New Mathematical Data Analysis for Magnetic Resonance Elastography

by

Kogo Yoshikawa

DISSERTATION

Submitted for the degree of

DOCTOR OF PHILOSOPHY

in

SCIENCE

at

DEPARTMENT OF MATHEMATICS, GRADUATE SCHOOL OF SCIENCE

HOKKAIDO UNIVERSITY

JAPAN

May, 2013

Ac proinde hæc cognitio, ego cogito, ergo sum,

est omnium prima & certissima, quæ cuilibet ordine philosophanti occurrat.

Kogo Yoshikawa

March 2013

Mathematics

New Mathematical Data Analysis for Magnetic Resonance Elastography

Abstract

For the diagnosing modality called MRE (Magnetic Resonance Elastography), the displacement vector of a wave propagating in a human tissue can be measured. The average of the local wave length of the wave from this measured data could be an index for the diagnosing, because the local wave length becomes larger when the tissue is stiffer. By assuming that the local form of the wave is given approximately as multiple complex plane waves, we identify the real part of the complex linear phase of the strongest plane wave of this multiple complex plane waves, by first applying the FBI transform (Fourier-Bros-Iagolnitzer transform) with an appropriate size of Gaussian window and then taking the maximum of the modulus of the transform with respect to the Fourier variable. We call this method LWV method. The real part of the linear phase is nothing but the real inner product of the wave vector and the position vector. Similarly the imaginary part of the linear phase describes the attenuation of the wave and it is given as a real inner product of a real vector and the position vector. This vector can also be recovered by our method called the LAV method. We also apply these methods to design some de-noising and filtering for noisy MRE data. Although we restricted applying our methods to MRE wave images, it can also be applied to other wave images.

Table of Contents

List of Figures

Chapter 1

Introduction

A new measurement modality called MRE (Magnetic Resonance Elastography) consists of an MRI (Magnetic Resonance Imaging), mechanical vibration system and additional MRI pulse sequence called MSG (Motion Sensitizing Gradient) synchronized with the time harmonic vibration generated by the vibration system. Given a time harmonic external vibration generated by the vibration system to a human body which yields a wave in the human body, MRE gives a snapshot of the displacement vectors of the wave over each slice of the human body. We call this snapshot MRE data. The slice can be the cross section of the body by any one of the x_1-x_2 plane, x_2-x_3 plane and x_3-x_1 plane, where $(x_1, x_2, x_3)$is the Euclidean coordinates. If we can recover the stiffness of the tissue in a human body from the MRE data, MRE can provide a realization of doctors' palpation inside human bodies which has been dreamed about by all the doctors for many years(cf. [1],[2]). We call any procedure to recover the stiffness or extract any information about the stiffness MRE data analysis.

There are two kinds of MRE data analysis. The one is the model independent MRE data analysis which only assumes that any local wave forms of the wave is given approximately as multiple complex plane waves, and recover the real part of the complex linear phase of the strongest wave in this multiple complex plane waves which can be represented by the so called wave vector. We call this wave vector divided by the angular frequency of vibration the *local wave vector* of the multiple complex plane waves. The other is the model dependent MRE data analysis which considers some partial differential equation to describe the wave and stiffness as its solutions and coefficient, respectively, and recover the coefficient from the MRE data via this equation. We will call such a partial differential equation the PDE model. In this paper we will give a model independent MRE data analysis based on the FBI transformation (Fourier-Bros-Iagolnitzer transform). For the model dependent MRE data analysis see for instance [2], [3], [4], [5], [6] and the references therein.

As it is well known that the wave length becomes larger if the tissue becomes stiffer. In terms of the wave vector this means that the wave vector becomes shorter if the tissue become stiffer. Hence, by looking at the wave vectors in the tissues, we can qualitatively know a change of their stiffness. Since the modeling error is always a big problem in the MRE data analysis, the model independent analysis has some advantage if it is not so important to recover the stiffness quantitatively but qualitatively.

In the rest of this section we will explain more precisely about our model independent MRE data analysis. Since the wave length of the longitudinal wave in human tissue is too long to be observed, we can only observe shear waves when the tissues is isotropic tissues and quasi-shear waves if the tissues is anisotropic. Suppose that a shear wave or quasi-shear wave is mainly propagating toward the x_2-direction and we are looking this wave over a slice parallel to the x_1-x_2 plane which is the cross section of a human body. Then, let $\varphi = \varphi(x_1, x_2)$ be the one of the component of displacement vector of this wave perpendicular to x_2 direction, say x_3 component. We also take such a wave whose phase of the vibration is 90 degrees advanced and denotes its component similar as before by $\psi = \psi(x_1, x_2)$. For our data analysis, it is more convenient to consider

$$u = u(x) = \varphi(x) - i\psi(x) \tag{1.1}$$

than considering φ and ψ separately.

A naive way of looking at u near a point p in the cross section is that it is locally given by a finite linear combination of the complex plane wave $ae^{\omega(\alpha+i\beta)\cdot(x-p)}$ with an amplitude $a \in \mathbb{C}$, vectors $\alpha, \beta \in \mathbb{R}^2$ which do not depend on $x = (x_1, x_2)$ and the angular frequency $\omega/(2\pi)$ of the vibration system. Note that α and β describe the attenuation and propagation direction of the wave u, respectively. We call this form of u the *local single-wave form* if the linear combination consists of just one term and *local multiple-wave form* if otherwise.

Let u be described approximately as the local multiple-wave form near a point p in a region of interest (ROI) of a human tissue. Then, by our method called LWV method (Local Wave Vector method) and LAV method (Local Attenuation Vector method) which are based on the FBI transformation, we can recover β and α in the strongest local single-wave form of the local multiple-wave form. We will call these β and α in this strongest local single-wave form the *local wave vector* and *local attenuation vector*, respectively. Here the FBI transformation is a weighted Fourier transformation with the Gaussian window centered around p. Once we have recovered β at several points in the ROI, we can filter the wave fields with many waves interfering with each others in the ROI to a single major wave. If the ROI is located near the boundary of tissue, for instance the boundary between a tissue and organ, there is an interference of incoming waves and reflected waves from the boundary. In such a place of ROI the wave length and amplitude of wave could become smaller than the other parts of the ROI and hence the profiles of the distribution of the local wave vectors there will become quite complicated. But by our filtering method based on the LWV method, we can extrapolate the major wave up to the boundary in this ROI. As a consequence, we can get very clear filtered wave image having just a major wave in this ROI. We call this de-noising method the *LWV de-noising* of wave.

To transform the recovered local wave vector β and local attenuation vector α to the stiffness of tissue, we need to have a PDE model. Suppose that our tissue can be considered as an nearly incompressible isotropic viscoelastic medium, then the above u can be considered approximately as the x_3 component of $\operatorname{rot} v$, where v denotes the displacement vector of the wave and $\operatorname{rot} v$ denotes the rotation of v. Then, each local single-wave form u' of the local multiple-wave form u should satisfies

$$(\rho\omega^2 + (G' + iG'')\Delta)u'(x) = 0 \tag{1.2}$$

approximately in a small neighborhood of p with the density $\rho \approx 10^3 kg/m^3$, the storage modulus G' and loss modulus G''. We remark here the G', G'' can change from one region to another region where the local multiple-wave form of u changes. Further, we remark that u' always satisfies (1.2) approximately, if the tissue is modeled as whichever type of nearly incompressible isotropic viscoelastic media ([3],[7]). Suppose that we have identified β and α in the strongest local single-wave form u' of u. Then, by substituting this local single-wave form into (1.2), we have

$$\rho + (G' + iG'')(\alpha + i\beta) \cdot (\alpha + i\beta) = 0 \tag{1.3}$$

which immediately implies that G', G'' are given by

$$\begin{pmatrix} G' \\ G'' \end{pmatrix} = \frac{\rho}{(|\alpha|^2 - |\beta|^2)^2 + 4(\alpha\cdot\beta)^2} \begin{pmatrix} |\beta|^2 - |\alpha|^2 \\ 2\alpha\cdot\beta \end{pmatrix}. \tag{1.4}$$

Hence (1.4) gives the link between α, β and G', G''.

The rest of the paper is organized as follows. In Section 2.1 and 2.2 we give the theories of the LWV method and LAV method, respectively. Then, in the succeeding section we will provide some numerical results for these two methods. Especially, in order to see the effectiveness of these method, we tested our methods by recovering G', G'' of a phantom made of PAAm gel by the MRE group in our university (Prof. J. Gong, Laboratory of Soft and Wet Matter, Hokkaido Univ.) and for a phantom made of agarose gel by Mayo Clinic so that we can compare our results with the other results obtained by different MRE data analysis. In the final section, we will apply our methods to the de-noising and sharpening of the MRE data. Before closing this introduction, we would like to acknowledge Mayo Clinic providing us the data and emphasize that Mayo Clinic is the front runner of the MRE study.

Chapter 2

LWV method and LAV method

2.1 LWV method

In this section we will give the details of the LWV method mentioned in the introduction. Let $W(u;p,\sigma)(\xi)$ be the two dimensional FBI transform (cf.[8]) of a locally integrable function u in $\mathbb{R}^2$ with the Gaussian window of size σ localized around $p \in \mathbb{R}^2$ as follows

$$W(u;p,\sigma)(\xi) = \int_{\mathbb{R}^2} e^{-ix\cdot\xi} u(x) e^{-\frac{|x-p|^2}{2\sigma^2}}\, dx \ (\xi \in \mathbb{R}^2) \tag{2.1}$$

provided that this integral converges which is the case for the local multiple-wave form u. This transformation is also called the two dimensional continuous wavelet transform (cf. [9]). If we take $u(x)$ as a local single-wave form $u(x) = ae^{\omega(\alpha+i\beta)\cdot(x-p)}$, then $W(u;p,\sigma)(\xi)$ is expressed as

$$\begin{aligned} &W(u;p,\sigma)(\xi) \\ &\quad = 2\pi a\sigma^2 \exp\left[i\omega^2\sigma^2\alpha\cdot\beta + \frac{\omega^2\sigma^2|\alpha|^2}{2}\right] \exp\left[-i(p+\omega\sigma^2\alpha)\cdot\xi - \frac{\sigma^2|\xi-\omega\beta|^2}{2}\right] \end{aligned} \tag{2.2}$$

Here we note that $\omega\beta$ is a unique Gaussian peak of $W(u;p,\sigma)$. The details of this derivation is given in Appendix. The maximum $\arg\max_{\xi} \left|W(u;p,\sigma)(\xi)\right|$

of the modulus $\left|W(u;p,\sigma)(\xi)\right|$ for $\xi \in \mathbb{R}^2$ is clearly achieved at $\xi(p) = \xi = \omega\beta$. Hence, we have

$$\beta = \arg\max_{\xi} \left|W(u;p,\sigma)(\xi)\right| / \omega. \tag{2.3}$$

Here we note that σ^2 sitting in the denominator of the exponential of the Gaussian window will sit in the numerator of $W(u;p,\sigma)(\xi)$. This is nothing but the Heisenberg uncertainty principle about the window sizes in the *real space* x and *Fourier space* ξ. We have an option to tune a parameter σ that influences the localization in the real space and Fourier space.

If $u(x)$ is given as the multiple-wave form

$$u(x) = \sum_n c_n e^{\omega(\alpha_n + i\beta_n)\cdot(x-p)}, \quad c_n \in \mathbb{C} \tag{2.4}$$

around p, $\arg\max_{\xi} \left|W(u,p,\sigma)(\xi)\right|$ can expect to give the local wave vector β_n of the strongest single wave form $c_n e^{\omega(\alpha_n + i\beta_n)\cdot(x-p)}$ in its modulus. This can be understood by accepting a very reasonable interpretation which says that the Gaussian peaks of the FBI transformed u is well separated in most cases. We call this method to obtained the local wave vector β_n obtained above the local wave vector the LWV method.

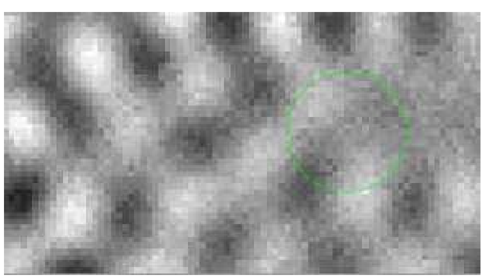

Fig. 2.1: Example of function Re $u(x)$ of Eq(2.4)

Fig. 2.2: Localization of Fig.2.1 by Gaussian

We will show in several figures how the LWV method is performed. The above two figures show the localization by a Gaussian window. Since the key to the LWV method is the assumption that the local approximate expression of the wave u is given by (2.4), we need to localize u to find the local wave vector of u.

The next Fig.2.3 is what can be seen in the Fourier space ξ. More precisely this is the FBI transformed image of Fig.2.2.

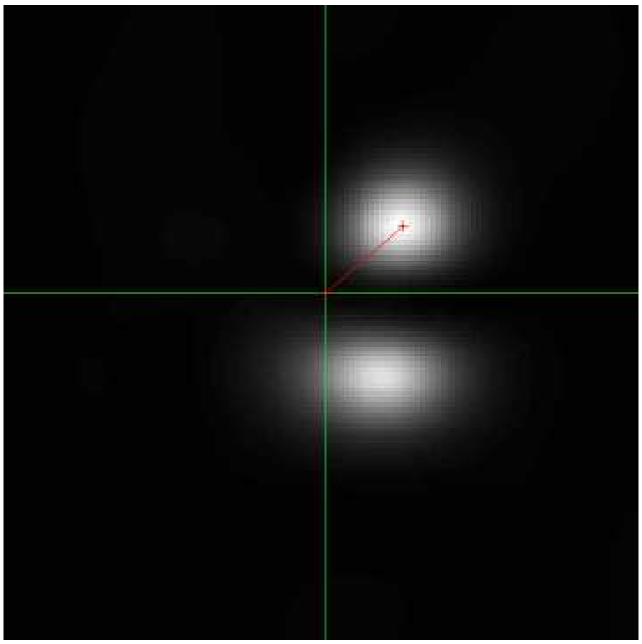

Fig.2.3: Example of $\left|W(u;p,\sigma)(\xi)\right|$ in Eq(2.3)

In this figure, we find two Gaussian peaks in Fig.2.3 which means that there are basically two different directions to which the waves are propagating in Fig.2.2. This reasonably fits to Fig.2.1. It seems that in the Fourier space, the position of the peak of Gaussian is not strongly interfered by those of other peaks of Gaussian. Hence, the separation of interfered waves in the Fourier space should be quite good.

We repeated this process around enough sampled points and plotted the local wave vectors at the sampled points to obtain the following figure in which the sampled local wave vectors are superimposed over the figure of the real part of u.

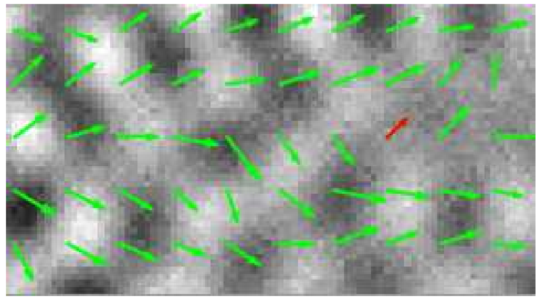

Fig.2.4: The vectors represent the wave vectors of the strongest waves. The red vector corresponds to that of Fig.2.3.

Let us finish this section by giving several comments on the method. First of all, concerning the choice of the Gaussian window size σ, we usually take σ in the range from half wave length to one wave length for having reasonable recovery of β by our experiences. Taking arg max may misfit β when there exists a strong noise with a specific frequency. But, for MRE data, it usually has only Gaussian type white noise that does not have a specific frequency. Finally, we would like to emphasize here an advantage of the LWV method. That is, even in the case that several waves coming from different directions merge at a point $p \in \mathbb{R}^2$, the effect of each wave is quite localized in the Fourier space, so that if there are several different waves merging at p, we can separate these major propagating directions by the LWV method.

2.2 LAV method

We will show in this section how to recover the local attenuation vector of the strongest wave in the local multiple-wave form (2.4). To begin with we first assume that $u(x)$ is given as a local single-wave form around a point $p \in \mathbb{R}^2$. Then the vector α at p in the local single-wave form with the wave vector β can be recovered by

$$\alpha = -\frac{1}{\omega\sigma^2}\big(\nabla_\xi \theta(p;\xi) - p\big), \tag{2.5}$$

where $\theta(p;\xi)$ is defined by

$$\theta(p;\xi) := \arctan\left(\frac{\operatorname{Im} W(u;p,\sigma)(\xi)}{\operatorname{Re} W(u;p,\sigma)(\xi)}\right). \tag{2.6}$$

In fact, substituting (2.2) into the right hand side of (2.5), introducing θ_0 as an initial phase that does not depend of ξ, the right hand side of (2.6) becomes

$$\begin{aligned}
(\mathrm{RHS}) &= \arctan\left(\frac{\sin(-\omega\sigma^2\alpha\cdot\xi+\theta_0)}{\cos(-\omega\sigma^2\alpha\cdot\xi+\theta_0)}\right) &(2.7)\\
&= \arctan\left(\tan(-\omega\sigma^2\alpha\cdot\xi+\theta_0)\right) &(2.8)\\
&= -\omega\sigma^2\alpha\cdot\xi+\theta_0. &(2.9)
\end{aligned}$$

Then, we will obtain (2.5) by taking the gradient of $\theta(p;\xi)$ with respect to ξ at $\xi=\omega\beta$. In order to compute the gradient numerically we used the following least square method. Let $\xi=(\xi_1,\xi_2)$, $\beta=(\beta_1,\beta_2)$ and denote $m=\xi_1-\beta_1$, $n=\xi_2-\beta_2$. Then, the least square minimization to compute the gradient $(\nabla_\xi\theta)(p;\beta)$ is

$$\arg\min_{\alpha_1,\alpha_2}\sum_{m,n} w(m,n)\left(\theta(p;\xi)-\theta(p;\beta)-\alpha_1 m-\alpha_2 n\right)^2 \tag{2.10}$$

where $w(m,n)=e^{-\frac{m^2+n^2}{2s^2}}$ with some constant $s>0$.

Fig.2.5 given below illustrates the 3-dimensional view of this minimization.

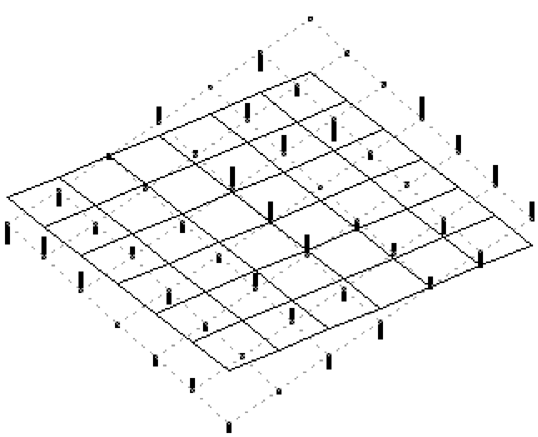

Fig.2.5: Least square method for 2 dimensional plane

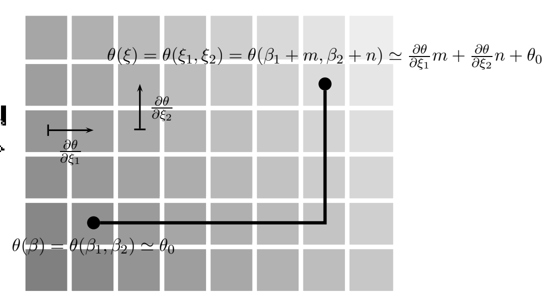

Fig.2.6: θ with respect to ξ

Even for u having the local multiple-wave form, we apply the same formula (2.5) to compute the attenuation vector α associated with the local wave vector β by expecting that we have already picked up the strongest local single wave form with the local wave vector β in the local multiple wave form and the contribution coming from the other local single wave forms is small. This is the precise description of the LAV method.

In the rest of this section, we give a reminder for programming the LAV method. That is to handle the discontinuities of (2.6) at $\theta = \pm\pi/2$. Instead of using the formula

$$\theta(p;\xi) - \theta(p;\xi_0) = \arctan\left(\frac{\operatorname{Im} W(u;p,\sigma)(\xi)}{\operatorname{Re} W(u;p,\sigma)(\xi)}\right) - \arctan\left(\frac{\operatorname{Im} W(u;p,\sigma)(\xi_0)}{\operatorname{Re} W(u;p,\sigma)(\xi_0)}\right), \tag{2.11}$$

we used as its reasonable approximation the following formula

$$\theta(p;\xi) - \theta(p;\xi_0) = \operatorname{Im}\left(\frac{W(u;p,\sigma)(\xi)}{W(u;p,\sigma)(\xi_0)}\right). \tag{2.12}$$

Chapter 3

Numerical testing of LWV and LAV methods

In this section which consists of three subsections we will show some results on the numerical testing of our LWV and LAV methods. As we have mentioned before in the introduction, the methods are model independent methods, but we will also show the numerical recoveries of G', G'' in order to see the quantitative performance of our methods. The first subsection is for the numerical testing of our methods for simulated data and the succeeding two subsections are that for the real data obtained for phantoms by Mayo Clinic and MRE study group in our university, respectively. We call these real data the *phantom data* for simplicity. We did not test our methods for any clinical data, but the phantoms have some values close to the tissues of human levers.

3.1 Simulated data

For simulated data in an unbounded domain without any boundary and noise, the results of the numerical testing of our methods are perfectly fine. Hence, we will directly go to the numerical testing for simulated data in a bounded domain with boundary and a noise. We added a considerably large Gaussian type noise to a simulated datum in order to see whether our methods work for the data with poor S/N ratio less than 0.1 which could be the case for real data. For the simulated datum, we made the length of α ten times longer than that of β which is the case for the phantoms data. Hence, the attenuation of wave is small. In other word, the amplitude of wave gradually decreases as the wave propagates. The superimposed arrows in the figures below show the recoveries of β and α. Hence, the variance of α in the figures below is smaller than it looks in Fig.3.2.

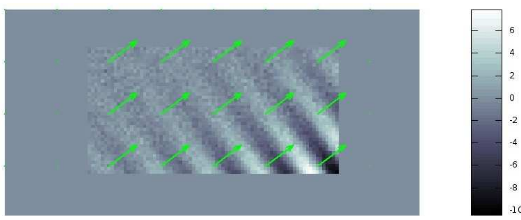

Fig.3.1: Recovered β

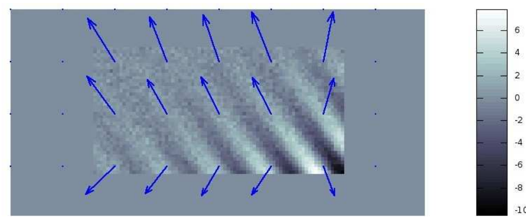

Fig.3.2: Recovered α

If there were no noise, then the recovered α and β should have been just constant vectors with the right upper direction and left upper direction for β and α, respectively. The recovery of β is quite good almost everywhere while that of α is less tolerant to noise and position.

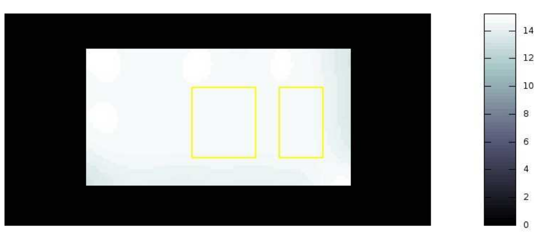

Fig. 3.3: Distribution of G' for simulated noisy data

Estimation of G'	center	right part
average	14.89kP	14.82kP
standard deviation	0.07kP	0.11kP

Table 3.1: Estimation of G'

Next we computed G' by using the formula (1.4). The above figure shows the distribution of the value G', and the table shows the average and standard deviation of the distributed values of G'. We note that the true value of G' was 14.4kP. Hence, we can conclude from these that the recovery of β is quite good. We also observed by doing more numerical testing for simulated data, we observed that the estimate of G' is always stable even under poor S/N ratio like this simulated data. Further we give two remarks. Firstly for example, around the part of upper left corner of Fig.3.1, the signal is much less than background noise and hence we are nearly unable to see the pattern of waves there. Secondly, if α is much smaller than β, the simple approximate formula (cf.[10])

$$G' = \rho/|\beta|^2 \tag{3.1}$$

of G' works well.

We also computed G'' by using the formula (1.4). The figure and table below show the distribution of the value G'' and the average value, standard deviation of the distributed values of G''. These results show that the recovery of G'' is not good in center, because expected value of G'' is 0.69kP. This insufficient recovery of α can be explained as follows. As we have seen before that the recovered β is almost a constant vector, but the recovered α fluctuates near the lower boundary almost completely changing its direction. Then, recalling the formula (1.4), the recovered G'' is influenced by this fluctuation of α which can have negative sign. As far as we know, any MRE data analysis has a difficulty recovering α in an efficient way and we do have the same difficulty.

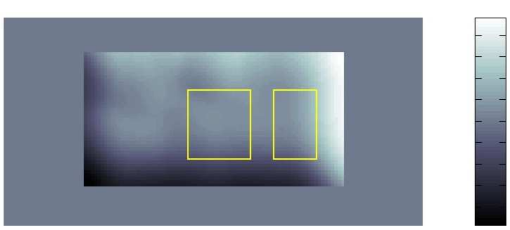

Fig. 3.4: Distribution of G'' for simulated noisy data

Estimation of G''	center	right part
average	0.17kP	0.70kP
standard deviation	0.59kP	0.79kP

Table 3.2: Estimation of G''

3.2 Phantom data from Micro MRE system

Now, we will show the testing of our method to a phantom datum obtained from MRE study group in Hokkaido University. The MRE system in Hokkaido University consists of micro MRI with a 0.3 tesla permanent magnetic, function generator and vibrating system. We call this MRE system the *micro MRE system.*

The resolution of the micro MRI is 1.2mm square per pixel. The data obtained by this micro MRE system for a phantom is given as the backgrounds of Fig.3.5 and Fig.3.6 which are the same data for $\mathrm{Re}\, u$. The phantom is a two layered PAAm gel and it has the cross section given as the rectangular region given in Fig.3.5, Fig.3.6 about 6cm times 12cm which is the plane containing the vibrating source. In this cross section, the location of the vibrating source is at the middle of the left edge and interface of the two layers appears in the middle. The left part of the cross section is stiffer than the right part. Also, the wave is generated from this source by the vibration system with the 250Hz angular frequency and it travels to the right direction. The wave field looks much complicated than what we have seen before for the simulated simple sinusoidal wave and we can observe reflection and refraction of waves at the boundaries and interface, respectively.

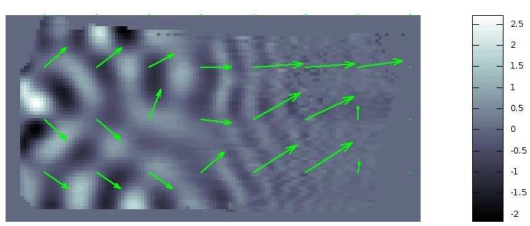

Fig.3.5: Fitting of β

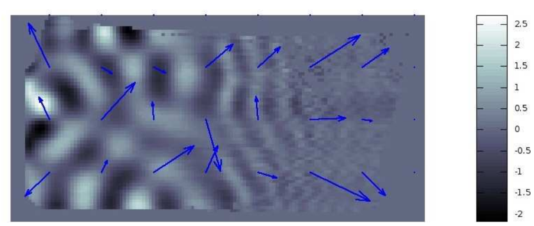

Fig.3.6: Fitting of α

We applied our method to recover β and α. The recovered β and α are shown in Fig.3.5 and Fig.3.6. The result for β given in Fig.3.5 matches quite well to the profile of the wave field. From Fig.3.6, we can see that the direction of α is not the same as direction of β. By plotting the modulus of $u(x)$, i.e.

$|u(x)|$, we can observe that major waves, reflected waves and transmitted waves are mixed together to yield standing waves which have small amplitude at some place and big amplitude antinode at other places creating some nodes. We can observe that α inclines to the nearest node.

By the formula (1.4), we can transform the recovered β, α into G', G''. The recovered G' is given below.

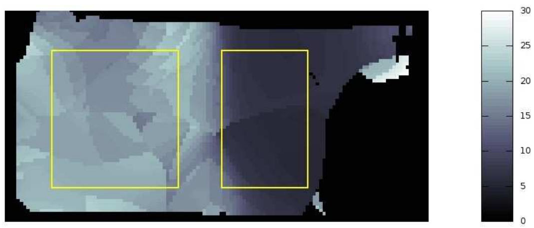

Fig. 3.7: Distribution of G' for two layer phantom

Estimation of G'	center	right part
average	18.56kP	7.00kP
standard deviation	1.47kP	1.83kP

Table 3.3: Estimation of G'

The G' values of the two layered phantom were also measured by a conventional rheometer giving the values 31.1kPa and 10.7kPa for the stiffer and softer parts of this phantom. The frequency of twisting the phantom was 10Hz for this measurement. Since it is known that G' depends on the frequency (cf.[11]), we can not directly compare our result with these G' values. The gray scale values in Fig.3.7 clearly shows the location of the interface. Hence, we can say that our method can show the contrast of the stiffness. This is quite important in clinical application of MRE.

The next figure and table show the recovered G''.

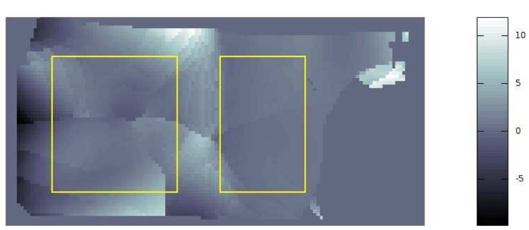

Fig. 3.8: Distribution of G'' for two layer phantom

Estimation of G''	center	right part
average	0.68kP	0.60kP
standard deviation	1.05kP	0.51kP

Table 3.4: Estimation of G''

Although we we could recover G'' to have a positive average value, comparing it with its standard deviation, the average value is smaller than its standard deviation. Looking more closer into the distribution of the recovered G'', the average value in the center part is a small positive value, but there are some negative value in that part. Further the average value in the right part is uniformly positive which means that this value is reliable. As far as we know, our result is quite good compare with the other recovered values of G'' by the direct method (cf.[12]) and modified integral method (cf.[3]). Nevertheless, we have to say that estimating the value of G'' is not easy because it is a small value compared with the value of G'.

3.3 Data of Mayo Clinic

We used the data by courtesy of Mayo Clinic. From the attached information, the view is 20cm square composed of 256 pixels each size. On the left side of the gel phantom, external vibration is continuously applied with 100Hz sinusoidal displacement. The sample have four cylindrical inclusions and their diameters are 5, 10, 16, and 25mm. The inclusions are stiffer than container. The original data have eight snapshots in 360 degrees phase shift. We altered the data into one complex-valued datum $u(x)$ by using a weighted average for input of our method.

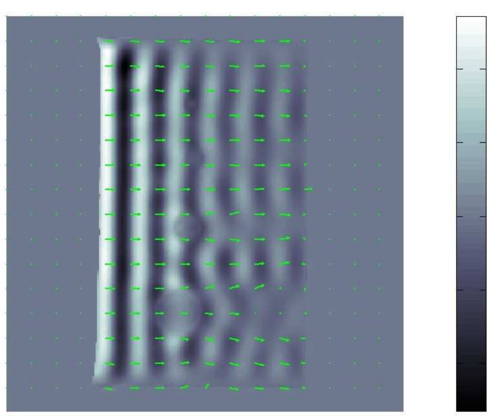

Fig.3.9: Fitting of β

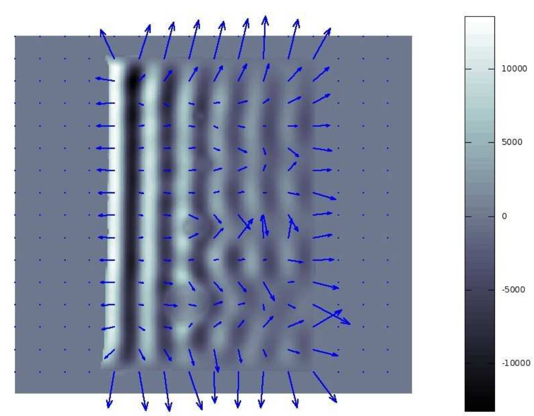

Fig.3.10: Fitting of α

The original data is less noisy compared to our previous data. It is very near to the plain parallel wave except at the parts of inclusions. The vectors β in Fig.3.9 are nearly constant throughout the image. They are perturbed slightly near at the inclusions and their backsides.

The vectors α change their directions more sensibly than those of vectors β. The lengths of the vectors α are represented ten times longer than those of the vectors β. Therefore, an average, the values of α are smaller than those of β.

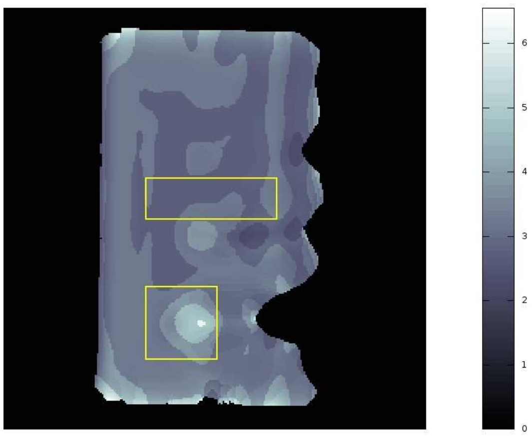

Fig. 3.11: Distribution of G' for Mayo Clinic data

Estimation of G'	bottom	center
average	3.71kP	2.92kP
standard deviation	0.62kP	0.24kP

Table 3.5: Estimation of G'

The Table.3.5 shows G' in the bottom square is bigger than center. This gives the information that the inclusions are stiffer than the container. The value of standard deviation in bottom square is bigger than the another, because it nearly encloses the inclusion. On the result for the bottom square, we average G' at a biggest inclusion and its neighborhood. If we take the area to be smaller, the estimated value of G' goes higher. We compared our result with the result obtained by the modified integral method. The result by that method is believed to be stable and numerically reliable. It also support our result of the average value because that output also has around 3.0kP in the region without inclusion part.

The recovered result of G' is not fully given. To be more specific, some part in the right hand side of the recovered result is intensionally cut off so that the result looks better. We have to explain why we did so. If β is zero or close to zero, we don't have any problem showing β as a vector. However in this case G' will become so large, because G' is proportional to $|\beta|^{-2}$ by Eq(3.1). This happens in the shadowed parts of the inclusions. In fact it is very difficult to see the nodal points of wave in these parts which could be coming from unsuccessful unwrapping of the MRE data, that is to specify the nodal value of wave in the MRE data. If there is not any nodal points in these parts, then the wave length there becomes infinitely long and hence the modulus of β will be very close to zero. This means that we could not trust the MRE data in these parts and this is why we cut off such parts.

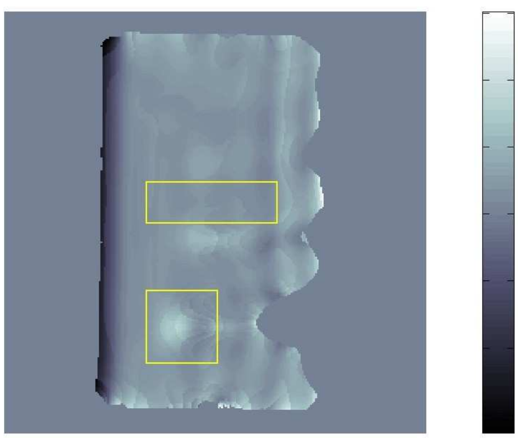

Fig.3.12: Distribution of G'' for Mayo Clinic data

Estimation of G''	bottom	center
average	0.58kP	0.34kP
standard deviation	0.36kP	0.08kP

Table 3.6: Estimation of G''

For the recovered values of G'', the ratio of G'' to G' fits to the ratio which is commonly believed, that is G'' is about one tenth order of G'.

Chapter 4

De-noising and Sharpening

In this chapter, we will show that by a simple modification, the LWV method can be applied as a de-noising for the MRE data. The principle behind this is as follows. For the local multiple wave form u with n local single-wave forms, we have already observed that $W(u;p,\sigma)(\xi)$ in most cases would have n well separated Gaussian peaks. This can be used to filter the MRE data which de-noises and sharpens the data.

4.1 LWV de-noising of MRE data

The next figure is the whole view of Fig.2.4 which will be de-noised.

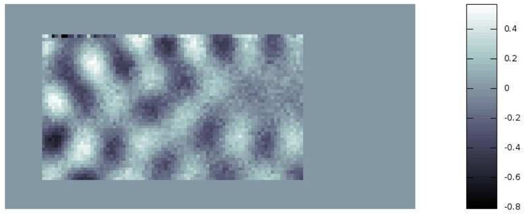

Fig.4.1: Real part in Spatial Domain

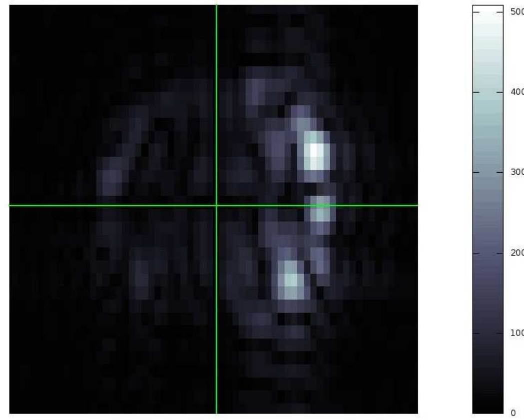

Fig.4.2: Modulus in Fourier Domain

The profile of waves is not so clear due to the noise. Our purpose here is to filter the data to reduce the noise and interferences of waves in the data shown by Fig.4.1.

The Fig.4.2 shows the distribution of the modulus of $W(u;p,\sigma)(\xi)$, From this we can know to which major directions the waves are propagating. Each Gaussian peak represents the major propagation direction for a certain group of waves. If these amplitudes of waves are large, then the peak becomes large also.

There are two ways to do the filtering. The one is to choose only the highest peak in the Fourier domain and remove the others. In detail, for each center point p of pixel, we replace $W(u;p,\sigma)(\xi)$ by

$$W(u;p,\sigma)(\xi)\delta(\xi-\xi(p)), \tag{4.1}$$

where $\delta(\xi-\xi(p))$ is the delta function with a singularity at $\xi(p) = \arg\max_{\xi} |W(u;p,\sigma)(\xi)|$, i.e. $\xi(p)$ gives the position of the peak of $|W(u;p,\sigma)(\xi)|$, and then takes the inverse Fourier transform of (4.1) which is multiplied by the characteristic function of the aforementioned pixel. This process is done for each pixel and we obtain filtered waves by superposition. As a result we have Fig.4.3. We can tune de-noising effect by replacing $\delta(\xi-\xi(p))$

by a Gaussian window centered at $\xi(p)$.

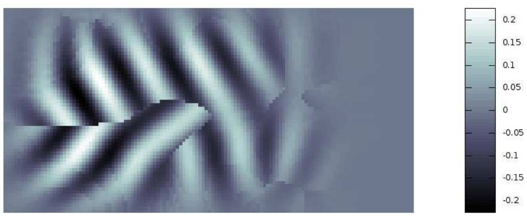

Fig.4.3: Real part in Spatial Domain

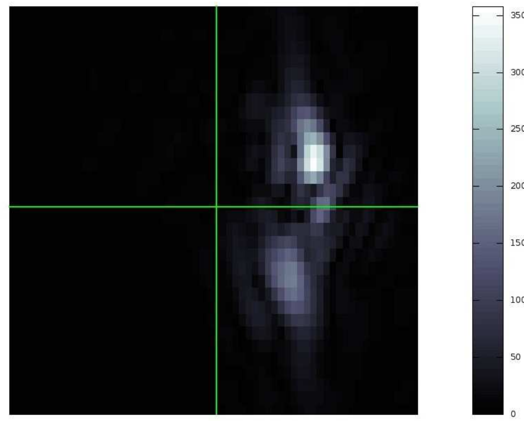

Fig.4.4: Modulus in Fourier Domain

Fig.4.4 gives the modulus of the Fourier transformation of Fig.4.3 and we can see that there are two peaks. Also, as can be seen in Fig.4.3, there will be a discontinuity where two waves correspond to these peaks. Comparing Fig.4.2 and Fig.4.4, we know that Fig.4.4 has much more clear and sharp images of waves.

Next, we will show another way of filtering. This is to filter the wave u around the globally strongest wave in the Fourier domain. That is let ξ_∞ be the peak of the modulus of the Fourier transform of u. Then this filtering is to filter u in the previous way just around ξ_∞ for each pixel. Then, we have the following figures for the filtered wave.

Fig.4.5: Real part in Spatial Domain

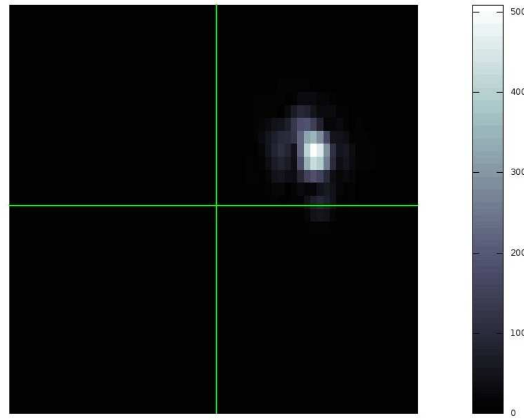

Fig.4.6: Modulus in Fourier Domain

There is only a single wave which is close to a simple sinusoidal wave (Fig.4.5) and single peak in the Fourier domain (Fig.4.6).

4.2 Testing with Mayo Clinic data

The LWV de-noising always makes any data smooth taking off segmentations as well as noise in the data. Hence, if the input data is nearly free from noise, then the de-noising process is unnecessary. For example, we applied the LWV de-noising to the Mayo Clinic data Fig.4.7.

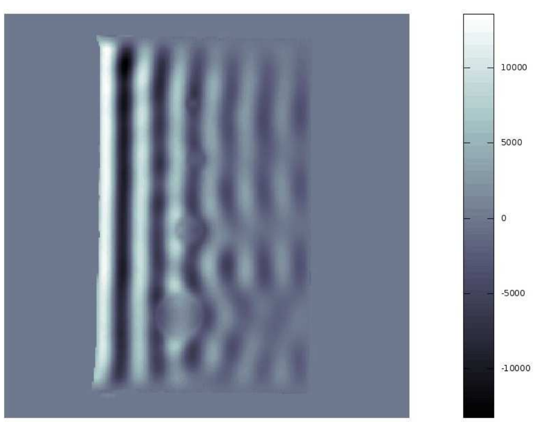

Fig.4.7: Data to be processed

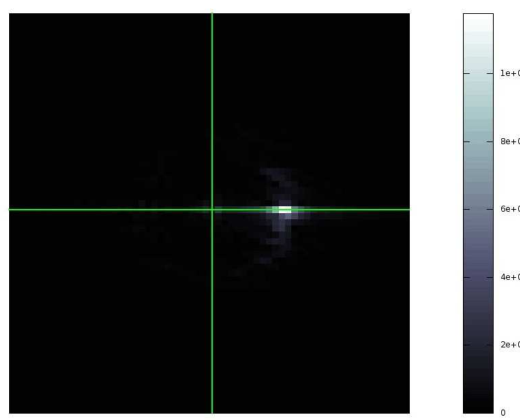

Fig.4.8: Modulus in Fourier domain

Then, we obtained the following two figures.

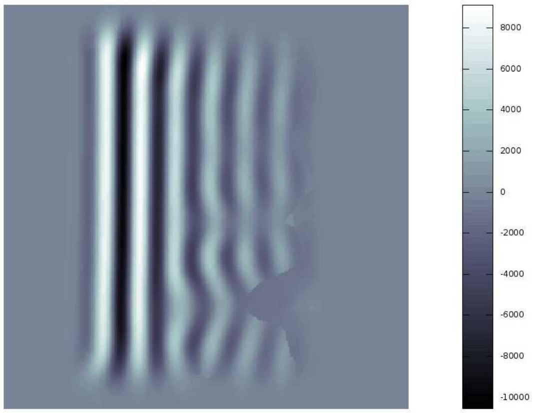

Fig. 4.9: Filtering by the locally strongest wave

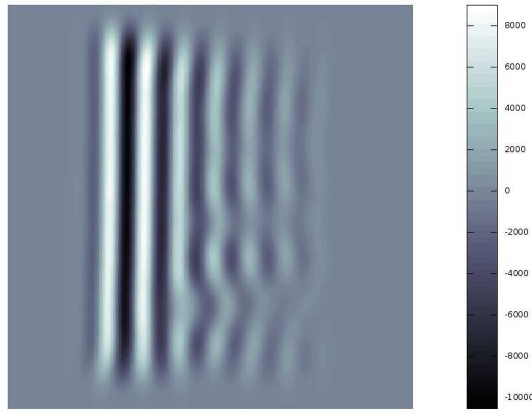

Fig.4.10: Filtering by the globally strongest wave

We can see that the de-noising made the boundary of inclusions smoother and masked the inclusions.

Chapter 5

Extension from two dimension to three dimension

We have studied this theory only two dimensional space $x, \xi \in \mathbb{R}^2$, however, this theory can be easily applied to three dimensional space because we did not use anything specific to two dimension. Therefore, we have no problem even if we change the dimension size to three.

In the programming, unfortunately, this change will run into trouble. As of today, three dimensional Fourier transformation is time consuming task for ordinary computer. We should plan our strategy that we expand to three dimensional case while computer calculation is limited to two dimension.

The following figures are outputs of two dimensional cross section spanned by two of three dimensional coordinates by a software ImageJ (http://rsb.info.nih.gov/ij/) The input data comes from simulated wave pattern. The figure shows the profiles of displacement values represented by gray scale.

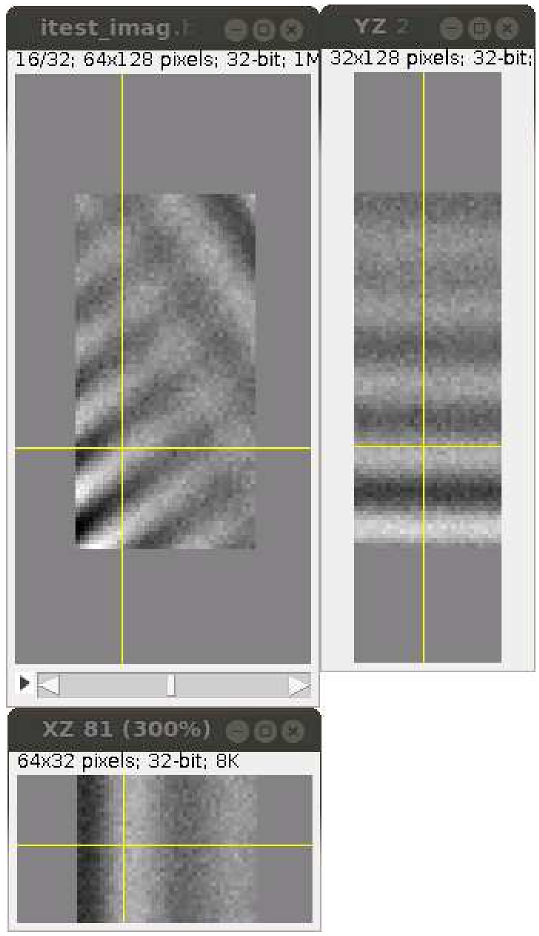

Fig.5.1: Top left figure is a profile of x-y plane. bottom left is of x-z. top right is of y-z.

We will be able to show each β in two dimensional space is just a projection from real β exists in three dimensional space. For example, x component of β in x-y plane is same as x component of β in x-z plane. Therefore, we get three dimensional β direction sufficiently, if we have all three of two dimensional projected β redundantly.

This modification keeps calculation cost increase only three times point-wise. Compared to calculation cost increase is at least $n \log n$ (n is size of length of an edge of three dimensional data) times bigger if we simply use three dimensional Fast Fourier Transformation, the difference is overwhelming, and the advantage of this modification is clear.

As of today, all the three dimensional data from real MRE system are essentially stacks of very similar two dimensional data. In addition, the three dimensional 'boxel' is not regular hexahedron but rectangular cuboid. In these circumstance, we cannot evaluate the performance of this modification simply because we do not have the data that is intrinsically three dimensional.

Chapter 6

Source code

Last but not least, this is very important part of this doctor thesis. Here attached is the source code of the version as of January 2013, Here shows only essential parts, not full set of the source code. It has tested and is running fine as expected, but it is still under development for improvement. It is subject to bug fix, major and minor change. Contact me by way of e-mail y4kw@math.sci.hokudai.ac.jp for latest full version.

6.1 Software Implementation for LWV and LAV method

These codes are developed using two scientific software GNU Octave and Matlab. For our convenience, we have used only common functions to both Octave and Matlab.

This program is an implementation of LWV method and LAV method. Only the important and essential part is listed here. Lines before 209 is skipped for listing. They are just for input data reading. Lines after 423 is skipped, too. They are just for output data writing.

We have got good performances of this program as we have shown in previous sections. We are still fixing this program to get best performance for newly available and forthcoming data.

Some other methods have a limitation of data shape or are not able to analyze near boundary. No such limitations are imposed to this method. This method requires two displacement data that differs 90 degrees apart.

As we explained, this method is so tolerant to noise that we don't need previous noise reduction.

We have a smart option that automatically set appropriate Gaussian size. We can also manually set that parameter arbitrarily in such cases as we want global average, or we want to detect small inclusion.

The results tend to go smoother. Saying positively, the results tend to go stabler. Saying negatively, we have a risk such as missing a small abnormality.

Code 6.1: ldone.m

```
invsquare=4*pi*pi*mechf^2*resolution^2;


%
%
%
%
%         #       #     # #     #         #          #    #     #
%         #       #  #  # #     #         #         # #   #     #
%         #       #  #  # #     #         #        #   #  #     #
%         #       #  #  # #     #         #       #     # #     #
%         #       #  #  #  #   #          #       #######  #   #
%         #       #  #  #   # #           #       #     #   # #
%         #######  ## ##     #            ####### #     #    #
%
%
%


%f is complex-valued input. It is spatial domain wave image.
f = img - i * imgphd;

%finding peak in Fourier domain modulus
globalG = abs(fftshift(fft2(f)));
[iy ix]=find(globalG==max(globalG(:)),1);

%distance from zero in Fourier domain
globalAve=2*pi*sqrt((iy-Nm/2-1)/Nm*(iy-Nm/2-1)/Nm+(ix-Nl/2-1)/Nl*(ix-Nl/2-1)/Nl);

% s is size of gaussian used in localization of a wave, it is used in gaussweight
% case 1: automatic tuning of size, zero division error if globalAve is zero.
s=2*pi/globalAve;
% case 2: you can assign arbitary value
%s=64;bad too big

% t is size of gaussian used in weighting in Fourier domain, it is used gaussweightxi
% case 1: rough automatic tuning of size
%t=2/sqrt(Nm*Nl); is good for noisy sim data, but t=4/sqrt is not stable
t=2/sqrt(Nm*Nl);
% case 2: you can assign arbitary value
%t=1/128;

[l m]=meshgrid(-Nl+1:Nl-1,-Nm+1:Nm-1);
gaussweight=exp(-(l.^2+m.^2)/(2*(s^2)));
gaussweightxi=exp(-((m/Nm).^2+(l/Nl).^2)/(2*(t^2)));

% G, G(.*) P are used in finding G'
G=zeros(Nm,Nl);
```

```
Gs=zeros(Nm,Nl);
Gsx=zeros(Nm,Nl);
Gsxr=zeros(Nm,Nl);
Gsy=zeros(Nm,Nl);
Gsyr=zeros(Nm,Nl);
P=zeros(Nm,Nl);

% V, V(.*) are used in finding G''
V=zeros(Nm,Nl);
Vs=zeros(Nm,Nl);
Vsx=zeros(Nm,Nl);
Vsxr=zeros(Nm,Nl);
Vsy=zeros(Nm,Nl);
Vsyr=zeros(Nm,Nl);

% to distinguish data value 0 and outside data value zero
% 0 0 0 1 -2 3 5 2 0 0 0
% 0 0 0 1 -2 0 4 1 0 0 0
% 0 0 0 0 -1 3 2 1 0 0 0
% such a case zero in center happen to be accidentary zero, then must be vaild data.
% if square matrix size is 2 * neighb + 1 is zero matrix, it must be outside,
% skipping calculation outside saves time. neighb is parameter works for time saving.
% it does not change results in ROI, it affects outside of ROI only.
% neighb=1 is the smallest and fastest.
neighb=1;

% main loop
for mm=1:Nm
    fprintf('.');
    if OctaveMode
        fflush(stdout);
    end
    for ll=1:Nl
        if (f(max(1,mm-neighb):min(Nm,mm+neighb),max(1,ll-neighb):min(Nl,ll+neighb)) == ...
                zeros(1-(max(1,mm-neighb)-min(Nm,mm+neighb)),1-(max(1,ll-neighb)-min(Nl,ll+neighb))))
                Gsx(mm,ll)=0;
                Gsy(mm,ll)=0;
                Gsxr(mm,ll)=0;
                Gsyr(mm,ll)=0;
                %%for fake estimation formula
                %Gs(mm,ll)=globalAve;
                %%for exact formula
                %% background is zero
                Gs(mm,ll)=0;
                %% background is near average.
                %Gs(mm,ll)=1/(globalAve^2);
        else
                fsgauss = gaussweight(Nm-mm+1:2*Nm-mm,Nl-ll+1:2*Nl-ll);
                ff=f.*fsgauss;
                ffs=circshift(ff,[-mm+1,-ll+1]);
                G = abs(fftshift(fft2(ffs)));
                [iy ix]=find(G==max(G(:)),1);
                Gsx(mm,ll)=2*pi*(ix-Nl/2-1)/Nl;
                Gsy(mm,ll)=2*pi*(iy-Nm/2-1)/Nm;

                % start of least aquare fitting
                V = fftshift(fft2(ffs));

                %%DEBUG: DO NOT USE %Tt = V ./ G; USE Tt = exp(i*angle(V));
                %%DEBUG: It looks same but you should better avoid division small by small.
                %%DEBUG: If you get NaN, the reason may be division small by small.
                Tt = exp(i*angle(V));
                %Tt = V ./ G;
                Dt = imag(Tt/Tt(iy,ix));
                %if mm == Nm/2 && ll == Nl/2
                %stopsubplot(2,2,1);
                %imagesc(Dt); axis('equal'); colorbar;
                %set(gca,'YDir','normal');
                %end
                %Dt = imag(Tt./Tt(iy,ix));
                %Dt = l(Nm-iy+1:2*Nm-iy,Nl-ix+1:2*Nl-ix);
                %Use imag insted of arg like :Dt = imag(Tt./Tt(iy,ix));
                %Dt = arg(Tt./Tt(iy,ix));
                Dw = gaussweightxi(Nm-iy+1:2*Nm-iy,Nl-ix+1:2*Nl-ix);
                Dl = l(Nm-iy+1:2*Nm-iy,Nl-ix+1:2*Nl-ix);
```

```
                Dm = m(Nm-iy+1:2*Nm-iy,Nl-ix+1:2*Nl-ix);
                Wll = Dw .* Dl .* Dl;
                Wlm = Dw .* Dl .* Dm;
                Wmm = Dw .* Dm .* Dm;
                Wlt = Dw .* Dl .* Dt;
                Wmt = Dw .* Dm .* Dt;

                swll = sum(sum(Wll));
                swlm = sum(sum(Wlm));
                swmm = sum(sum(Wmm));
                swlt = sum(sum(Wlt));
                swmt = sum(sum(Wmt));
                A = [swll, swlm; swlm, swmm];
                b = [swlt, swmt]';
                %Ainvb = pinv(A)*b;
                Ainvb = A\b;
                %Vsx(mm,ll)=Ainvb(1);
                %Vsy(mm,ll)=Ainvb(2);
                %USE THIS
                Vsx(mm,ll)=Ainvb(1)*Nl/(s*s*pi);
                Vsy(mm,ll)=Ainvb(2)*Nm/(s*s*pi);
                %%%
                %% DEBUG ONLY; use exact formula below.
                %% fake simple approximate estimation formula
                %Gs(mm,ll)=(Gsx(mm,ll)^2+Gsy(mm,ll)^2);
                %P(mm,ll)=sqrt(Gsx(mm,ll)^2+Gsy(mm,ll)^2);
                %P(mm,ll)=Gsx(mm,ll)*Vsx(mm,ll)+Gsy(mm,ll)*Vsy(mm,ll);

                %% exact formula

                %IMPORTANT: if Gsx and Gsy are both small, then Gs is near to infinity,
                %IMPORTANT: output result may be nonsense value.
                % rangelim tunes this parameter, rangelim=2 is recommended.
                % very big postive value is equivalent to show as of formula.
                % negative value is also equivalent to show as of formula.
                rangelim=2;

            if ((Gsx(mm,ll)^2+Gsy(mm,ll)^2)-(Vsx(mm,ll)^2+Vsy(mm,ll)^2)>globalAve^2/rangelim)
                    %G' estimation
                    Gs(mm,ll)=invsquare* ...
                    ((Gsx(mm,ll)^2+Gsy(mm,ll)^2)-(Vsx(mm,ll)^2+Vsy(mm,ll)^2)) ...
                    /(((Gsx(mm,ll)^2+Gsy(mm,ll)^2)-(Vsx(mm,ll)^2+Vsy(mm,ll)^2))^2  ...
                    + 4*(Gsx(mm,ll)*Vsx(mm,ll)+Gsy(mm,ll)*Vsy(mm,ll))^2);
                    %
                    %G'' estimation
                    P(mm,ll)=invsquare* ...
                    2*(Gsx(mm,ll)*Vsx(mm,ll)+Gsy(mm,ll)*Vsy(mm,ll)) ...
                    /(((Gsx(mm,ll)^2+Gsy(mm,ll)^2)-(Vsx(mm,ll)^2+Vsy(mm,ll)^2))^2  ...
                    + 4*(Gsx(mm,ll)*Vsx(mm,ll)+Gsy(mm,ll)*Vsy(mm,ll))^2);
            else
                    %% when both Gsx and Gsy are small, inverse is too big.
                    Gs(mm,ll)=0;

                    %this shows all out of bound data in higher level not zero.
                    %Gs(mm,ll)=invsquare/((globalAve^2)/rangelim);
            end
        end
    end
end
fprintf('\n');


%Gsxr=1*Gsx./(Gsx.^2+Gsy.^2);
%Gsyr=1*Gsy./(Gsx.^2+Gsy.^2);
Gsxr=1*Gsx;
Gsyr=1*Gsy;

Vsxr=1*Vsx;
Vsyr=1*Vsy;

[lsl,msm]=meshgrid(1:16:Nl,Nm:-16:1);
Gsxrr=Gsxr(Nm:-16:1,1:16:Nl);
Gsyrr=Gsyr(Nm:-16:1,1:16:Nl);
Vsxrr=Vsxr(Nm:-16:1,1:16:Nl);
Vsyrr=Vsyr(Nm:-16:1,1:16:Nl);
```

```
Ps=P(Nm:-16:1,1:16:Nl);
%[lsl,msm]=meshgrid(1:16:Nl,1:16:Nm);
%Gsxrr=Gsxr(1:16:Nm,1:16:Nl);
%Gsyrr=Gsyr(1:16:Nm,1:16:Nl);
%Vsxrr=Vsxr(1:16:Nm,1:16:Nl);
%Vsyrr=Vsyr(1:16:Nm,1:16:Nl);
%Ps=P(1:16:Nm,1:16:Nl);
%%[lsl,msm]=meshgrid(Nl:-16:1,1:16:Nm);
%%Gsxrr=Gsxr(1:16:Nm,Nl:-16:1);
%%Gsyrr=Gsyr(1:16:Nm,Nl:-16:1);
%%Vsxrr=Vsxr(1:16:Nm,Nl:-16:1);
%%Vsyrr=Vsyr(1:16:Nm,Nl:-16:1);
%%Ps=P(1:16:Nm,Nl:-16:1);
%%%%[lsl,msm]=meshgrid(Nl:-8:1,1:8:Nm);
%%%%Gsxrr=Gsxr(1:8:Nm,Nl:-8:1);
%%%%Gsyrr=Gsyr(1:8:Nm,Nl:-8:1);
%%%%Vsxrr=Vsxr(1:8:Nm,Nl:-8:1);
```

6.2 De-noising processor as application of LWV method

We take advantage of a strong noise tolerance of LWV method, it also works for de-noising. The input and output data formats are common, therefore this de-noising processor can be inserted to another MRE diagnosing software as de-noising process.

Lines before 228 is skipped for listing. They are just for input data reading. Lines after 339 is skipped, too. They are just for output data writing.

Code 6.2: lg.m

```
228 f = img - i * imgphd;
229 globalG = abs(fftshift(fft2(f)));
230 [iy ix]=find(globalG==max(globalG(:)),1);
231 globalAve=sqrt((iy-Nm/2)/Nm*(iy-Nm/2)/Nm+(ix-Nl/2)/Nl*(ix-Nl/2)/Nl);
232
233 [l m]=meshgrid(-Nl+1:Nl-1,-Nm+1:Nm-1);
234 counter=0;
235 %s=8 is good for 256pixels;
236 %s=ceil(max(Nl,Nm)/32) is good for normal type;
237 s=ceil(max(Nl,Nm)/32);
238 %s=ceil(max(Nl,Nm)/8) is good for extrapolation;
239 %s=ceil(max(Nl,Nm)/8);
240 %t=0.00000000000000016/sqrt(Nm*Nl);
241 %t=16/sqrt(Nm*Nl);
242 gaussweight=exp(-(l.^2+m.^2)/(2*s.^2));
243 %gaussweightxi=exp(-((m/Nm).^2+(l/Nl).^2)/t.^2);
244 gaussweightxi=0*l;
245 gaussweightxi(Nm,Nl)=1;
246
247 G=zeros(Nm,Nl);
248 Gs=zeros(Nm,Nl);
249 Gsx=zeros(Nm,Nl);
250 Gsxr=zeros(Nm,Nl);
251 Gsy=zeros(Nm,Nl);
252 Gsyr=zeros(Nm,Nl);
253 U=zeros(Nm,Nl);
254 V=zeros(Nm,Nl);
255 Vs=zeros(Nm,Nl);
256 Vsx=zeros(Nm,Nl);
257 Vsxr=zeros(Nm,Nl);
```

```
Vsy=zeros(Nm,Nl);
Vsyr=zeros(Nm,Nl);
P=zeros(Nm,Nl);
R=zeros(Nm,Nl);
RR=zeros(Nm,Nl,3);
%RF=zeros(Nm,Nl,3);
RF=zeros(max(Nm,Nl),max(Nm,Nl),3);

RR(:,:,1)=f;
fsquare=f;
fsquare(max(size(fsquare)),max(size(fsquare)))=0;
RF(:,:,1)=abs(fftshift(fft2(fsquare)));
for nn=2:3
    f = img - i * imgphd;
    %RF(:,:,nn)=abs(fftshift(fft2(f)));
    for mm=1:Nm
        fprintf('.');
        if OctaveMode
            fflush(stdout);
        end
        for ll=1:Nl
            if (f(max(1,mm-neighb):min(Nm,mm+neighb),max(1,ll-neighb):min(Nl,ll+neighb)) == ...
                zeros(1-(max(1,mm-neighb)-min(Nm,mm+neighb)),1-(max(1,ll-neighb)-min(Nl,ll+neighb))))
                Gsx(mm,ll)=0;
                Gsy(mm,ll)=0;
                Gsxr(mm,ll)=0;
                Gsyr(mm,ll)=0;
                Gs(mm,ll)=globalAve;
            else
                fsgauss = gaussweight(Nm-mm+1:2*Nm-mm,Nl-ll+1:2*Nl-ll);
                ff=f.*fsgauss;
                %ffs(1:Nm-mm+1,1:Nl-ll+1)=ff(mm:Nm,ll:Nl);
                %ffs(mod(-mm+2,-Nm)+Nm:Nm,1:Nl-ll+1)=ff(1:mod(mm-1,-Nm)+Nm,ll:Nl);
                %ffs(1:Nm-mm+1,mod(-ll+2,-Nl)+Nl:Nl)=ff(mm:Nm,1:mod(ll-1,-Nl)+Nl);
                %ffs(mod(-mm+2,-Nm)+Nm:Nm,mod(-ll+2,-Nl)+Nl:Nl)=ff(1:mod(mm-1,-Nm)+Nm,1:mod(ll-1,-Nl)+Nl);
                %ffs=circshift(ff,[mm-1,ll-1]);
                ffs=circshift(ff,[-mm+1,-ll+1]);
                V = fftshift(fft2(ffs));
                G = abs(V);
                if nn==3
                    [iy ix]=find(G==max(G(:)),1);
                end
                Gsx(mm,ll)=(ix-Nl/2-1)/Nl;
                Gsy(mm,ll)=(iy-Nm/2-1)/Nm;
                Gs(mm,ll)=sqrt(Gsx(mm,ll)^2+Gsy(mm,ll)^2);

                %               Tt = V ./ G;
                %               Dt = imag(Tt./Tt(iy,ix));
                Dw = gaussweightxi(Nm-iy+1:2*Nm-iy,Nl-ix+1:2*Nl-ix);
                %               Dl = l(Nm-iy+1:2*Nm-iy,Nl-ix+1:2*Nl-ix);
                %               Dm = m(Nm-iy+1:2*Nm-iy,Nl-ix+1:2*Nl-ix);
                %               Wll = Dw .* Dl .* Dl;
                %               Wlm = Dw .* Dl .* Dm;
                %               Wmm = Dw .* Dm .* Dm;
                %               Wlt = Dw .* Dl .* Dt;
                %               Wmt = Dw .* Dm .* Dt;
                %               swll = sum(sum(Wll));
                %               swlm = sum(sum(Wlm));
                %               swmm = sum(sum(Wmm));
                %               swlt = sum(sum(Wlt));
                %               swmt = sum(sum(Wmt));
                %               A = [swll, swlm; swlm, swmm];
                %               b = [swlt, swmt]';
                %               Ainvb = A\b;
                %               Vsx(mm,ll)=Ainvb(1);
                %               Vsy(mm,ll)=Ainvb(2);
                %               Vs(mm,ll)=sqrt(Vsx(mm,ll)^2+Vsy(mm,ll)^2);
                %
                %               P(mm,ll)=Gsx(mm,ll)*Vsx(mm,ll)+Gsy(mm,ll)*Vsy(mm,ll);
                U=(Nl*Nm)/(2*pi*s*s)*ifft2(fftshift(V.*Dw));
                %U=(Nl*Nm)/(2*pi*s*sqrt(s))*ifft2(fftshift(V.*Dw));
                R(mm,ll)=U(1,1);
                RR(mm,ll,nn)=R(mm,ll);
            end
        end
```

```
    end
    f=R;
fsquare=f;
fsquare(max(size(fsquare)),max(size(fsquare)))=0;
    RF(:,:,nn)=abs(fftshift(fft2(fsquare)));
end
```

Chapter 7

Generalization of α and β to function

7.1 Definitions

The discussion we did so far, α and β was constant independent of x. Now we would like to generalize them as functions of x, where they have Taylor series expansion.

$$u(x) = c_0 e^{f(x)+ig(x)} \quad c_0 \in \mathbb{C},\ x \in \mathbb{R}^2 \tag{7.1}$$

Without losing generality, we can substitute c_0 as 1, then,

$$u(x) = e^{f(x)+ig(x)} \quad x \in \mathbb{R}^2 \tag{7.2}$$

$$f(x) = \alpha \cdot x + {}^t x A x + A_3(x,x,x) + \cdots \in \mathbb{R} \tag{7.3}$$

where A is a symmetric matrix, A_3 is third order symmetric tensor, and

$$g(x) = \beta \cdot x + {}^t x B x + B_3(x,x,x) + \cdots \in \mathbb{R} \tag{7.4}$$

where B is symmetric matrix, B_3 is third order symmetric tensor.

7.2 Recovering frequency function of position x

Now, we would like to state a useful performance of Fourier transformation. Considering a complex valued function $u(x) = e^{f(x)+ig(x)} \quad x \in \mathbb{R}^2$ in spatial domain is not straightforward, since given $u(x)$ solve $f(x)$ and $g(x)$ numerically, we encounter a difficult unknown parameter fittings. In addition, change of $f(x)$ and change of $g(x)$ are tangled with each other and they cannot be easily separated.

On the other hand, if we investigate the Fourier transformed function, to be precise, Gaussian localized $u(x)$ be Fourier transformed, $f(x)$ and $g(x)$ are mapped differently and independently.

In fact, $g(x)$ will appear in the modulus part only in Fourier domain. Similarly, $f(x)$ will appear in the argument part only in Fourier domain.

As of today, I believe the following equations are good approximation, though, they are conjectures we are working now. Mathematically, we still need to show how close in average, how bad in worst case coming from the approximation done.

Here we show the equations we are talking about.

$$\begin{aligned} \left| \mathscr{F}\left[e^{-x^2/2} e^{f(x)+ig(x)} \right] \right| &\approx \mathscr{F}\left[e^{-x^2/2} e^{ig(x)} \right] \\ &= \mathscr{F}\left[e^{-x^2/2} e^{i(\beta \cdot x + {}^t x B x + B_3(x,x,x) + \cdots)} \right] \end{aligned} \tag{7.5}$$

Applying inverse Fourier transformation for both side, we get,

$$e^{i(\beta \cdot x + {}^t x B x + B_3(x,x,x) + \cdots)} \approx \frac{\mathscr{F}^{-1} \left| \mathscr{F}\left[e^{-x^2/2} e^{f(x)+ig(x)} \right] \right|}{e^{-x^2/2}} \tag{7.6}$$

This operation is so called deconvolution and we ask for help a software to calculate. The deconvolution itself is not always stable operation mathematically, especially when the transformed function is not belong to L^1. When we talk about numerical calculation in computer, this is not a problem, because it is always deals finite number of points only. But we take into consideration if we use Fast Fourier Transformation, that deals only 2π periodic functions. Finally taking logarithm and divide by i, we conclude.

$$\beta \cdot x + {}^t x B x + B_3(x,x,x) + \cdots \approx \frac{1}{i} \log \frac{\mathscr{F}^{-1}\left|\mathscr{F}\left[e^{-x^2/2} e^{f(x)+ig(x)}\right]\right|}{e^{-x^2/2}} \tag{7.7}$$

Now, from the point of numerical computation, the LHS is just a calculation, and fitting β, B and so on are simple task as they are fitting into linear and quadratic function. In other words, we will get these parameters infinitely by increasing fitting order.

We have a good example data as shown below.

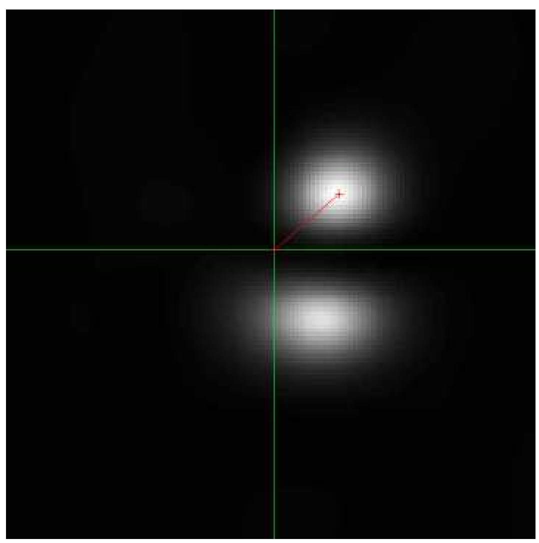

Fig.7.1: One of peaks is elliptic shaped on Fourier domain modulus

This Fig.7.1 is same as Fig.2.3. If B and higher order of B_n is zero, the shape of peak must be radially symmetric, which is same as Gaussian. If B and higher order of B_n is not zero, the shape of peak can be deformed from the shape of radially symmetric Gaussian. We observe the wave component directing bottom right has a frequency term changing.

7.3 Recovering attenuation function of position x

$$\begin{aligned} \arg \mathscr{F}\left[e^{-x^2/2}e^{f(x)+ig(x)}\right] &\approx \arg \mathscr{F}\left[e^{-x^2/2}e^{f(x)}\right] \\ &= \arg \mathscr{F}\left[e^{-x^2/2}e^{\alpha\cdot x+{}^txAx+A_3(x,x,x)+\cdots}\right] \end{aligned} \tag{7.8}$$

If f is function of ξ as

$$f(x) = \alpha \cdot x + {}^txAx + A_3(x,x,x) + \cdots \in \mathbb{R} \tag{7.9}$$

Up to this we stated just a conjecture, but let us go further. We will expect to have, if we will extend function class from $\mathscr{S}'$ to more rapidly increasing.

$$\alpha \approx \nabla_\xi \arg \mathscr{F}\left[e^{-x^2/2}e^{\alpha\cdot x+{}^txAx+A_3(x,x,x)+\cdots}\right]\Big|_{\xi=0} \tag{7.10}$$

$$A - \frac{1}{2}I \approx \nabla_\xi\nabla_\xi \arg \mathscr{F}\left[e^{-x^2/2}e^{\alpha\cdot x+{}^txAx+A_3(x,x,x)+\cdots}\right]\Big|_{\xi=0} \tag{7.11}$$

$$A_3 \approx \nabla_\xi\nabla_\xi\nabla_\xi \arg \mathscr{F}\left[e^{-x^2/2}e^{\alpha\cdot x+{}^txAx+A_3(x,x,x)+\cdots}\right]\Big|_{\xi=0} \tag{7.12}$$

Chapter 8

Conclusions

It is a mathematical rigorous scheme applicable to numerical inversion methods to recovery the shear modulus of the soft tissues of a living body. The results show the algorithm is robust for the simulated data and real output data from MRE data even the data contain relatively strong noise.

We developed a model independent data analysis for MRE data based on the FBI transformation to recover the local wave vector and local attenuation vector of the strongest local single wave assuming that waves in MRE data are locally given as a local multi-wave form. This can be also applied to other wave images. We also linked the recovered local wave vector and local attenuation vector to the storage modulus and loss modulus by using a nearly incompressible isotropic viscoelastic equation which describes the displacement vector of time harmonic waves propagating in a MRE phantom. The recoveries of β and G' were quite good and stable.

The results show the algorithm is robust for the simulated data and real output data from MRE data even the data contain relatively strong noise.

Further, we showed that a modified version of LWV method which enables to recover the local wave vector can be used to de-noise the MRE data.

Besides the two dimensional data, we applied the algorithms to the three dimensional data. Three dimensions are just theoretical attempts and more work should be processed in future.

Our MRE data analysis was conducted using a numerical computational software on linux-based ordinary desktop computer. The fast Fourier transformation is not so time consuming for maximum 256x256 pixels MRE data. The overall calculation finished in order of minutes.

Acknowledgments

In particular, I am grateful to my advisor, Prof. Gen Nakamura. I thank him for all the help from him not only for my study and research but also for his support and advice for my life. I thank the members in the Committee in Charge, Prof. Tonegawa. I thank Prof. Jiang Yu in Shanghai Economic University, China, who received Ph.D when I was a master student. It is him who helped me a lot, he always answered my questions very kindly. I thank Dr. Haibing Wang, who is now in China, he have recommended me to compare with Modified Integral Method by Prof. Nakamura and Prof. Jiang.

I thank all the members in MRE research group of Japan: Prof. Gen Nakamura (Hokkaido Univ.), Prof. Mikio Suga (Chiba Univ.), Prof. Jianping Gong (Hokkaido Univ.), Prof. Takao Yamamoto (Gunma Univ.), Prof. Kenichi Nagai (Gunma Univ.), Prof. Kenji Shirota (Ibaraki Unv.), Prof. Hiroshi Fujiwara (Kyoto Univ.), Prof. Kazuaki Nakane (Osaka Univ.), Dr. Nobuyuki Higashimori (Kyoto Univ.), Prof. Tadano (Hokkaido Univ.), Prof. Kajiwara (Hokkaido Univ.). The people and MRE data from Micro MRE system at Department of Engineering in Hokkaido university are the most valuable in writing this paper. Especially, I admire Prof. Tadano and his student Mr. Hayato Suzuki for their organizing MRE project and for introducing MRE system.

I hope that we can continue our collaboration to make a big achievement for MRE. Also, I thank Dr. Sei Nagayasu, who helped me a lot when I was a master course student.

Before start writing this paper, discussions on FBI transformation with

Prof. Hideki Takuwa (Doshisha Univ.) imspired me that a formulation on FBI transformation can be applied to MRE analysis.

Bibliography

[1] R. Muthupillai, D. Lomas, P. Rossman, J. Greenleaf, A. Manduca and R. Ehman, Magnetic resonance elastography by direct visualization of propagating acoustic strain waves, Science **269** (1995) 1854–1857.

[2] H. Ammari, P. Garapon, H. Kang and H. Lee , A method of biological tissues elasticity reconstruction using magnetic resonance elastography measurements, Quar. Appl. Math., **66**(2008) 139-175.

[3] Y. Jiang and G. Nakamura, Viscoelastic properties of soft tissues in a living body measured by MR elastography, J. Phys. Conf. Ser., **290** (2010), 012006.

[4] Higashimori N, 2007 Identification of viscoelastic properties by magnetic resonance elastography *J. Phys.: Conf. Ser.* **73** 012009 (6pp) online.

[5] Jiang Y, Fujiwara H, Nakamura G, 2011 Approximate Steady State Models for Magnetic Resonance Elastography *SIAM J. Appl. Math.* **71** 1965–1989.

[6] Nakamura G, Jiang Y, Nagayasu S and Cheng J, 2008 Inversion analysis for magnetic resonance elastography *Applicable Analysis* **87** 165–179.

[7] Y. Jiang, Mathematical Data Analysis for Magnetic Resonance Elastography, Ph.D. thesis, Department of Mathematics, Graduate School of Science, Hokkaido University, Japan, 2009.

[8] Jean-Marc Delort, F.B.I. Transformation Second Microlocalization and Semilinear Caustics, Springer-Verlag

[9] P. S. Addison, The illustrated wavelet transform handbook : introductory theory and applications in science, engineering, medicine and finance, Bristol : Institute of Physics Pub., 2002.

[10] M.Suga, T.Matsuda, K.Minato O.Oshiro, K.Chihara, J.Okamoto, O.Takizawa, M.Komori and T.Takahashi, Measurement of in vivo Local Shear Modulus Using MR Elastography Multiple Phase Patchwork Offsets, IEEE Transactions on Biomedical Engineering, 50(7), (2003), 908-915.

[11] R.M.Christensen, Theory of viscoelasticity, Academic Press, (1982).

[12] Oliphant T, (2001) *Direct methods for dynamic elastography reconstruction: optimal inversion of the interior Helmholtz problem* Thesis (Ph.D.) – Biomedical Sciences – Biomedical Imaging – Mayo Graduate School.

Appendix: FBI transformation of waves

Let $u(x)$ be of a single wave form around a point p given by

$$u(x) = e^{\omega(\alpha+i\beta)\cdot(x-p)} \tag{A.1}$$

with $\alpha, \beta \in \mathbb{R}^2$. Then we compute its FBI transform as follows. By taking $x-p$ as a new variable for the integration, we have

$$\begin{aligned} W(u;p,\sigma)(\xi) &= \int_{\mathbb{R}^2} e^{\omega(\alpha+i\beta)\cdot(x-p)} e^{-\frac{|x-p|^2}{2\sigma^2}} e^{-ix\cdot\xi}\, dx \\ &= e^{-ip\cdot\xi} \int_{\mathbb{R}^2} e^{\omega(\alpha+i\beta)\cdot x} e^{-\frac{|x|^2}{2\sigma^2}} e^{-ix\cdot\xi}\, dx. \end{aligned} \tag{A.2}$$

Further, by $\omega\alpha\cdot x - \frac{|x|^2}{2\sigma^2} = \frac{\omega^2\sigma^2|\alpha|^2}{2} - \frac{|x-\omega\sigma^2\alpha|^2}{2\sigma^2}$ and taking $x-\omega\sigma^2\alpha$ as a new variable in the integration, we have

$$\int_{\mathbb{R}^2} e^{\omega(\alpha+i\beta)\cdot x} e^{-\frac{|x|^2}{2\sigma^2}} e^{-ix\cdot\xi}\, dx = e^{\frac{\omega^2\sigma^2|\alpha|^2}{2}} e^{i\omega^2\sigma^2\alpha\cdot\beta} e^{-i\omega\sigma^2\alpha\cdot\xi} \int_{\mathbb{R}^2} e^{i\omega\beta\cdot x} e^{-\frac{|x|^2}{2\sigma^2}} e^{-ix\cdot\xi}\, dx. \tag{A.3}$$

Note that the integration in (A.3) is the Fourier transform of the Gaussian with respect to $\xi-\beta$. Hence,

$$\int_{\mathbb{R}^2} e^{i\omega\beta\cdot x} e^{-\frac{|x|^2}{2\sigma^2}} e^{-ix\cdot\xi}\, dx = 2\pi\sigma^2 e^{-\frac{\sigma^2|\xi-\omega\beta|^2}{2}}. \tag{A.4}$$

After all, we have obtained

$$
\begin{aligned}
&W(u;p,\sigma)(\xi)\\
&\qquad = 2\pi\sigma^2 \exp\left[i\omega^2\sigma^2\alpha\cdot\beta + \frac{\omega^2\sigma^2|\alpha|^2}{2}\right]\exp\left[-i(p+\omega\sigma^2\alpha)\cdot\xi - \frac{\sigma^2|\xi-\omega\beta|^2}{2}\right].
\end{aligned}
\tag{A.5}
$$