fitted by Trust Region algorithm than Levenberg-Marquardt algorithm. However, the implementation of Trust Region algorithm through a Mex file(C code) is slightly less accurate than Matlab based built-in routine that implements the same algorithm. However the Mex file(C code) is faster than Matlab code. The future work would be to improvise the Mex file(C code) so as to obtain more accurate results which are much faster than Matlab based routine.

References:

1. Tofts PS, Kermode AG, 1991. Measurement of the Blood Brain Barrier Permeability and Leakage Space using Dynamic MR Imaging: Fundamental Concepts. Magn Reson Med; 17:357–67.
2. Tofts PS, 1997. Modeling tracer kinetics in dynamic Gd-DTPA MR imaging. J Magn Reson Imaging; 7:91-101.
3. Singh A, Rathore RKS, Gupta RK, Haris M, Rathore DKS, Verma SK, Purwar A, Bayu G, Sarma MK, Singh JK, 2007a. Fitting of the Piecewise Linear Function to Signal Intensity Time Curve and its Application in improving the Analysis of Concentration Time Curve of Dynamic Contrast Enhanced-MRI Data. Proceedings of the 15th Joint Annual Meeting ISMRM-ESMRMB; No. 2240.
4. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in C, The art of Scientific Computing, Second Edition, page 683-688
5. T. F. Coleman and Y. Li, An interior trust region approach for non-linear minimization subject to bounds, SIAM J. Optim., 6 (1996), pp. 418-445
6. J. J. More and D.C. Sorensen, Computing a trust region step, SIAM J. Sci. Statist Comput., 4 (1983), pp. 553-572
7. Jorge Nocedal, Stephen J. Wright, Numerical Optimization, page 77-85