

---

# QUANTIFICATION OF LIVER FAT USING MAGNETIC RESONANCE IMAGING

---

By  
Rewa Sood  
Department of Electrical and Computer Engineering  
University of Illinois at Urbana-Champaign

MAY 9, 2017

## Abstract

Currently, the gold standard for quantifying the fraction of the liver that is composed of fat is liver biopsy. A liver biopsy is a painful and uncomfortable procedure where the patient's liver is sampled using a large needle. In addition, the liver biopsy only samples a small area of the liver, which may not accurately represent the fat content in the entire liver. Magnetic resonance imaging (MRI) provides an alternative, non-invasive, technique for the quantification of liver fat that is faster, more representative, and more comfortable for the patient. The goal of my project is to create a MATLAB program that reads in DICOM MRI images corresponding to each echo and produces a proton density fat fraction (PDFF) map of the liver. My initial solution to this problem was to create my own function in MATLAB that would provide the necessary map. However, my method would not converge. My second approach involved using `lsqcurvefit`, a built-in MATLAB function, which provided nonlinear curve-fitting functionality and allowed me to construct accurate PDFF maps. My code creates a GUI that shows the percent fat in different parts of the liver provided six echo images and their associated echo times. As a result of my work, a researcher would be able to generate the PDFF map to extract fat quantification values for acquired liver images.

Subject Keywords: Magnetic Resonance Imaging, Liver Fat Quantification

**Contents:**

1. Introduction .....	1
2. Literature Review.....	4
3. Methods.....	7
4. Results.....	12
5. Conclusion and Next Steps.....	15
References.....	17

## 1. Introduction

The goal achieved by my thesis research was to create a tool that would allow other researchers to generate a proton density fat fraction (PDFF) map of the liver given in-phase and out-of-phase magnetic resonance imaging (MRI) images. Before beginning work on the project, I first built my knowledge base regarding the basic concepts of MRI and then researched other work in this area. I then attempted my own solution to the problem using two methods. My first method involved using the Gauss-Newton algorithm; however, it did not converge. My second method used MATLAB's built-in curve-fitting function, `lsqcurvefit`. This method did provide a solution, and after applying certain optimizations, it computed the PDFF map in under one minute. Lastly, I created an executable application so that other researchers looking to create PDFF maps would be able to obtain the map by simply providing the images and associated echo times. For the remainder of the introduction, I will discuss the background MRI concepts needed in order to conduct this research. In the following section, I will discuss the other work conducted in this area of research via a literature review. In the Methods section, I will step through the process of my own research and I will present the results in the succeeding section. Finally, I will provide a discussion of next steps and a few concluding remarks in the final section.

MRI uses powerful magnetic fields to take advantage of the natural nuclear magnetic resonant (NMR) properties of protons within the body. MRI systems use three magnetic field types: the field of the main magnet (on the order of 1.5T), the magnetic fields created by a radiofrequency (RF) coil (on the order of micro-Tesla), and the magnetic fields created by gradient coils (on the order of milli-Tesla). Naturally, protons in the water and fat are all aligned along random directions due to random motion caused by thermal energy. These directions cancel out and the protons cannot emit any MRI signal. The main magnetic field provides a uniform static field which acts on the protons such that they are more aligned in one direction, creating a spin imbalance. Although the protons as a whole have some alignment, the system still cannot produce an MRI signal. The RF coil generates a magnetic field at a very specific frequency, the resonance frequency of the protons at the magnetic field strength of the main magnet, which excites a particular spin system for imaging. For our application, the signal results from the excitation of the system that includes both water and fat. Since protons are components

of water and fat, the resulting images reflect the proton density within the liver, and thus allow us to construct the PDFF map.

There are two main methods of image acquisition in MRI: spin-echo and gradient-echo imaging. Spin-echo imaging uses RF pulses to realign the phases of the nuclear spins to create echoes of the original signal, which are easier to digitize than the original signal. Gradient-echo imaging also realigns the phases of the proton system, but does so with the magnetic field created by the gradient coil instead of using RF pulses. In so doing, image acquisition by the gradient-echo imaging method is faster than the traditional spin-echo method. The images of the liver used for my thesis work were all obtained via gradient-echo imaging in a clinical MRI scanner at 1.5T.

The protons in the fat and water components of the body are excited at different frequencies. Therefore, at certain times, the molecules are in phase and at others they are out of phase. The six images that form the input to the algorithm are obtained at six different echo times, with three of the images in phase and the other three out of phase. These in-phase and out-of-phase images, along with their associated echo times, form the basis by which the optimized water and fat images are used to create the PDFF map. Besides optimizing water and fat images, the algorithm also estimates T2-star ( $T2^*$ ).  $T2^*$  is another nuclear magnetic resonance relaxation parameter which can influence the appearance of MRI images and which can change the calculation of the PDFF, if not accounted for. For PDFF estimation,  $T2^*$  can be calculated separately and on a pixel-by-pixel basis in order to remove any dependency of the PDFF on  $T2^*$ .  $T2^*$  can also be calculated for whole regions of an image at a time by executing the algorithm for a predefined region instead of just one pixel. The PDFF should only be a measure of proton density and is assumed to not have any influence from other MRI relaxation parameters such as  $T2^*$ . The calculated  $T2^*$  value depends on the number of different frequencies present in the signal. In this case, there are seven different frequencies present in the signal, six from fat and one from water. The initial assumption is that the  $T2^*$  value is uniform across the image. However, since the magnetic fields are not necessarily homogeneous and since the RF field excites multiple frequencies, the  $T2^*$  value is not homogeneous over the entire image [1].

This section has covered the basic MRI concepts that play a significant role in this project, including the spin systems that are excited by the magnetic fields, gradient echo

imaging, in-phase and out-of-phase images, and T2\* relaxation. The next section will discuss the previous work conducted regarding calculating the fat fraction in the liver.

## 2. Literature Review

The first papers I read described the motivation for calculating the fat fraction in the liver. Fat accumulation in the liver is a characteristic sign of liver disease, possibly due to alcohol abuse or an unhealthy diet. Quantifying liver fat is also an important part of understanding the causes of non-alcoholic fatty liver disease (NAFLD) and the effects of this disease on the body. Studying this disease is especially important because it can lead to cirrhosis and certain types of liver cancer. Studies have shown that the amount of fat contained in the liver can be reduced through treatment. Additionally, the type of treatment that a certain patient needs depends on the amount of fat they have in their liver. Thus, the quantification of fat in the liver will help researchers understand the severity of the condition and ascertain whether a certain treatment is improving the condition [2,3].

The traditional method for quantifying liver fat is via liver biopsy. Liver biopsies allow the researcher to analyze the amount of fat within the liver cells along with other cellular information such as inflammation and injury. There are two main downsides to liver biopsies. First, liver biopsies are very painful procedures that involve long needles. Second, liver biopsies give information about only a small portion of the liver. The needle provides a sample of the liver which provides information about the liver only in that specific area. If the doctor needed to test another part of the liver, they would have to collect another sample, which would cause the patient more pain. Quantifying liver fat through MRI improves upon this method by removing the need for biopsies altogether and by providing information about the whole liver instead of just a sample [2,4].

One of the major ways other researchers are quantifying liver fat is through magnetic resonance spectroscopy (MRS), which is used to calculate signal fat-fraction as opposed to the PDFF. Signal fat-fraction is the portion of the signal due to fat while PDFF is the portion of mobile protons in the liver attributable to fat. Since the resonance frequencies of water and fat are distinct, the signal fat-fraction can be calculated by dividing the signal intensity of the fat peaks by the total signal intensity. In general, fat has six distinct NMR peaks, visible at clinical magnetic field strengths, each of which represents a different fat moiety. Some studies only include the main fat peak, which leads to simpler calculations. However, this creates error because it incorrectly counts some of the signal from certain minor fat moieties as part of the

water signal. Using all six fat peaks leads to a more accurate quantification of the fat fraction because all six peaks are in reality contributing to the MRI signal from fat [5,6].

The first water-fat separation calculation technique I explored used the Gauss-Newton nonlinear curve-fitting algorithm. The paper I referenced for this method used complex input images as the basis for their algorithm, although the algorithm could also be used with magnitude images. The output of this algorithm was two separate water and fat images that could be used to create the PDFF map. The Gauss-Newton algorithm is an iterative process that updates the initial guesses for the water and fat images using the following signal equation.

$$S(TE) = \sqrt{(W^2 + F^2 |\sum_{p=1}^6 r_p e^{j2\pi\Delta f_p TE}|^2 + 2 * RE\{WF \sum_{p=1}^6 r_p e^{j2\pi\Delta f_p TE}\}) * e^{-\frac{TE}{T2^*}} \quad (1)$$

$S$  refers to the original image and  $W$  and  $F$  refer to the water and fat images, respectively.  $r_p$  and  $\Delta f_p$  are the weights and relative frequencies of each of the six fat peaks. The delta in front of the fat peak frequencies highlights the fact that they are reported relative to the frequency of the water peak. The frequencies and weights of the fat peaks are shown in Table 1.

**Table 1: Fat Peak Weights and Frequencies**

Fat Peak	Chemical Shift Freq. ( $\Delta f_p$ )(Hz)	Relative Amplitude ( $r_p$ )
1	+39	0.047
2	-32	0.039
3	-125	0.006
4	-166	0.120
5	-217	0.700
6	-243	0.088

Each iteration produces error terms,  $\Delta W$  and  $\Delta F$ , that are added to the initial  $W$  and  $F$  estimates to create the new input into the next iteration. The updates are calculated using a matrix that contains the partial derivatives of the original image  $S$  with respect to  $W$ ,  $F$ , and  $T2^*$ . The pseudoinverse of this matrix, multiplied by the difference between the original image and the estimated image results in the error terms. Unfortunately, when I applied my version of this algorithm, it did not converge. One possible source of error could be the method I used to adapt the Gauss-Newton algorithm to magnitude images instead of complex images. In my algorithm,  $S$  is a magnitude image, which explains the square root in the equation. Additionally, my matrix

of partial derivatives only included real partial derivatives while the original algorithm used real and imaginary partial derivatives [7].

Overall, the literature survey provided the background motivation for the research in this area, described the traditional methods of liver fat quantification, and delineated multiple methods of water-fat separation. Although the first method of water-fat separation I explored, the Gauss-Newton algorithm, did not converge, my work in this area provided me with a good understanding of the problem at hand. The following section describes the final method I used that provided separate water and fat images and the PDFFF map.

### 3. Methods

After working with the Gauss-Newton algorithm, I explored MATLAB's built in nonlinear curve-fitting algorithm, `lsqcurvefit`. In order to use this function, I had to first read the six echo images into certain variables. The echo images were provided in DICOM format, which is the typical format for MRI and other medical images. First, I created a function that read and displayed DICOM images using MATLAB's `dicominfo` and `dicomread` functions. This function also allowed me to store these images as integer matrices. I used the Dixon two-point method, discussed below, to create the initial water and fat images, which I inputted into the `lsqcurvefit` function as initial conditions. The `lsqcurvefit` function fits Equation (1) for each pixel, and for images with a matrix size of 256x256, this leads to a long runtime. I modified the function so that instead of fitting per pixel, it would fit patches of size 8x8 instead, which significantly reduced the runtime. Lastly, in order to make my work more accessible to other researchers, I created both an executable version of my MATLAB files and a graphical user interface (GUI). Below is pseudocode outlining the workflow and functions used to create this program.

```
%MATLAB PSEUDOCODE FOR PDFF MAP FUNCTION

%Input in-phase(IP) and out of phase(OP) images
IP1= input(IP1_filename);
IP2= input(IP2_filename);
IP3= input(IP3_filename);
OP1= input(OP1_filename);
OP2= input(OP2_filename);
OP3= input(OP3_filename);

%Input TE times for each IP/OP image
TE_IP1= input(TE_IP1);
TE_IP2= input(TE_IP2);
TE_IP3= input(TE_IP3);
TE_OP1= input(TE_OP1);
TE_OP2= input(TE_OP2);
TE_OP3= input(TE_OP3);

%Calculate seed images
W=Dixon_2_point(IP_OP_Images);
F=Dixon_2_point(IP_OP_Images);

%initial T2* value
T2_star=20*10^-3;

%array that stores estimated T2* values
T2_star_new = [];

%prepare paramters used in lsqcurvefit function call
%The lsqcurvefit call is: lsqcurvefit(fun,x0,xdata,ydata,lb,ub,options)
```

```

fun=@image_equation;      %function to fit
x0=[W F T2_star];        %initial estimates
xdata=IP_OP_Images;      %original data
ydata=stepsizes;         %original output
lb=0;                     %lower bound
ub=inf; %upper bound
options=optimset('Display', 'off', 'UseParallel', false, 'JacobPattern',
jacobpattern);
%these options prevent extraneous information from being displayed in the
command prompt
%when the program is running, allows for the use of parallel cores if
necessary, and
%uses the jacobian to deal with sparse matrices

%compute final water and fat images
patch_size=8; %set size of patch
for i=1:(256/patch_size)
    for j=1:(256/patch_size)
        ydata_patch=get_appropriate_patch(xdata);
        step=lsqcurvfit(fun,x0,xdata,ydata_patch,lb,ub,options);
        xdata_patch=xdata_patch+step; %update W, F, and T2* estimates
    end
end

%Display the PDFFF map
PDFFF=F/(W+F);
plot(PDFFF);

```

*Figure 1: Pseudocode for PDFFF map generating algorithm*

The subsequent paragraphs go into further detail regarding the various functions used in the pseudocode above.

The Dixon two-point method was used to compute the seed images for the curve-fitting algorithm. The Dixon two-point method takes in-phase and out-of-phase images in pairs and creates water and fat images using the following equations.

$$\begin{aligned}
 W &= \frac{1}{2}(IP + OP) \\
 F &= \frac{1}{2}(IP - OP) \quad (2)
 \end{aligned}$$

Here IP stands for the in-phase image, while OP stands for the out-of-phase image. This method is based on the assumption that the only two components in the original images are water and fat. This assumption is mostly valid because the signals from the water and fat components of the image are significantly stronger than from other components, since the magnetic field setup was chosen to excite the proton system and most of the hydrogen found in the body is contained in water and fat. The downsides to this method are that it assumes that fat has only one peak and it ignores the effects of T2\* relaxation. Regardless, the Dixon two-point method provides a good

starting point for the lsqcurvefit algorithm. Additionally, since we have three pairs of in-phase and out-of-phase images, this algorithm produces three pairs of water and fat images. In order to attenuate any bias due to the difference in echo times and T2\* relaxation over time, I used the average of these three sets of images as the input to the lsqcurvefit function [8].

The lsqcurvefit function was used to estimate the water and fat components of the original image given the seed water and fat images derived from the Dixon two-point method. The parameters required by lsqcurvefit are delineated in the pseudocode above. The overarching concept for this nonlinear curve-fitting function is that it takes the data set (xdata) and produces a set of coefficients (step) based on the fitting function (fun) and the initial estimate (x0) that best fits the initial data set to the expected output (ydata) in the least squares sense [9]. The fitting function used by lsqcurvefit models Equation 1. The pseudocode for this function follows.

```
% pseudocode for the fitting function used by lsqcurvefit

function image_equation
patch_size=8; %same as previous size
rp=[0.047, 0.039, 0.006, 0.12, 0.7, 0.088]; %relative magnitude of fat peaks
fp=[39, -32, -125, -166, -217, -243]; %fat peak frequencies relative to
water (Hz)
for i=1:6
    output=S(rp[i],fp[i]); %S is the signal equation
end
end
```

Figure 2: Pseudocode for the function used by lsqcurvefit

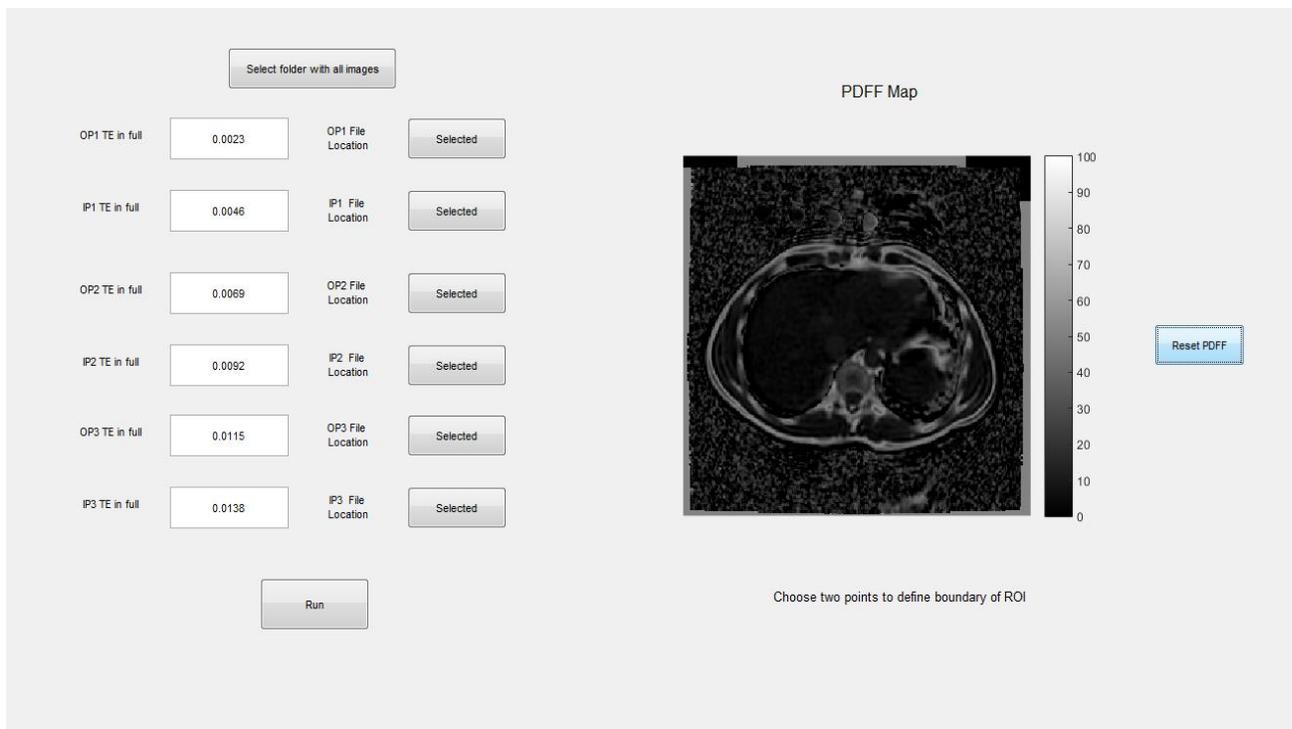
lsqcurvefit stops looking for the appropriate set of coefficients once it has found a minimum or near minimum mean square error. Here, near minimum means that the fitting algorithm has not found the minimum, but the change in the current best estimate is below a certain threshold over an iteration of the fit. The number of iterations lsqcurvefit takes to find this minimum ranges from four to nine. Initially, this algorithm was implemented such that the lsqcurvefit function would fit each pixel of the image. The two issues with this method were that the algorithm took over 45 minutes to complete and fitting each pixel individually did not necessarily accurately capture the relationships between neighboring pixels. Particularly, T2\* values should not vary significantly between neighboring pixels because, on average, pixels within a small range have the same contribution from water and fat. The only location in the image where this argument breaks down is at edges, such as between adipose tissue and internal organs. However, the purpose of this algorithm is to find the fat fraction specifically within the liver, so any discrepancy at the edges is not a significant issue. By running this algorithms over 8x8 patches

instead of per pixel, the run time decreased to just over a minute. The patch method also provided a more accurate physical description of the T2\* variation within the image. At the end of the algorithm, the PDFF map is derived by using the following equation.

$$PDFF = \frac{F}{W + F} \quad (3)$$

The PDFF map is essentially the ratio of the fat present in the image to the whole image, or the fat fraction, assuming that the signal derived from other components in the image is negligible.

After finalizing the algorithm, I assisted in the construction of a GUI that would allow users to input the in-phase and out-of-phase images and their associated TE times, and would output the PDFF map, which is simply the ratio of the fat image to the sum of the water and fat images derived from the fitting algorithm. The GUI also allows the user to select a region of interest (ROI) and provides the average fat fraction for that region. Figure 3 is an image of the interface.



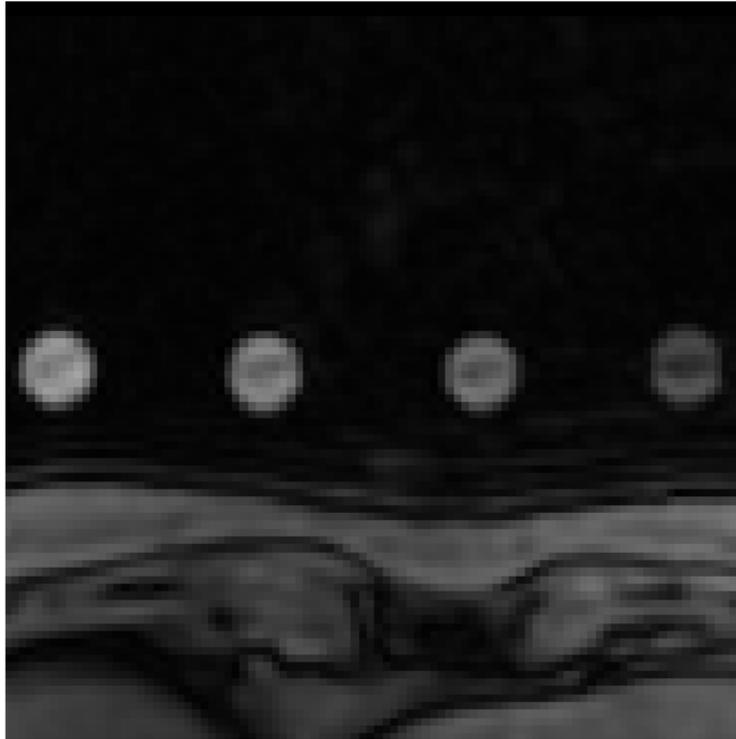
*Figure 3: PDFF Map Creator GUI*

The button labeled “Select folder with all images” specifies which folder the in-phase and out-of-phase images are in. The buttons labeled “Selected” make a file navigation screen pop up that allows the user to specify which file contains the appropriate image. The button labeled “Run” runs the algorithm on the files and the specified TE times. The output PDFF map is displayed to

the right of the window. Upon selecting two points, the average fat fraction of that ROI is displayed below the map. Finally, the button labelled “Reset PDFF” clears the selected ROI, which allows the user to select another region [10]. The following section will discuss the results of this algorithm and compare it to the current standard.

## 4. Results

Each in-phase and out-of-phase image contained four reference samples called phantoms. Figure 4 displays these phantoms. From left to right in the image, these phantoms contained



*Figure 4: Reference Phantoms, out-of-phase image.*

solutions that had 0, 10, 20, and 30 percent fat. Since Figure 4 is out-of-phase, the 0 percent phantom has the highest intensity. The opposite is true for the in-phase images and PDFF map. The phantoms served as reference values for the map. These regions were also used in the comparison between the results derived from this algorithm and from those of the current standard. Figure 5 is an image of the phantoms and their associated fat fractions as well as an ROI within the liver along with its fat fraction.

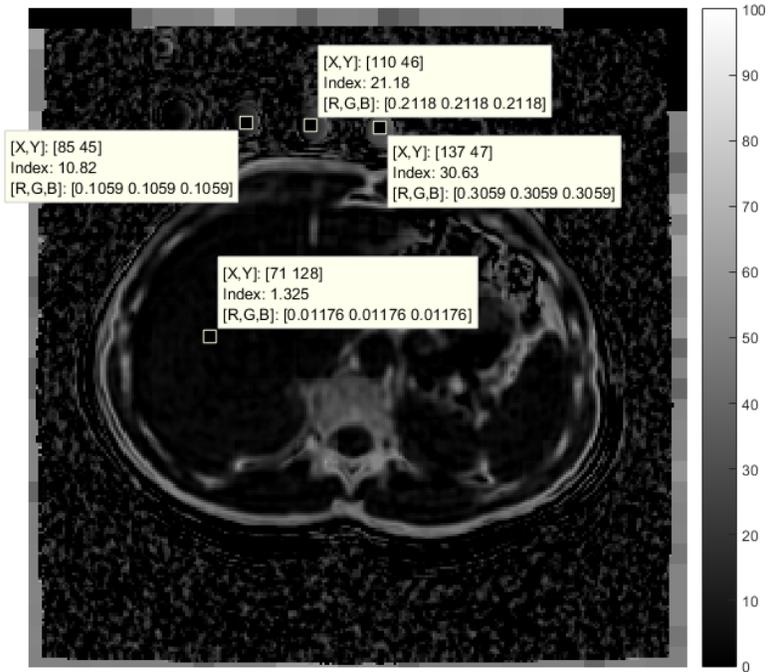


Figure 5: PDFF Map with highlighted ROI

Tables 2-4 compare individual values within the 30 percent phantom for this algorithm's PDFF map and the reference map. Table 4 gives difference values for the ROI within the 30 percent phantom.

**Table 2: Pixel Values within 30% phantom, curve fitting algorithm**

35.94837	33.49743	33.13469	33.44578	33.39977	32.91773	32.47688
34.18414	33.04777	33.09675	33.43917	33.59128	33.19137	32.7646
32.87909	32.00813	32.26051	32.5175	32.98816	33.17489	32.69463
33.06415	32.45798	32.84695	32.82385	32.91145	33.89445	33.9419
33.48673	33.06651	34.18523	33.9635	33.70868	34.83257	35.13354
32.11109	31.44875	32.6064	32.89465	32.79845	33.48861	33.37586
31.06212	30.86131	31.26734	31.36252	31.39568	31.62143	30.54322

**Table 3: Pixel Values within 30% phantom, reference**

32	31	30	31	31	30	30
32	31	30	30	31	30	29
31	29	28	27	28	28	27
28	26	24	23	24	26	26
29	28	27	27	28	29	28
32	31	31	31	31	31	30
32	31	31	31	30	30	28

**Table 4: Difference between reference and algorithm**

-3.94837	-2.49743	-3.13469	-2.44578	-2.39977	-2.91773	-2.47688
-2.18414	-2.04777	-3.09675	-3.43917	-2.59128	-3.19137	-3.7646
-1.87909	-3.00813	-4.26051	-5.5175	-4.98816	-5.17489	-5.69463
-5.06415	-6.45798	-8.84695	-9.82385	-8.91145	-7.89445	-7.9419
-4.48673	-5.06651	-7.18523	-6.9635	-5.70868	-5.83257	-7.13354
-0.11109	-0.44875	-1.6064	-1.89465	-1.79845	-2.48861	-3.37586
0.937878	0.138692	-0.26734	-0.36252	-1.39568	-1.62143	-2.54322

The average value for the 30 percent phantom is 32.93 percent for the curve-fitting algorithm and 29.16 percent for the reference. The average value for the difference table is -3.77 percent, which shows that the results from the algorithm are on average about four percent higher than the reference. There are two possible explanations for this discrepancy. First, the algorithm used by the reference may be different from the algorithm used in this work. The `lsqcurvefit` function has certain built-in optimizations that make it different from other curve-fitting algorithms. Another possible reason for this discrepancy might be the treatment of the  $T2^*$  correction in this algorithm. One of the main areas of exploration during the construction of this algorithm was the effect of the  $T2^*$  estimation on the accuracy of the results. As can be seen from Equation (1), the  $T2^*$  component of the signal is effectively a scaling factor. This signal model assumes that the  $T2^*$  values are the same for both the water and fat components of the signal [2]. Since the  $T2^*$  values account for the contributions from all of the frequencies present in the signal, this assumption should be valid. As shown in Equation (3), the PDFF is a ratio, and since both the water and fat components are scaled by the  $T2^*$  component, this scaling factor cancels out. This means that the  $T2^*$  value does not affect the PDFF at all. Therefore, while correcting the  $T2^*$  values is important for the water and fat images themselves, it does not affect the accuracy of the PDFF map. The following section will summarize the findings of this work and will discuss possible areas for further research, specifically the effect of  $T2^*$  correction on the accuracy of this algorithm.

## 5. Conclusion and Next Steps

The purpose of this research project was to create an algorithm that would provide a proton density fat fraction map given three pairs of in-phase and out-of-phase MRI images of the ROI and their associated TE values. The first attempt at forming a solution involved using the Gauss-Newton algorithm to calculate the water and fat images by calculating a step size and updating the initial guess. However, the second attempt using lsqcurvefit did provide a solution, and the final algorithm was encompassed in a GUI for ease of use.

The implementation of the Gauss-Newton algorithm did not provide a solution that successfully converged. The most likely reason for failure is that the implementation might not have included the optimizations that make curve-fitting algorithms like lsqcurvefit successful. One area of future research is to correct the errors in the implementation of the Gauss-Newton algorithm and include optimizations, such as the patch-wise fitting, to speed up the fitting process. The advantage of having a correct implementation of the Gauss Newton algorithm is that the program would not depend on any MATLAB built-in functions, and thus could be written most other programming languages.

Another significant area of future research is exploring the role of  $T2^*$  as a cofounder for the calculation of the PDFF map. As discussed previously, the  $T2^*$  value cancels out during the calculation of the PDFF because the current model uses the same  $T2^*$  value for both water and fat. Other models to consider are including the estimation of the T2 value, as distinct from  $T2^*$ , for fat along with the  $T2^*$  value of the system, constraining the  $T2^*$  value of fat to be shorter, and including T1 values, another NMR relaxation parameter independent of both T2 and  $T2^*$ , in the model [11]. The first possible model will help differentiate the decay rate between water and fat, which may lead to a more accurate representation of the fat fraction. The T2 decay values are unique to each frequency, and thus would be different for water and fat. One aspect of this model that is ambiguous is the fact that, since fat has six frequencies, the T2 values would be different depending on the peak. In order to incur the least error, the T2 value of the fat peak with the greatest relative weight could be used to represent all peaks. The second model will also differentiate between fat and water; however, shortening the  $T2^*$  fat may lead to an unforeseen bias. The final model adds another level of accuracy by including T1 values, which describe the decay of the longitudinal component of the net magnetization, which is the net magnetic field created by the proton spin system while in the presence of the main magnetic field. Overall,

exploring different  $T2^*$  models may lead to the discovery of a model that most accurately represents physical  $T2^*$  values and produces a PDF map, which is truly independent of confounding NMR relaxation parameters.

## References

- [1] Z. Liang, P. Lauterbur, *Principles of Magnetic Resonance Imaging*, New York, IEEE Press
- [2] S. Reeder, C. Sirlin, *Quantification of Liver Fat with Magnetic Resonance Imaging*, Magnetic Resonance Imaging Clinical North America, 2010 August
- [3] I. Idilman et al., *Hepatic Steatosis: Quantification by Proton Density Fat Fraction with MR Imaging versus Liver Biopsy*, Radiology, 2013 June
- [4] S. Reeder et al., *Relationship Between Proton Density Fat Fraction and True Fat Concentration for In Vivo Fat Quantification with Magnetic Resonance Imaging*, Proceedings of the International Magnetic Resonance Imaging Society, 2011
- [5] S. Reeder et al., *Quantitative Assessment of Liver Fat with Magnetic Resonance Imaging and Spectroscopy*, Journal of Magnetic Resonance Imaging, 2011
- [6] G. Hamilton et al., *In Vivo Characterization of the Liver Fat  $^1H$  MR Spectrum*, NMR Biomed., 2011 August
- [7] S. Reeder et al., *Combination of Complex-Based and Magnitude-Based Multiecho Water-Fat Separation for Accurate Quantification of Fat-Fraction*, Magnetic Resonance Imaging, 2011 July
- [8] J. Pauly, *Dixon Reconstruction*, 2005 October
- [9] *lsqcurvefit*, Mathworks, webpage, Available at: <https://www.mathworks.com/help/optim/ug/lsqcurvefit.html> , Accessed May 2017
- [10] *Create a Simple App Using GUIDE*, Mathworks, webpage, Available at: [https://www.mathworks.com/help/matlab/creating\\_guis/about-the-simple-guide-gui-example.html](https://www.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html) , Accessed May 2017
- [11] M. Bydder et al., *Relaxation Effects in the Quantification of Fat Using Gradient Echo Imaging*, Magnetic Resonance Imaging, 2008 April