Proposal

Student: Hardik J. Parikh
Chair: Prof. Alan. Kaminsky

Topic: Solving MRI Spin Relaxometry Problem (using PJ)

Abstract:

Magnetic resonance imaging (MRI) is an imaging technique used primarily in medical profession to produce high quality images of the inside of the human body, without using X-Rays. The process of obtaining the spin density spectrum from the time samples of the spin signal for each pixel of a magnetic resonance image is known as MRI Spin Relaxometry [5]. The spin density spectrum is obtained by performing Linear Regularization on samples of magnetic resonance images obtained at different times. The spin density spectrum obtained from body, is used by Radiologists to diagnose a disease in early stage.

However the computation involved is pretty substantial and processing the pixels of the images, in parallel, reduces the effective time taken to obtain the spin density spectrum. Parallel computing is suitable to the computations because the value of each pixel is independently calculated. The project involves developing a parallel algorithm implementation using the Parallel Java Library (developed by Prof. Alan. Kaminsky) and MPI implementation in C. The algorithm implementation is to be done using both, Linear Regularization and non-linear Least Squares approach along with proper load balancing. Also, performance comparison is to be done between both the implementations for C as well as Java. Appropriate user-interface is also to be developed for both the implementations.

1. Overview

• Magnetic Resonance Imaging (MRI)

Magnetic Resonance Imaging (MRI) is a technique, used by radiologists, to produce high quality images of the inside of the body. MRI uses electromagnetic waves and radio frequencies (a.k.a RF's) to produce the images of the inside of the human body, that can be used to diagnose a disease in the early stages. Tissues in the body have spin-lattice relaxation rate (R1) and spin-spin relaxation rate(R2). The distribution of R1 and R2, known as relaxometry, generates a signal, from which, an image is generated. R1 and R2 are said to possess diagnostic utility because the rates differ in a healthy tissue as compared to a diseased tissue and hence the disease can be diagnosed from the image of organ.

MRI scanner is a device that produces two-dimensional image of the part of the human body that is being scanned. MRI scanner sends magnetic pulses (RF signals) through the body, and captures the images sequence, to generate one slice of an image. By taking several images, from different angles, a three-dimensional image is constructed. Thus, each slice's data comprises of M images, which are taken at time values t and consists of image comprising of R * C pixels [5].

MRI Spin Relaxometry Problem

The main problem here is to find the spin density spectrum from the noisy MRI signal, which generated the signal. Thus, this is the inverse problem i.e. figuring out the input that generated the known output.

The dependency between the signal obtained from the body (which generates the image), the spin density and the relaxation rate (comprising of R1), is non-linear and is given by the equation: -

$$S(t) = \rho (1 - 2e^{-xt})$$

ρ = spin's density
x = spin's relaxation rate (sec^{-1}).
t = time at which image was taken

• Parallel Computing

Parallel Computing is the application of multiple processors to solve a single task. There are two major categories of parallel computing: -

Shared Memory Multiprocessors (SMP) - There is a single shared memory between all the processors, with processors just having their respective caches. OpenMP is the Middleware Standard for SMP computers.

OpenMP is thread based and relies on inter-thread communication and synchronization between threads. The programming in OpenMP can be done in C, C++, Fortran and Java with additional directives. There is also a pre-compiler involved, which translates the programs with directives to the program with threads.

E.g. paradise.cs.rit.edu, paragon.cs.rit.edu, parasite.cs.rit.edu

Clusters - All processors have their own memory. Thus, communication overhead is not that significant as compared to SMP computers. MPI (Message Passing Interface) is the Middleware Standard for Cluster computers.

MPI is process based and thus relies on inter-process communication. The programming in MPI can be done in C, C++ and Java using the API's (libraries). Thus, no pre-compiler is necessary and processes can be explicitly programmed.

E.g. paranoia.cs.rit.edu

Both, MPI and OpenMP standards are not based on Object Oriented Paradigm. Parallel Java (PJ) is a middleware standard, developed by Prof. Alan Kaminsky of Rochester Institute of Technology, to program in Object Oriented Paradigm, specifically in Java. Programming in PJ can be done for SMP computers, clusters as well as hybrid computers.

▪ MRI Spin Relaxometry Problem

As mentioned above, the dependency between the spin relaxation rate (x) and spin density (ρ) is non-linear. According to the inversion recovery sequence, the dependency is shown by equation

$$S(t) = \rho\,(1 - 2e^{-xt}) \qquad \text{------------------------------ (1)}$$

ρ = spin's density
x = spin's relaxation rate (sec^{-1})
t = time at which image was taken (sec)

The figure shown below is the graph of S vs. t when x = 0.5 and ρ = 1000. Diagram is plotted taking into consideration a single spin of only a single tissue. But a voxel (smallest resolvable element in MRI) [4] can contain more than one tissue.

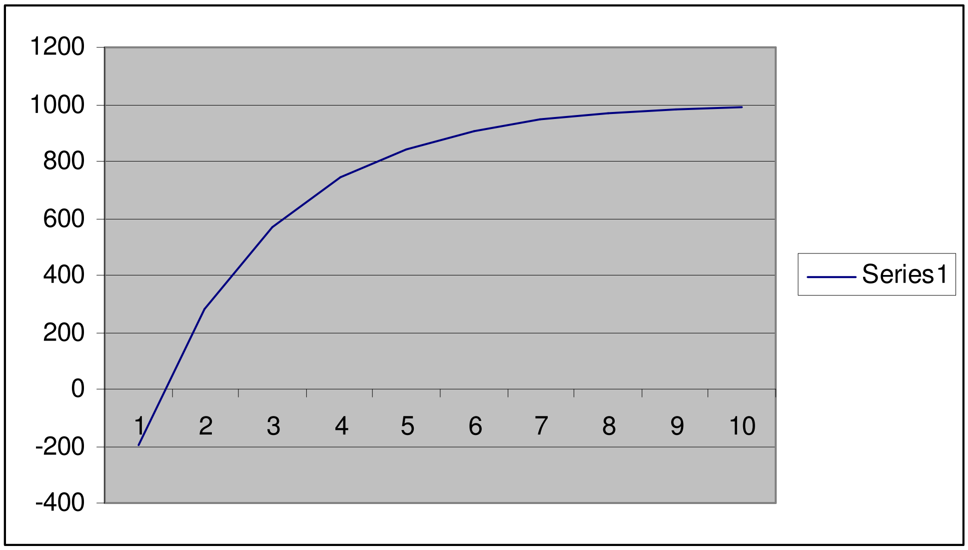

Figure 1

Thus, if there are two tissues, then the diagram becomes as shown below when ρ1 = 750, x1 = 1.5 for tissue 1, ρ2 = 250 and x2 = 0.5 for tissue 2. The resulting pixel's signal value S, at a given time t, is the summation of signals from all different kind of tissues, which is as depicted in the diagram, shown below, for the tissues with above stated values. Thus, the signal value S, at time t is

$$S(t) = \sum \rho (1 - 2e^{-xt}) \qquad \text{------------------------------ (2)}$$

where ρ is the spin density and x is the relaxation rate, across all the tissues.

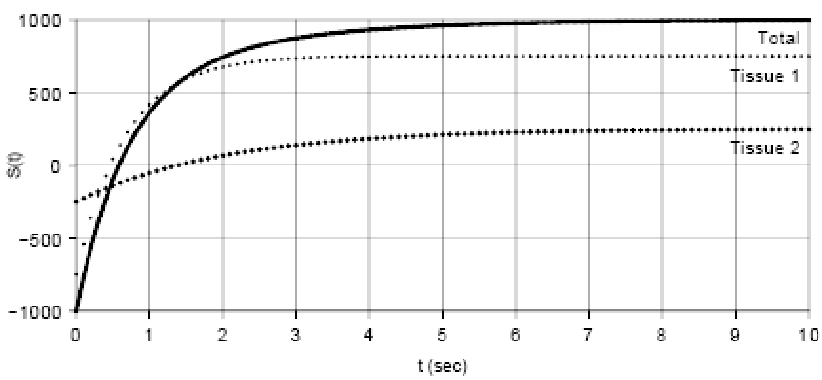

Figure 2

In the above diagrams, both, ρ and x, are shown in terms of the time t. But we can express ρ and x in other form of graph, if we take ρ as y-axis and x as x-axis. Considering only two tissues and taking the above ρ and x values, the plotted graph would look as shown below:

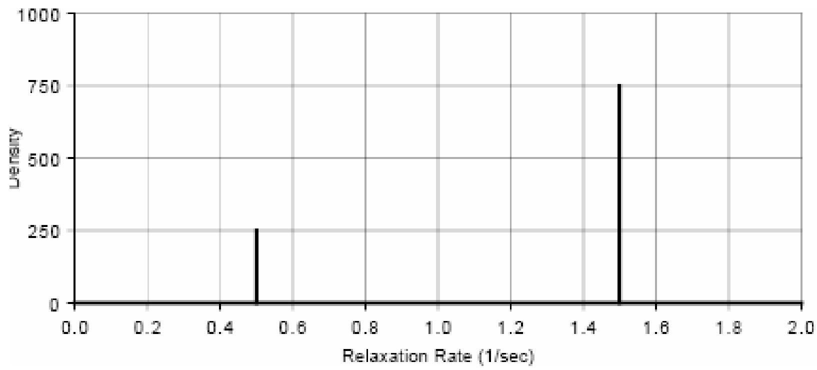

Figure 3

Real MRI scans would not have these much discrete values of ρ at various values of x. Instead, the values of ρ will be clustered around the possible values of x. Thus, in reality, above depicted, would look like as shown below

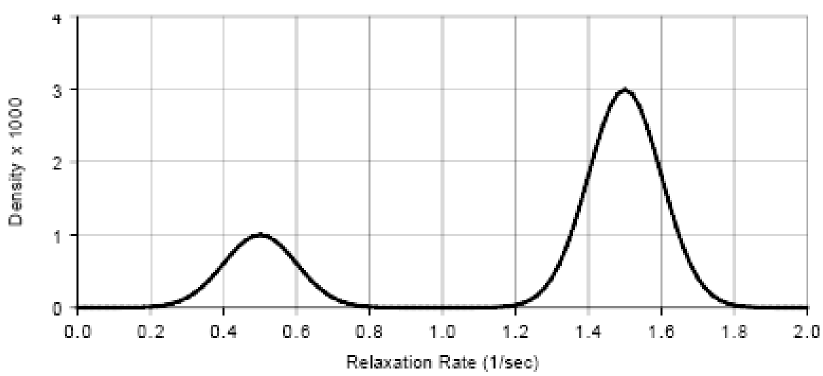

Figure 4

The ρ is a continuous function of x. Thus, the equation 2 transform to the following equation

$S(t) = \int \rho(x)\,(1 - 2e^{-xt})\,dx$ ------------------------------ (3)

Where ρ(x), known as spin density spectrum, is continuous function around the values of x and thus integral takes place of summation, resulting into above equation.

Also, the signal diagrams as depicted in Figure 1 and Figure 2 are not so ideal. There are some measurement errors induced due to MRI instruments. Thus, the original signal obtained looks somewhat like as shown below

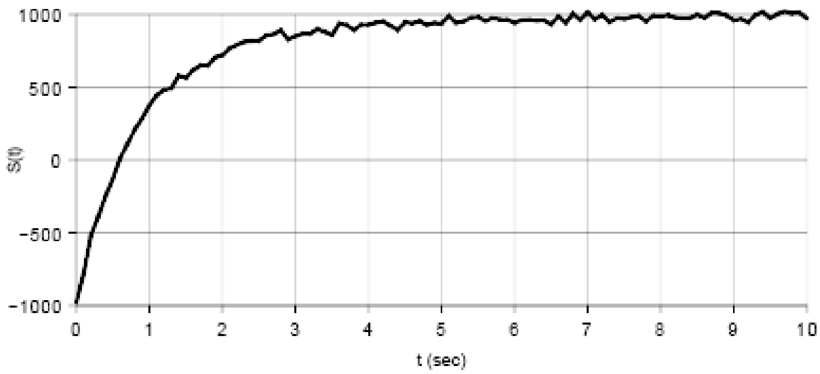

Figure 5

Thus, MRI Spin Relaxometry problem is obtaining spin density spectrum ρ(x) values from value of signal S(t) at time t. Its an Inverse Problem, because from output, we have to track back the input value that generated the output. This has to be done for each pixel of MRI image because each pixel contains unequal number of tissues and thus different spin spectra.

▪ Previous work done:

There is some work done in the area of solving MRI Spin Relaxometry Problem.

1. Running parallel serial algorithms

   Dr. Hornak, along with Prof. Schaller, tried to solve the problem by running parallel versions of the serial algorithm that used to solve the MRI Spin Relaxometry problem.

2. Running parallel algorithm

   Prof. Kaminsky tried to solve the problem by designing and developing an algorithm that was parallel.

▪ Approach to MRI Spin Relaxometry Problem:

There are basically two methods for solving the problem and finding ρ(x) values.

1. Selecting the values of spin relaxation rate (x) in some acceptable range and then using particular method like Constrained Linear Regularization and finding non-negative least squares solution. This particular method was used by Prof. Kaminsky to solve the problem.

2. Figuring out the values of ρ(x) and x, at the same time. The relationship between both is non-linear, as seen from equation (3). Thus, we need to perform Non-Linear Least Squares approach and obtain the non-negative least squares solution.

The method to be used in the program is still to be decided considering other factors like the feasibility and complexity of the solution. Again, tradeoff parameters like time and resources used to obtain the solution also need to be taken into consideration.

3. Constrained Linear Regularization Approach

Linear regularization, also known as Tikhonov regularization [10], seeks to solve the following inverse problem. An output function g(y) is defined in terms of an input function f(x) and a response kernel r(x, y) as:

$$g(y) = \int f(x)r(x, y)dx \quad (4)$$

2. Functional Specification

The functional specifications of the project are:

1. Parallel Algorithm implementation is to be done using Linear Regularization technique in C (using MPI) and Java (using PJ).

2. Parallel Algorithm implementation is to be done using Non-Linear Least Squares technique (if feasible) in C (using MPI) and Java (using PJ).

3. Graphical user interfaces are to be developed for both the algorithm implementations.

3. Architectural Overview (Design specifications)

1. The implementation of algorithm is to be done on clusters. Each node in the cluster gets the same set of input data for processing.

2. Each node performs the algorithm (either using Linear Regularization or Non-linear least squares) and produces the output. The output format from all the nodes of the cluster also remains the same.

3. Also, advance load balancing like Master-Worker technique is to be implemented in the implementations.

4. Again, the implementations are to be done on clusters.

4. Scope/Deliverables of the Project:

One of the main research areas of this project is solving the problem with Non-Linear Regularization and obtains the non-negative least square solution for the values of ρ(x).

The implementation of this project is to be done using MPI C library and the PJ Library. The list of deliverables includes:

1. Technical Report of the project.
2. The code (written in C as well as Java) done for the project.
3. User manual.
4. Graphical user interface for demonstration of the project.
5. List of References.

References:

1. Basics of MRI -- http://neuroland.com/neuro_images/mri_basics.htm

2. Joseph P. Hornak. Basics of MRI. In Rochester Institute of Technology Chemistry and Imaging Science. http://www.cis.rit.edu/htbooks/mri/

3. Joseph P. Hornak. Basics of NMR. NMR-MAIN. In Rochester Institute of Technology Chemistry and Imaging Science. http://www.cis.rit.edu/htbooks/nmr/bnmr.htm

4. Andrew P. Bak, Joseph P. Hornak, and Nan C. Schaller. From impractical to practical: Solving an MRI problem using parallelism. In Rochester Institute of Technology B. Thomas Golisano College of Computing and Information Sciences 2005 Conference on Computing and Information Sciences, January 2005. https://ritdml.rit.edu/dspace/bitstream/1850/423/1/ABakAbstract2005.pdf

5. Alan Kaminsky and Luke McOmber. Solving an MRI Spin Relaxometry Problem with Parallel Computing. In Rochester Institute of Technology B. Thomas Golisano College of Computing and Information Sciences 2005. http://www.cs.rit.edu/~ark/sr/sr20050627.pdf

6. Stephen W. Provencher. A constrained regularization method for inverting data represented by linear algebraic or integral equations. Computer Physics Communications, 27:213-227, 1982.

5. All the images taken as snapshots from http://www.cs.rit.edu/~ark/sr/sr20050627.pdf

10.