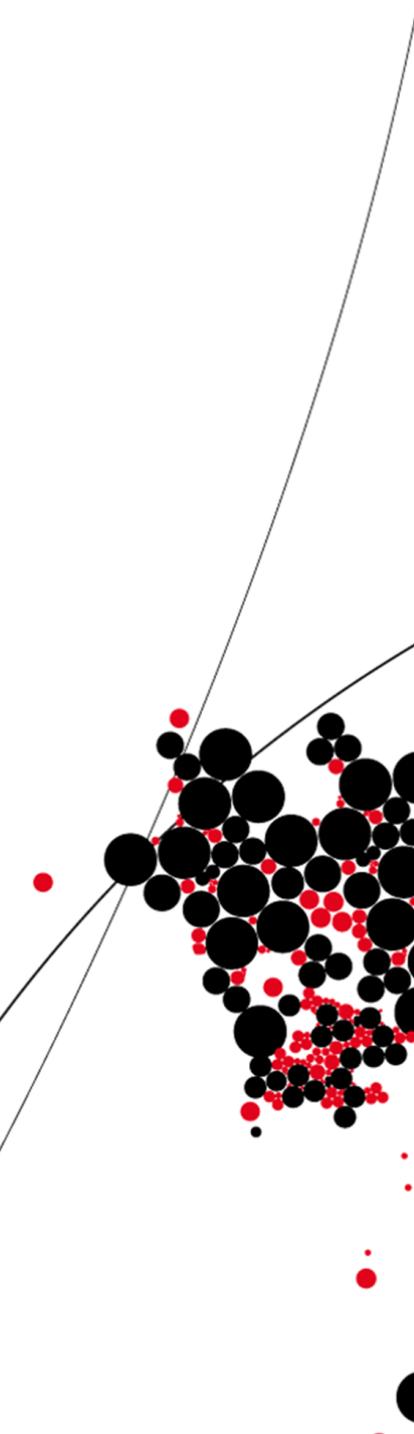




UNIVERSITY OF TWENTE.



**Parallel magnetic particle imaging:
compressed sensing of field free
line excitations**

Normen Oude Booijink

MSc. Thesis

October 2016

Assessment committee

Prof. dr. S.A. van Gils

Dr. C. Brune

Dr. B. ten Haken

Prof. dr. S. Conolly

Supervisor

Dr. C. Brune

Applied Analysis group
Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

We developed a speed versus quality improvement of the currently most promising imaging modality that images superparamagnetic nanoparticles which are injected into a body: Magnetic Particle Imaging (MPI). We design a more informative hardware setup, that uses multiple receive channels similar to the setup in parallel Magnetic Resonance Imaging (MRI). To process the simulated data from these parallel coils, we use convex optimization reconstruction algorithms that can combine the extra information to find either the same image quality at shorter scan time, or improve image quality at the same scan time. To get the speed improvement, we decrease the number of projections in field free line MPI, so that Fourier space is undersampled. The idea for this was inspired by results from Magnetic Resonance Imaging, and the Fourier slice theorem. This motivated us to be the first to prove that the concept of parallel imaging can be applied to MPI.

Keywords: Magnetic Particle Imaging, undersampling, parallel imaging, convex optimization

Contents

1	Introduction	4
2	Parallel Imaging and MPI physics	5
2.1	Parallel imaging	5
2.1.1	Checklist for parallel imaging	8
2.2	Physical principles of MPI	8
2.3	Reconstruction approaches in MPI	10
2.3.1	System matrix MPI	10
2.3.2	x-space MPI	10
2.4	Field free line MPI	14
2.5	Parallel MPI	16
2.6	The parallel FFL MPI physics simulator	19
3	Inverse problem formulation.	19
3.1	Notation and basics	19
3.2	Forward model formulation for parallel MPI	21
3.3	Image reconstruction via minimization	21
3.4	The adjoint operator for parallel MPI	22
4	Convex optimization and regularizers	23
4.1	Convex functions	23
4.2	Regularization	23
4.2.1	Spaces of images	25
4.2.2	Properties of the solution to the objective function.	25
4.2.3	Total variation	26
4.3	From theory to practice	28
4.4	Algorithms	29
4.4.1	Conjugate gradients	29
4.4.2	Primal-dual hybrid gradients	29
4.4.3	Simple case	32
4.4.4	Adding the total variation regularizer	32
5	Results	34
5.1	Parallel MPI reconstructions from data generated by the forward operator	35
5.2	Parallel MPI reconstructions from simulated data	36
5.3	Noise and regularization performance	40
5.3.1	Added noise, no regularization	40
5.3.2	Regularization	41
6	Conclusion and future scope	43

Acknowledgements

Working on this project was a lot of fun for me; it required knowledge of both mathematics and physics, applied to medical imaging. These are disciplines I have always liked to spend time on. I spent about half of the time at the University of Twente (UT), and the other half at the University of California in Berkeley (UCB). At both universities I was guided very well, for which I would like to express my gratitude. First of all, I thank my supervisor from the UT: Dr. Christoph Brune. Christoph has put effort into the project from start to end, and has continuously been involved and available for questions in our weekly meetings, which lead to very good, informal discussions. Also I would like to thank Prof. dr. Stephan van Gils, who is the head of the Applied Analysis group that I belong to at the UT and has always taken the time to help me if needed. The collaboration between the UT and UCB was made possible by Dr. Bennie ten Haken, who introduced me to the Berkeley Imaging Systems Laboratory. I am very grateful for that opportunity.

The head of the laboratory in Berkeley is Prof. dr. Steven Conolly, with whom I had good discussions and I thank him a lot for the freedom he gave me to create my own results. Also, Steve is a very nice person to have as the head of a group, as there is a very low threshold for questions and sharing your opinion. Furthermore, I thank all the people in the lab for the help they gave me, with special thanks to Daniel Hensley and Zhi Wei Tay for the awesome discussions we had and results we got. Next to the work in the lab, I did a course on Magnetic Resonance Imaging. Thanks are due to Dr. Michael Lustig, Wenwen Jiang and Michael Driscoll, who have helped and collaborated with me on both the lectures and the final project of the course. Finally, I would like to thank my parents, the UT, the Hendrik Mullerfonds and the Fundatie van de Vrijvrouwe van Renswoude for their financial aid during my time in Berkeley.

1 Introduction

Magnetic Particle Imaging (MPI) is a relatively young imaging modality, that aims to image the density distribution of a magnetic tracer in a body. One could say that MPI is currently in the state where Magnetic Resonance Imaging (MRI) was in the 1980s and 90s: the concept works and (animal) scanners have been built. Back then people tried to mathematically optimize excitation strategies for MRI, for example using control theory. Also, mathematics was used to extract more information from a scan, by optimizing over hardware setups and scanning trajectories [34],[35],[40]. The latter are called compressed sensing and parallel imaging techniques, and have inspired this thesis. We argue that there are similar opportunities for MPI, as the hardware of MPI has similarities with MRI. But let us first motivate why MPI is an imaging modality worth spending time on in the first place.

The use of magnetic nanoparticles in medicine was first investigated during the end of the 20th century, and started to take serious forms in the 2000s as people started to see competitive advantages over some of the established techniques, such as improvements in contrast and patient safety in imaging applications, and accurate drug delivery through magnetic targeting [1],[2],[3],[4],[5],[6]. Also more recently, several applications have been presented that use the size and magnetic properties of certain nanoparticles [6],[7],[8],[9],[10]. Regarding medical imaging techniques, the most promising of these applications is Magnetic Particle Imaging (MPI). The modality was invented during the years 2000 and 2001 by Jurgen Weizenecker and Bernard Gleich [11], both working for the Philips Research Laboratory in Hamburg, Germany. The physical principles along with some first experiments were published in the journal Nature by these two researchers in 2005 [7]. The concept of MPI relies on the nonlinear magnetization curve of superparamagnetic iron oxide nanoparticles (SPIONs). Because of their size - only a few atoms wide - the SPIONs are very suitable for injection into the human body and can penetrate through lots of different tissues. The body is then placed in a scanner that uses a smart hardware setup mostly consisting of coils, to image the nanoparticle distribution in the body. A more detailed description of the exact functioning of an MPI scanner and how the image is formed are given in sections 2.2 and 2.3. With this knowledge, we investigate how both the hardware and the reconstruction should be redesigned to make parallel MPI feasible.

The big competitive advantage of MPI is its contrast performance in combination with patient safety. Contrast agents and tracers in medical imaging are present in almost every modality as they provide visualization of structures that can not be seen without, and are crucial for many diagnoses. But most of the current contrast agents like iodine in X-ray, can not be used for the imaging of Chronic Kidney Disease (CKD) patients, which is a very large group [12]. And in PET and SPECT scans, the radioactive tracers yield safety and logistic problems for both the patient and the medical personnel. Importantly, the iron oxide nanoparticles used in MPI are not radioactive and are processed in the liver, so they don't affect the kidneys. The only other current imaging modality where SPIONs are used as a contrast agent is MRI, but it is suboptimal because the background signal from the host tissue is a limiting factor with respect to image contrast [13],[14]. MPI aims to only measure the signal that emanates from the injected magnetic tracer, and it does so by taking advantage of the nonlinear magnetization curve of SPIONs. By analyzing the higher harmonics of the signal that emanates from the nonlinear response of these SPIONs to an exciting magnetic field, the signal due to the SPIONs can be extracted from the entire inductively received signal. This yields an image that results from a signal that is not distorted by signals from the hosting tissue or the exciting magnetic field. Therefore, very high contrast can be obtained via this imaging modality. In order to compete with other imaging modalities, future research on MPI will have to focus on resolution enhancement, speed and new applications. Room for improvement can mainly be found in the hardware of MPI scanners, the image reconstruction process and the design of the nanoparticles. The speed improvement presented in this thesis will mainly focus on a change in the reconstruction approach, that also implies a new hardware setup.

Finally, it is good to know that there are currently two different approaches to the image reconstruction process in MPI. The first approach, initiated by Weizenecker and Gleich, was developed by the Philips Research Laboratory in Hamburg in collaboration with the University of Lübeck. It first "calibrates" the MPI system by scanning a phantom with unit concentration, recording the response of the system at each location. This information is stored in a matrix, called the system matrix. Next, the desired phantom or body is scanned, and the recorded signals are compared with the system matrix to obtain an image. The other approach, initiated by Steven Conolly and Patrick Goodwill from the University of California in Berkeley, is a more direct reconstruction method. They mathematically derived a relation between the amplitude of the signal and the nanoparticle concentration. This relation is then used to grid this density onto the right spatial location in the image. How this spatial location is determined, is actually the main innovatory concept of MPI, which got Gleich and Weizenecker their fame. This concept is explained in section 2.2. The (dis)advantages of both approaches will be discussed in subsection 2.3. In this project we collaborated with the University of California

in Berkeley and focused on their method, but our parallel imaging model can also easily be applied to the system matrix approach.

2 Parallel Imaging and MPI physics

In this section we will first describe what parallel imaging is, and why it is useful. Then, we explore the possibilities for parallel imaging in MPI.

2.1 Parallel imaging

In parallel imaging, multiple receive channels are used to pick up a certain signal, where the spatial location of these receive channels is exploited to extract extra information. This extra information can either be used to improve image quality, or to allow for less data acquisition leading to shorter scanning times. For example, in parallel MRI, multiple images m_1, m_2, \dots, m_L are obtained due to the presence of L different receive coils. These images can then be combined into the desired final image m by defining a good forward model, that can be inverted directly or solved algorithmically, dependent on the complexity of the scanning procedure. A good reference for a historical overview of the different algorithms and methods that use parallel imaging to increase MRI performance is [32]. The most important reconstruction algorithms are SMASH [34], SENSE [35], GRAPPA [36] and (E)SPIRiT [37],[38]. SENSE and ESPIRiT are the best option in our opinion, since they aim to fully reconstruct the actual image $m(x, y)$, whereas SMASH and GRAPPA aim to compute an approximation. We now give two examples from MRI, to illustrate the power of parallel imaging. The first example gives a good intuition as to why the power of parallel imaging comes from the extra spatial information. The second example is more general, and will be very useful for our approach to parallel MPI.



Figure 1: Parallel MRI in practice. In this photo, an MRI of the brain is made using 12 parallel receive coils.

First example (Cartesian MRI)

Assume that we aim to reconstruct an image that contains density information $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$ at pixels \mathbf{x}_p , where p is an index that runs over the number of pixels. Each receive channel $C_i, i \in \{1, \dots, L\}$, has a spatially variant sensitivity map $S_i(\mathbf{x})$, yielding images

$$\rho_i(\mathbf{x}) = S_i(\mathbf{x})\rho(\mathbf{x})$$

after an MRI scan. So, for a particular pixel p , we have

$$\underbrace{\begin{pmatrix} \rho_1(\mathbf{x}_p) \\ \rho_2(\mathbf{x}_p) \\ \vdots \\ \rho_L(\mathbf{x}_p) \end{pmatrix}}_{\boldsymbol{\rho}_s} = \underbrace{\begin{pmatrix} S_1(\mathbf{x}_p) \\ S_2(\mathbf{x}_p) \\ \vdots \\ S_L(\mathbf{x}_p) \end{pmatrix}}_{\mathbf{S}} \rho(\mathbf{x}_p) \quad (1)$$

and we can solve

$$\rho(\mathbf{x}_p) = (\mathbf{S}^\dagger \mathbf{S})^{-1} \mathbf{S}^\dagger \boldsymbol{\rho}_s \quad (2)$$

for each pixel, to obtain the desired density image $\rho(\mathbf{x})$. Now, the power of parallel imaging lies in the acceleration of scanning times in MRI. In Cartesian sampling trajectories, the scan time is dictated by the number of phase encodings, as usually only one phase encode is done within every time repetition (TR). So, if one would divide the number of phase encodes by R , the entire MRI scanning time will reduce by a factor of R . In our example of Cartesian parallel MR imaging with undersampling factor R , the phase encode is in the x -direction, so we put zeros on the horizontal lines in Fourier space (k -space) we have not sampled. Then, we compute the inverse Fourier transform to yield an aliased image (due to the increase in Δk_y and thus decrease in field of view in the y -direction). Also, for each pixel \mathbf{x}_p in the aliased image, one backtracks the R ‘source pixels’ in the actual image that have contributed to the aliasing in pixel \mathbf{x}_p . Call these source pixels $\mathbf{x}_{1p}, \mathbf{x}_{2p}, \dots, \mathbf{x}_{Rp}$. Then, for each pixel \mathbf{x}_p we can solve the system

$$\underbrace{\begin{pmatrix} \rho_1(\mathbf{x}_p) \\ \rho_2(\mathbf{x}_p) \\ \vdots \\ \rho_L(\mathbf{x}_p) \end{pmatrix}}_{\boldsymbol{\rho}_s} = \underbrace{\begin{pmatrix} S_1(\mathbf{x}_{1p}) & S_1(\mathbf{x}_{2p}) & \cdots & S_1(\mathbf{x}_{Rp}) \\ S_2(\mathbf{x}_{1p}) & S_2(\mathbf{x}_{2p}) & \cdots & S_2(\mathbf{x}_{Rp}) \\ \vdots & \vdots & \ddots & \vdots \\ S_L(\mathbf{x}_{1p}) & S_L(\mathbf{x}_{2p}) & \cdots & S_L(\mathbf{x}_{Rp}) \end{pmatrix}}_{\mathbf{S}} \underbrace{\begin{pmatrix} \rho(\mathbf{x}_{1p}) \\ \rho(\mathbf{x}_{2p}) \\ \vdots \\ \rho(\mathbf{x}_{Rp}) \end{pmatrix}}_{\boldsymbol{\rho}(\mathbf{x}_p)} \quad (3)$$

which has solution

$$\boldsymbol{\rho}(\mathbf{x}_p) = (\mathbf{S}^\perp \mathbf{S})^{-1} \mathbf{S}^\perp \boldsymbol{\rho}_s. \quad (4)$$

So we see that we obtain the density values for R pixel of the actual image using a system with size $L \times R$. An example of parallel imaging results for an 8 coil system is given in Figures 2 and 3. We see that the system does quite a good job at reconstructing the original brain image while we undersample by a factor of $R = 2$. Important to note, is that undersampling in MRI occurs in Fourier-space. Due to the undersampling, less information is obtained, but not all information is lost. One needs to be careful with this; there is only room for undersampling when it doesn’t directly delete fundamental information that can not be recovered. We will see that this forms a problem in standard MPI.

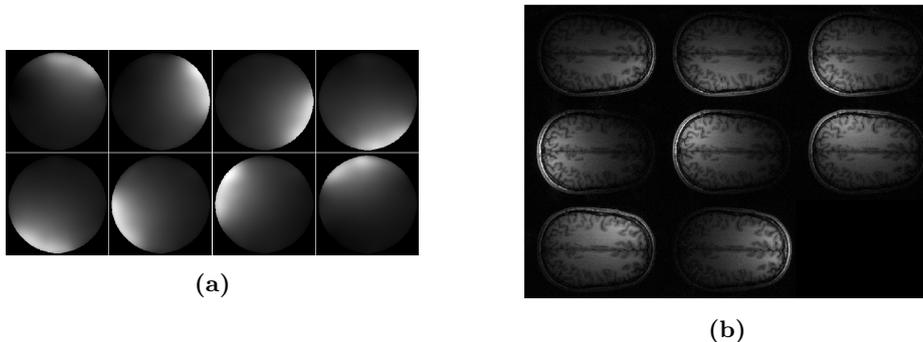


Figure 2: **(a)** Estimates for the sensitivity maps of eight coils used for parallel MR imaging of the brain. **(b)** The resulting eight MR images of a brain slice when sampling at the Nyquist rate.

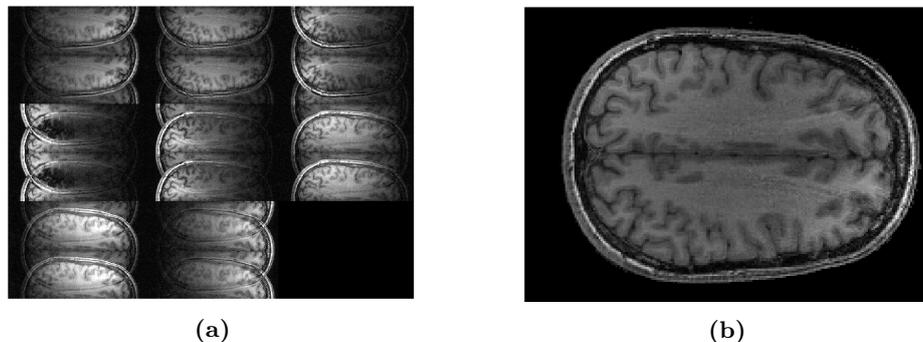


Figure 3: **(a)**: The 8 coil images reconstructed from undersampled data: a factor $R = 2$ in the y -direction. Note the increased signal intensity at aliased pixels. **(b)**: Corresponding parallel imaging reconstruction using SENSE.

Second example (non-Cartesian MRI)

In MRI, other sampling trajectories than Cartesian can be favorable in terms of scanning times and signal information. Spiral and radial trajectories for example, collect more data at the center of k-space, which contains most of the information. An other example is random sampling, which is favorable for compressed sensing [40]. Parallel imaging can again be used to obtain good quality images while k-space is being undersampled

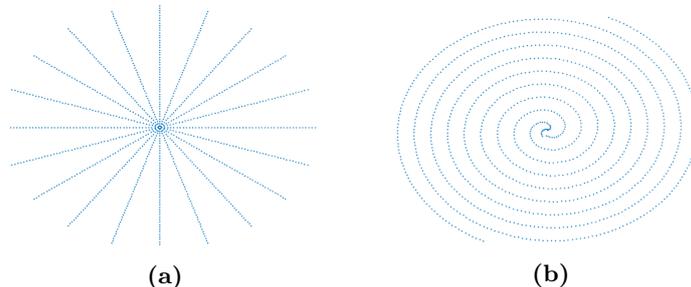


Figure 4: Two non-Cartesian sampling trajectories in MRI. We will see that trajectory (a) is the key to parallel MPI.

non-uniformly. Two of these sampling trajectories are given in Figure 4. Unfortunately, in these situations it is impossible to backtrack which pixels contributed to the aliasing in each coil image like we did in the first example. Hence, the reconstruction approach from the previous example doesn't apply anymore, because basically any pixel could have contributed to the aliasing and the matrix \mathbf{S} would become too big and impossible to form. Therefore, a forward model is derived and the corresponding inverse problem is solved iteratively through an algorithm. The general form of the forward model is

$$E\rho = d, \tag{5}$$

where $E : X \rightarrow Y$ is an operator that represents the MRI scanning procedure. With each iteration of the algorithm that computes ρ given the operator E and data d , an estimate $\hat{\rho}$ of the desired ρ is determined such that $E\hat{\rho}$ lies closer to d than in the previous iteration. In Figure 5, we see $\hat{\rho}$ in several iterations of the parallel MRI reconstruction algorithm SENSE. In this image, a spiral undersampling trajectory was chosen, for an axial slice of a human brain. Now, since the coil sensitivity maps are incorporated in E , the extra information is implicitly used. We will talk more about this in section 3, because the forward model approach to parallel imaging described in this example will be the one we have to use for parallel MPI.

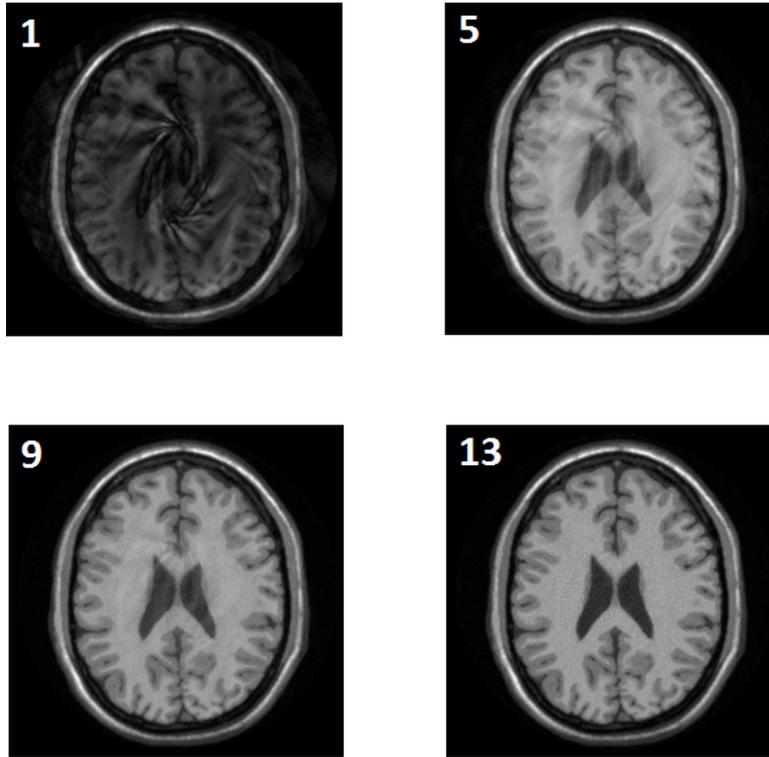


Figure 5: Iterative SENSE reconstruction using the conjugate gradient method. The reconstructions after the 1st, 5th, 9th and 13th iteration of the algorithm that solves (5) are shown.

2.1.1 Checklist for parallel imaging

In the previous section, we saw that parallel imaging can be used to decrease scan times in MRI. This is possible because MRI fulfills three key requirements that we identified. To obtain shorter scanning times using parallel imaging, a medical imaging technique has to suffice to:

1. **It is possible to measure the signal at different locations.** Multiple receive channels with different sensitivities to signals can only be realized if the signal can be detected from different locations around the object. An imaging technique that fails this condition is X-ray CT ; the signal that is being measured is just an attenuation of the original beam, that only goes in one direction. Hence, the only location of the detector that makes sense, is on the opposite side of the transmitter. Therefore, parallel imaging is not suitable for X-ray CT.
2. **There is room for undersampling while still scanning each location of the object** Undersampling, i.e. obtaining less data, is only desirable if the whole object can still be reconstructed. That is why data acquisition in Fourier space is so powerful, because each data point in Fourier space could reflect information from any location in the object. There might still be enough information in a subset of the Fourier data to reconstruct the whole object.
3. **Except for sensitivity differences, each receive channel has to receive the same signal.** If the data from two receive channels reflect fundamentally different physics, these data can not be combined, and parallel imaging makes no sense anymore.

In the next subsections, we will explore if MPI in its standard form fulfills these requirements, and if not, which hardware adjustments have to be made.

2.2 Physical principles of MPI

In order to explore the possibilities for parallel Magnetic Particle Imaging, we need to understand how MPI works. MPI is the first imaging modality in which nanoparticles are not just supportive contrast agents, as in MRI [5], but the only source for the informative signal. As a result, only the SPION density is visual in the image, yielding high contrast. The recorded signal in MPI is originating from the change in magnetization of the SPIONs in the body, when they are subject to changing magnetic fields. This causes a change in magnetic flux through the receive coils, inducing a current through the coil that we can measure.

A beautiful part of the MPI technique is the way Weizenecker and Gleich determine where each part of the received signal, which is just a time series, came from in space. First, a body is subjected to a strong gradient field generated by opposing Maxwell coils, which is also referred to as the *gradient field*. In this gradient field, a field free point (FFP) is present. An illustration of this is given in Figure 6.

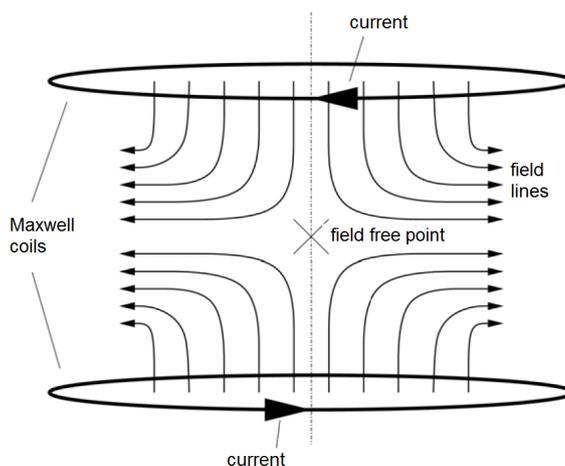


Figure 6: The gradient field produced by two opposite Maxwell coils contains a field free point (FFP).

To magnetically saturate all magnetic material outside the FFP, the gradient field generated by the Maxwell coils is chosen to be about 3 to 6 Tm^{-1} in strength. In such a strong field, all magnetic material is saturated *except for* the magnetic material inside the FFP. In the absence of a magnetic field, a population of SPIONs has zero magnetic moment: as each individual SPION's magnetic moment is randomly oriented due to thermal agitation, their magnetic moments cancel out yielding zero magnetization for an ensemble of these particles. So after the gradient field is turned on, only the particles in the FFP are randomly aligned. Now comes the crucial part: a spatially homogeneous magnetic field is added, causing the FFP to shift to another position. The particles that were at the previous location of the FFP will now all align their magnetic moments with the gradient field, causing a change of magnetic flux through the receive coils that we can register, and know to have originated from the FFP. The amplitude of the signal then holds information about the nanoparticle density at that location.

So two more kinds of hardware are needed: *Drive field coils*, which generate the additional homogeneous magnetic field, and *receive coils* that register the response of the nanoparticles in the FFP. Two type of drive field coils are present. The first one is the exciting drive field, that rapidly changes it's amplitude so that the FFP also shifts rapidly, causing a sudden change in magnetization of the SPIONs. The second type of drive field coil creates a field called the *bias field* that is similar to the drive field but changes it's amplitude relatively slow, by a bigger range. In this way, the bias field aims to slowly move the FFP over the body so that the whole body is being scanned. A schematic setup of a three-dimensional MPI scanner as proposed by Weizenecker and Gleich [15] is given in Figure 7. The German group has a slightly different nomenclature: the gradient fields are called selection fields.

When a single frequency excitation is applied to the system, the signal generated by the particles will contain a series of harmonics due to the nonlinear Langevin response of the SPIONs. These higher harmonics are unique to the SPIONs and are therefore crucial in extracting those parts of the signal that come from the SPIONs, and not the drive fields. The excitation frequency is filtered out, leaving only the signal that is generated by the change in magnetization of the particles.

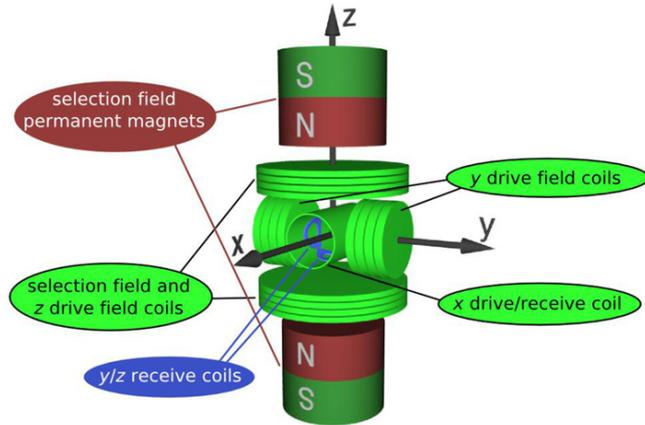


Figure 7: Schematic setup of a 3-D MPI scanner as proposed by the Philips research group in Hamburg. Note that the selection field is what we call the gradient field.

2.3 Reconstruction approaches in MPI

The goal of the MPI reconstruction process is to find the exact spatial distribution of the SPIONs over the body, including quantitative density information. In medical imaging, the region of interest in the body is often referred to as field of view (FOV). The FFP has to be moved over the entire FOV to obtain the spatial density distribution of the SPIONs, recording a signal at each time step to form a discrete image. As mentioned in the previous section, we don't have to move the body to move the FFP's relative position. Magnetic fields add, so every contribution by the homogeneous drive fields and bias fields will shift the location of the FFP created by the gradient field. By building the drive field coils in each of the three Cartesian directions, the FFP can be moved over 3-D space in any way that is desirable. The drive field rapidly shifts the FFP and causes a rapid change in magnetization of the SPIONs. In addition, the bias field slowly moves the FFP over the entire FOV. The change in magnetization is recorded by a receive coil, forming the data d . Currently there are two approaches to reconstructing an MPI image from d : system matrix MPI and x-space MPI.

2.3.1 System matrix MPI

This reconstruction method was initiated by Weizenecker and Gleich, and later adapted by several researchers, mostly active in Germany [7] [16] [17]. The conventional model assumes a linear connection between the particle density $\rho(x)$ and the MPI signal $\mathbf{s}(t)$. The system matrix \mathbf{G} encodes which coefficient should be used in this relationship:

$$\mathbf{s}(t) = \int G(\mathbf{x}, t) \rho(x) d\mathbf{x}, \quad t \in [t_s, t_e], \quad (6)$$

where t_s and t_e denote the start and end time of the MPI scan, respectively. G indirectly contains all information hidden in the MPI system and scanning trajectory: at each location of the FFP in the *calibration scan* on a phantom with unit concentration, the receive coils register a signal that originated from the FFP. But, even though the particle density is equal over the entire FOV, the geometry of the scanning procedure causes a variation of amplitudes in the signal. More on this can be read in the discussion section of [19]. When the actual scan on a phantom with unknown particle density distribution is done, the scanning procedure needs to be exactly the same as in the calibration scan. Then, the obtained data s and the system matrix G can be used to compute $\rho(x)$. In the papers by Gleich and Weizenecker, this is done using direct inversion methods like Singular Value Decomposition on equation (1) [16].

A big disadvantage of the system-matrix approach is the time-consuming calibration scan. In [20], it took 6 hours to scan a 34x20x28 3D grid. Moreover, this calibration is only valid for a specific FOV and scanning trajectory. Changes in the setup would need a new calibration scan. But, from about 2009 up to now, the researchers from the Philips group try to fix this problem by theoretically deriving the system matrix [18],[19]. If this can be done accurately, there is no need for a physical calibration to obtain G .

2.3.2 x-space MPI

The basic principle of magnetic particle imaging is in fact quite simple: one receives a signal and knows where this signal came from. Why not directly grid this signal to its location in space? Well, x-space MPI aims to do exactly that. In order to do so, one must be able to relate the received signal at time step t_k to the SPION

density at the location of the FFP at time t_k . So the first papers on x-space MPI are all about the physics that ultimately lead to the received signal \mathbf{s} . In [21], the first complete x-space model is given for 1-D signals. The location of the FFP can explicitly be given by setting the total magnetic field equal to zero. In 1-D, this yields

$$H(x, t) = H_0(t) - G \cdot x = 0.$$

Note that the drive field H_0 is only time-dependent, and the gradient field $G \cdot x$ is space dependent. The gradient G was conveniently chosen with a negative sign. The total magnetic field expressed as a function of space, time and the FFP is thus

$$H(x, t) = G \cdot (x - x_s(t)), \quad (7)$$

where $x_s(t)$ denotes the location of the FFP as a function of time. As said before, the MPI signal is the result of the change in magnetic flux ϕ through the receive coils. This change is due to the change in total magnetization M of the nanoparticles. In the 1-dimensional case, we get

$$s(t) = B_1 \frac{d}{dt} \phi(t) = B_1 \frac{d}{dt} \int M(x, t) dx. \quad (8)$$

Where B_1 [T/A] is just a constant that models the sensitivity of the receive coil. Langevin theory [23] is used to describe the magnetization of a SPION as a function of the applied magnetic field that they are subject to. The magnitude of the magnetization can be seen as the amount of atomic dipoles in the SPION molecules that are aligned with the applied magnetic field. If all atomic dipoles are aligned, we call it magnetically saturated. Langevin theory claims that the magnetization at a single point x in a magnetic field H is

$$M(H) = m\rho(x)\mathcal{L}(kH). \quad (9)$$

m is the magnetic moment of a nanoparticle [$\text{A}\cdot\text{m}^{-2}$], ρ is the nanoparticle density [$\text{kg}\cdot\text{m}^{-3}$] and

$$k = \frac{\mu_0 m}{k_B T}.$$

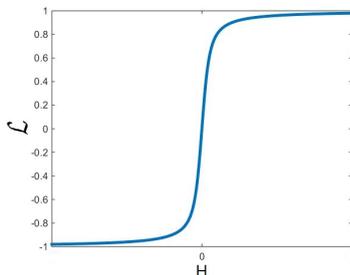
Where μ_0 is the vacuum permeability, k_B is Boltzmann's constant and T is the temperature. The Langevin function

$$\mathcal{L}(x) = \coth(x) - \frac{1}{x}$$

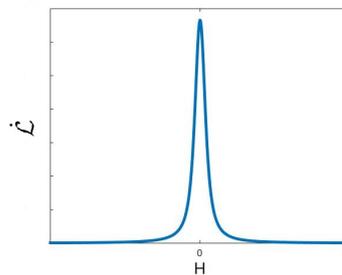
is depicted in Figure 8a. Note that it's extreme values are -1 and 1, which is what we would expect from equation (9). With this knowledge, we can combine equations (7),(8) and (9) to obtain

$$s(t) = B_1 m \rho(x) * \dot{\mathcal{L}}(kG \cdot x) \Big|_{x=x_s(t)} \quad kG \cdot \dot{x}_s(t). \quad (10)$$

So, ignoring the constants involved, the received signal s at a time instant t_k of a 1-D MPI scan is the result of a convolution of the nanoparticle density in the FFP at time t_k with the derivative of the Langevin function, multiplied by the FFP velocity. In [22], the above result is derived for the multidimensional case. There are



(a) The Langevin function \mathcal{L} describes the magnetization of SPIONs as a function of the externally applied magnetic field.



(b) The point spread function in 1-D MPI is the derivative of the Langevin function.

a lot of analogies, but there are differences as well. The increased number of degrees of freedom in scanning trajectories bring about interesting physical and mathematical challenges to find the best image in terms of resolution and signal to noise ratio (SNR). We close this section with a short summary of the dynamics in multidimensional x-space MPI, as we use the theory to prove that parallel MPI is feasible.

Multi-dimensional x-space MPI

In [22], the researchers from the Berkeley Imaging Systems Laboratory write a multi-dimensional follow-up to the 1-D paper [21]. A gradient field in three dimensions with a FFP can be realized by $\mathbf{H}=\mathbf{G}\mathbf{x}$, with

$$\mathbf{G} = \begin{bmatrix} G_{xx} & 0 & 0 \\ 0 & G_{yy} & 0 \\ 0 & 0 & G_{zz} \end{bmatrix} = G_{zz} \begin{bmatrix} -\frac{1}{2} & 0 & 0 \\ 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where G_{zz} is typically chosen such that $\mu_0 G_{zz} \approx 2.4$ to $6 T/m$. To move the FFP over the 3-D volume, an additional homogeneous but time-varying field $\mathbf{H}_s(t) = [H_x(t), H_y(t), H_z(t)]$ is added to the gradient field. Giving G_{zz} a convenient negative sign, the total field is

$$\mathbf{H}(\mathbf{x}, t) = \mathbf{H}_s(t) - \mathbf{G}\mathbf{x},$$

so that the location of the FFP is

$$\mathbf{x}_s(t) = \mathbf{G}^{-1}\mathbf{H}_s(t). \quad (11)$$

In 3-D, there are more degrees of freedom w.r.t the excitation direction and the receive coil geometries. Therefore, the derived PSF $\mathbf{h}(\mathbf{x})$ is a 3×3 matrix with a function of (x, y, z) at each entry:

$$\begin{aligned} \mathbf{h}(\mathbf{x}) &= \dot{\mathcal{L}}(\|\mathbf{G}\mathbf{x}\|/H_{sat}) \frac{\mathbf{G}\mathbf{x}}{\|\mathbf{G}\mathbf{x}\|} \left(\frac{\mathbf{G}\mathbf{x}}{\|\mathbf{G}\mathbf{x}\|} \right)^T \mathbf{G} \\ &+ \frac{\mathcal{L}(\|\mathbf{G}\mathbf{x}\|/H_{sat})}{\|\mathbf{G}\mathbf{x}\|/H_{sat}} \left(\mathbf{I} - \frac{\mathbf{G}\mathbf{x}}{\|\mathbf{G}\mathbf{x}\|} \left(\frac{\mathbf{G}\mathbf{x}}{\|\mathbf{G}\mathbf{x}\|} \right)^T \right) \mathbf{G}, \end{aligned} \quad (12)$$

where \mathcal{L} denotes the langevin function as in section 2.3.2. An explicit PSF in scalar functional form is obtained once a scanning direction $\hat{\mathbf{x}}_s$ of the FFP is chosen:

$$h(\mathbf{x}) = \mathbf{x}_c^T \mathbf{h}(\mathbf{x}) \hat{\mathbf{x}}_s, \quad (13)$$

where \mathbf{x}_c is the receive coil vector. Note that we use $\hat{\cdot}$ to indicate a normalized vector. Similar to the 1-D case, the blurred MPI image results from a convolution with the PSF:

$$\tilde{\rho}(\mathbf{x}) = \frac{s(t)}{\|\hat{\mathbf{x}}\|} = \frac{\mathbf{B}_1(\mathbf{x})m}{H_{sat}} \rho(\mathbf{x}) * * * \mathbf{x}_c^T \mathbf{h}(\mathbf{x}) \hat{\mathbf{x}}_s \Big|_{\mathbf{x}=\mathbf{x}_s(t)}, \quad (14)$$

where $\mathbf{B}_1(\mathbf{x})$ is the receive coil sensitivity. We will omit the term $\frac{\mathbf{B}_1(\mathbf{x})m}{H_{sat}}$ in most of the calculations in this report, for notation purposes. To give the reader some more intuition and visual support, we will shortly elaborate on the 2-D MPI PSF. Also, a good understanding of the PSF was crucial to the design of our parallel MPI approach.

Image resolution and the point spread function

In the world of imaging research, there are a lot of different measures for resolution [24]. In medical imaging, the measure that is mostly used is the full-width at half maximum of the point spread function of the imaging system. So in order to determine or enhance the resolution of a medical imaging technique, the point spread function must be understood in detail. For us, knowledge of the PSF will turn out to be crucial to prove that parallel MPI is possible. So let's look at the details of the PSF for the 2-D case. Recall from equation (12) that for multidimensional MPI, the PSF is a tensor, that turns into a scalar PSF as soon as one picks an excitation (drive field coil) direction and a listening (receive coil) direction. Hence, the resolution in MPI will depend heavily on the choice for these two directions. After some math, one obtains from (12) that the 2D tensor PSF in the xy -plane is

$$\mathbf{h}(\mathbf{x}) = E_T(\mathbf{x}) \cdot \frac{G}{x^2 + y^2} \begin{bmatrix} x^2 & xy \\ -yx & -y^2 \end{bmatrix} \quad (15)$$

$$+ E_N(\mathbf{x}) \left(\begin{bmatrix} G & 0 \\ 0 & -G \end{bmatrix} - \frac{G}{x^2 + y^2} \begin{bmatrix} x^2 & xy \\ -yx & -y^2 \end{bmatrix} \right), \quad (16)$$

where we assumed a gradient field of

$$\mathbf{G}\mathbf{x} = \begin{bmatrix} G & 0 \\ 0 & -G \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (17)$$

and set

$$E_T(\mathbf{x}) := \dot{\mathcal{L}}(\|\mathbf{G}\mathbf{x}\|/H_{sat}), \quad (18)$$

$$E_N(\mathbf{x}) := \frac{\mathcal{L}(\|\mathbf{G}\mathbf{x}\|/H_{sat})}{\|\mathbf{G}\mathbf{x}\|/H_{sat}} \quad (19)$$

to be the tangential and normal envelopes of the PSF respectively. The tangential envelope and normal envelope of the MPI PSF represent different physics. The tangential envelope (good resolution) represents the change in magnitude of the magnetization, while the normal envelope represents change in magnetization due to rotation of the magnetic moments (bad resolution). In Figure 9, these two envelopes are shown. To see how these two

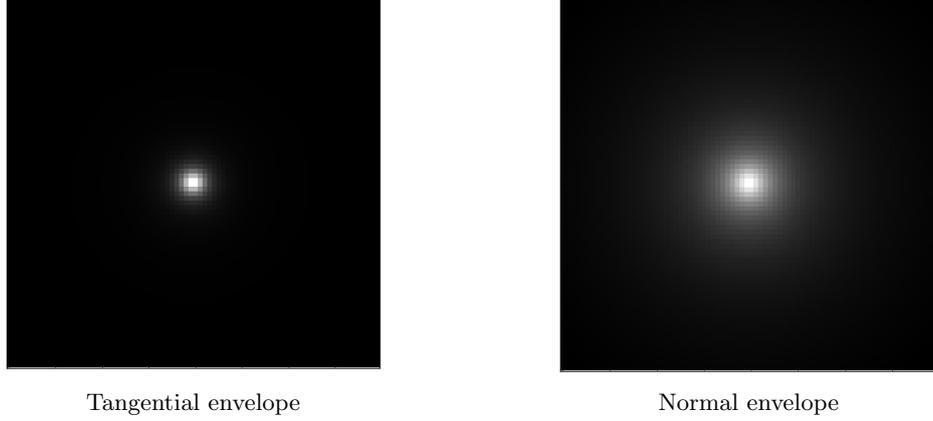


Figure 9: The tangential envelope and normal envelope of the MPI multi-D PSF represent different physics. The tangential envelope (good resolution) represents the change in magnitude of the magnetization, while the normal envelope represents change in magnetization due to rotation of the magnetic moments (bad resolution).

envelopes affect the final scalar MPI PSF, choose the excitation vector $\hat{\mathbf{x}}_s$ and receive coil vector \mathbf{x}_c . Then, according to equation (13)

$$\begin{aligned} h(\mathbf{x}) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{h}(\mathbf{x}) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{Gx^2}{x^2 + y^2} E_T(\mathbf{x}) + \left(G - \frac{Gx^2}{x^2 + y^2} \right) E_N(\mathbf{x}) \\ \implies h(\mathbf{x})/G &= \frac{x^2}{x^2 + y^2} E_T(\mathbf{x}) + \frac{y^2}{x^2 + y^2} E_N(\mathbf{x}). \end{aligned}$$

In Figure 10, we show this expression in terms of 2-D images. The resolution of the PSF is the best in the excitation direction, because the 'bad' normal envelope is suppressed in that direction.

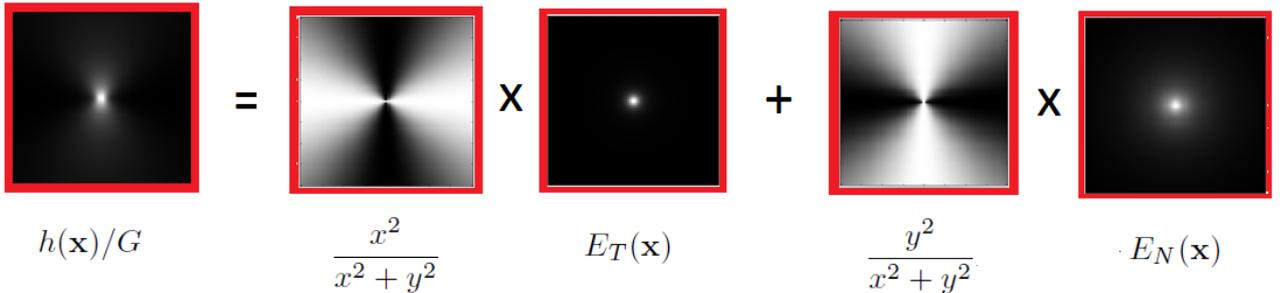


Figure 10: Visual representation of the different components of the point spread function in MPI, when exciting and recording in the x -direction.

The anisotropic nature of the PSF is inconvenient. Therefore, usually two scans are made: one exciting in the x -direction, and one in the y -direction. The corresponding images are then added to yield an image with isotropic resolution. This image is then the true nanoparticle density convolved with the sum of the tangential and the normal envelope:

$$h(\mathbf{x}) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{h}(\mathbf{x}) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{h}(\mathbf{x}) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = E_T(\mathbf{x}) + E_N(\mathbf{x})$$

So the final resolution is dictated by the two functions given in (18) and (19) and their arguments. In the argument, we find the gradient strength G and the unitless constant H_{sat} , which is influenced by the temperature, the material that the SPIONs consist of, and the particle diameter. In [22] it is shown that theoretically, MPI resolution improves linearly with increasing gradient strength, and cubically with increasing particle diameter.

2.4 Field free line MPI

To decrease the image acquisition time in MPI, one can create a field free line (FFL) instead of a field free point (FFP). The benefits were already seen by Weizenecker and Gleich in 2008 [30]. Scanning speed is gained because only two dimensions need to be scanned rather than three, so projection scanning is inherently faster by a factor equal to the number of pixels in the projection direction. Of course, this factor is reduced by the number of angles that we have to project in. Standard results from CT ensure that a 3-D volume can be imaged if a sufficient amount of projections are made. The research groups that use the system matrix MPI approach as well as the x-space MPI research group in Berkeley are exploring the possibilities of FFL MPI. We will regard the x-space approach, and begin with an explanation of the basic principles of x-space projection MPI via field free lines, mostly based on the article by Kunckle et al. [31].

In Figure 11 we see the schematic setup of the necessary coils in a FFL MPI scan. Two opposing NdFeB permanent magnets produce a gradient field that has a field free line along the y -axis. As the FFL needs to scan the entire xz -plane, there are shift coils in both the x and z directions. Typical TX/RX coils are used to excite the SPIONs at 20kHz in the z -direction, and receive the change in magnetization. A 3-D phantom can be placed inside the imaging bore, and is mechanically rotated about the z -axis with angle θ to create projections at different angles. Afterwards, reconstruction techniques similar to those in X-ray CT enables one to reconstruct the distribution of the SPIONs across the 3-D volume.

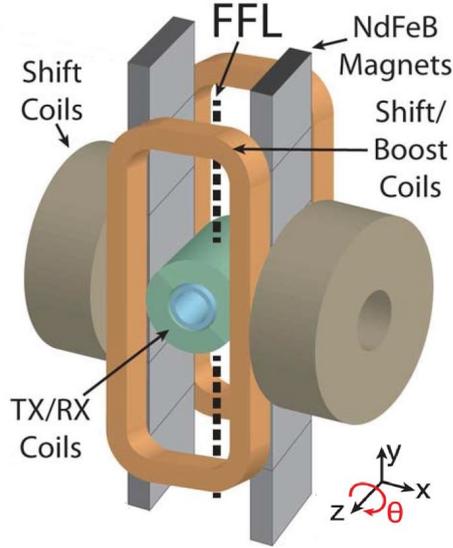


Figure 11: Schematic setup of a FFL MPI scanner. The TX/RX coils are used as receive coils.

As we saw in section 2.3.2, it is crucial for the x-space MPI reconstruction technique to relate the received MPI signal $\mathbf{s}(t)$ to what is physically happening inside the imaging bore. A first step is to mathematically deduce the position of the FFL, dependent on the magnetic fields induced by all the involved coils. Fortunately, a FFL gradient for projection MPI along the y direction can be constructed while obeying Maxwell's equations. The NdFeB coils are chosen to produce the following field:

$$\mathbf{G}\mathbf{x} = \begin{bmatrix} G & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -G \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The total magnetic field inside the imaging bore is

$$\mathbf{H}(\mathbf{x}, t) = \mathbf{G}\mathbf{x} + \begin{bmatrix} H_x(t) \\ 0 \\ H_z(t) \end{bmatrix}. \quad (20)$$

Note that this is the magnetic field with respect to the standard (x, y, z) oriented coordinate system as given in fig. 11. At $\theta = 0$, a sample in the imaging bore is oriented with this coordinate system. As we rotate the sample at a different angle θ , we wish to know where the FFL is located w.r.t. the newly oriented sample. Therefore, we introduce a coordinate system (x', y', z') that rotates with the sample. This rotation can mathematically

be written as

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Conveniently, \mathbf{R} is a unitary matrix so $\mathbf{R}^{-1} = \mathbf{R}^T$. Looking at (20), we can now write the total magnetic field experienced by a rotated sample in the \mathbf{x}' coordinate system as

$$\mathbf{H}(\mathbf{x}', \theta, t) = \mathbf{R} \left(\mathbf{G} \mathbf{R}^T \mathbf{x}' + H_x(t) \hat{\mathbf{i}} + H_z(t) \hat{\mathbf{k}} \right) \quad (21)$$

To determine where the FFL lies in the \mathbf{x}' coordinate system, it is easiest to set $|\mathbf{H}(\mathbf{x}', \theta, t)|^2 = 0$. This yields that the field is zero at

$$z' = \frac{H_z(t)}{G} \quad (22)$$

on the line

$$x' \cos \theta + y' \sin \theta = \frac{H_x(t)}{G}. \quad (23)$$

See [31] for more details. Hence, $H_x(t)$ and $H_z(t)$ allow us to place the FFL anywhere in the xz -plane at an orientation that we influence by the choice of θ . Similar to section 2.3.2, we will now mathematically write down the physical origin of the FFL MPI signal, as done in [31].

In section 2.3.2 we saw that in FFP MPI, the blurred 3-D MPI image originates from the dynamics given in equation (14). For $\theta = 0$, the change of magnetization in the field free line is projected onto the xz -plane, at the position of the FFL. Define

$$\rho_2(x, z) = \int \rho(x, y, z) dy, \quad (24)$$

and recall that the position $\mathbf{x}_s(t)$ of the FFL is also totally defined through its xz -coordinates. Therefore, the projection $\rho_2(\mathbf{x})$ satisfies

$$\tilde{\rho}_2(\mathbf{x}) \propto \rho_2(\mathbf{x}) * * \hat{\mathbf{x}} \cdot \mathbf{h}(\mathbf{x}) \hat{\mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}_s(t)}, \quad (25)$$

where $\mathbf{h}(\mathbf{x})$ is defined as (12), with $\mathbf{G} = \mathbf{G}_2$,

$$\mathbf{G}_2 = \begin{bmatrix} G & 0 \\ 0 & -G \end{bmatrix}.$$

We identify (25) as the 2-D convolution of the PSF with the ideal projection of the nanoparticle density.

Reconstruction of FFL MPI data

For the results in this thesis, we will only consider 2-D images. For this purpose, we don't have to consider the z -axis and simulate 1D projections of a 2-D density function $\rho(x, y)$. If we take N projections at angles θ_i , $i \in \{1, 2, \dots, N\}$, then according to the 1-D variant of equations 24 and 25, these projections P_i are proportional to

$$P(l, i) = \left[\int \rho(x, y) \delta(x \cos \theta_i + y \sin \theta_i - l) dy \right] * h(l), \quad (26)$$

i.e. a projection at angle θ_i convolved with a point-spread function $h(\cdot)$. This 1-D point-spread function can be found by using

$$\mathbf{G} = \begin{bmatrix} G & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -G \end{bmatrix}. \quad (27)$$

in equation (12), at $z = 0$. We do this explicitly in the next section. Now, \mathbf{P} forms the data set after a field free line MPI scan. Such a data set is called a sinogram, and if it is rich enough (i.e. if N is big enough), it can be used to reconstruct the original nanoparticle density $\rho(x, y)$ through a technique called filtered backprojection. As the name already reveals, it consists of a backprojection step and a filtering step. In a continuous setting, that is if we had projected along infinitely many angles θ , we could define the projections \mathbf{P} as a continuous 2-D function $P(l, \theta)$ and the backprojection step computes

$$\hat{\rho}(x, y) = \int_0^{2\pi} P(x \cos(\theta) + y \sin(\theta), \theta) d\theta. \quad (28)$$

In words, the equation above sums the contribution of every line that went through (x, y) when the projections were formed. In practice, the integral in this equation is a sum, as we only have a finite amount of angles θ_i . After the backprojection step, a filtering step is done, because the projection of a value $\rho(\mathbf{x}_0)$ with high density is more often backprojected onto the points in the vicinity of \mathbf{x}_0 than points far away. This effect can be negated by a simple linear ramp filter in Fourier space.

2.5 Parallel MPI

Now that we know how MPI works and what types of reconstruction techniques are currently available, we can see if we can design an MPI system that fulfills all the requirements given in section 2.1.1. Let us go by them one by one.

1. **It is possible to measure the signal at different locations.** As in MRI, the source of the received signal lies within the body. The source is a change in magnetization of the SPIONs and can be picked up by a coil at any location around the body. Hence, this requirement is fulfilled.
2. **There is room for undersampling while still scanning each location of the object.** Field free point MPI fails in this respect. As each point in the body has to be addressed to extract the nanoparticle density at that location, undersampling would simply mean that we will never know the nanoparticle density at the spots we did not visit with the field free point.
3. **Except for sensitivity differences, each receive channel has to receive the same signal.** Again, this fails for field free point MPI, as the coils that are placed in the excitation direction yield an image that contains fundamentally different information than coils that are perpendicular to the excitation direction..

So, requirements 2 and 3 fail for traditional field free point MPI. We now give the two key insights that enable us to do parallel MPI, using a field free line.

Key insight 1

The main reason that requirement 2 is fulfilled in MRI, is that data acquisition in MRI occurs in Fourier-space (k-space). In k-space, undersampling yields only aliasing, as opposed to the deletion of essential information that would occur in the x-space undersampling for field free point MPI. In parallel MRI, the aliasing is then unraveled using the extra spatial information of each parallel coil. Interestingly, if we look at Field Free Line MPI, the Fourier Slice Theorem tells us that FFL MPI also samples in k-space:

Theorem 2.1 (Fourier slice theorem). *Let F_1 and F_2 be the 1-D and 2-D Fourier transform operators respectively, and P_1 be the projection operator, that projects a 2-D function onto a line, i.e. sums the values along one dimension. Finally, let S_1 be an operator that makes a slice through the origin of a 2-D function, perpendicular to the projection lines. Then,*

$$S_1 F_2 = F_1 P_1$$

Hence, undersampling in FFL MPI can be done without necessarily deleting essential information. In this context, undersampling means taking less projection angles. From that, one could also make a second argument why FFL MPI fulfills requirement 2: although taking less projection angles, we still visit each part of the body, though less often.

Key insight 2

Next, we make the most fundamental insight that proves parallel MPI is realizable. Here, knowledge of MPI physics was indispensable. The goal is to prove that, opposed to field free point MPI, in field free line MPI we can record the same change in magnetization of the SPIONs with coils placed at different locations. To do so, we have to dig into the physics, and make use of sections 2.3.2 and 2.4 .

In FFL MPI, the gradient field is

$$\mathbf{G}\mathbf{x} = \begin{bmatrix} G & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -G \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (29)$$

which implies that the field free line will be along the y -axis as in Figure 11. Now, as we saw in section 2.3.2, the point spread function in 3D x-space MPI is a 3x3 matrix-valued function that turns into a scalar-valued function as soon as one picks a “listening” vector and an excitation vector. Recall from equation (14), that for the gradient field given in (29), the matrix PSF will be

$$\mathbf{h}(\mathbf{x}) = E_T(\mathbf{x}) \cdot \frac{G}{x^2 + z^2} \begin{bmatrix} x^2 & 0 & xz \\ 0 & 0 & 0 \\ -xz & 0 & -z^2 \end{bmatrix} + E_N(\mathbf{x}) \left(\begin{bmatrix} G & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -G \end{bmatrix} - \frac{G}{x^2 + z^2} \begin{bmatrix} x^2 & 0 & xz \\ 0 & 0 & 0 \\ -xz & 0 & -z^2 \end{bmatrix} \right), \quad (30)$$

where $E_T(\mathbf{x}) = \hat{\mathcal{L}}(\|\mathbf{G}\mathbf{x}\|/H_{sat})$, and $E_N(\mathbf{x}) = \frac{\mathcal{L}(\|\mathbf{G}\mathbf{x}\|/H_{sat})}{\|\mathbf{G}\mathbf{x}\|/H_{sat}}$ are the tangential and normal envelopes of the PSF.

Let us now choose the excitation and receive coil vectors. As an excitation direction, we pick the x -direction, i.e. $[1 \ 0 \ 0]$. We place the multiple receive coils in the xy -plane at $z = 0$, so that their positions are given by $[\cos(\theta), \sin(\theta), 0]$. The resulting PSF is then

$$h(x, z) = [\cos(\theta), \sin(\theta), 0] \cdot \mathbf{h}(\mathbf{x}) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = G \cos(\theta) \left[E_T(x, z) \frac{x^2}{x^2 + z^2} + E_N(x, z) \left(1 - \frac{x^2}{x^2 + z^2} \right) \right] \quad (31)$$

Hence, the images received by the different coils in the xy -plane are only different by a factor! This proves that FFL MPI with this coil setup fulfills requirement 3. Another nice observation here, is that if we place the center of the receive coils at $z = 0$, the normal envelope of the point spread function is multiplied by zero. Hence, FFL MPI gives a better resolution in the xy -plane, than FFP MPI. In the z -direction the normal envelope does play a role, of course. In Figure 12, two examples of receive coil setups for parallel MPI are given. The amount of receive coils has to be chosen carefully, influenced by practical hardware issues, sensitivity simulations and computational complexity. We talk more about this in section 6.

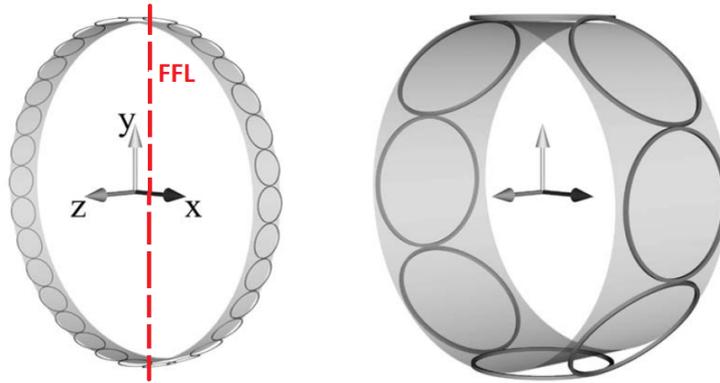


Figure 12: Two illustrative receive coil setups, with FFL orientation. The excitation of the FFL will be in the x -direction.

Hardware setup

We will run simulations and show results for one hardware setup, that we deem best for now. In future research, the reconstruction qualities of several setups can be investigated and optimized, which can be a whole new project on its own.

We choose to place six coils in the xy -plane at $z = 0$; we saw in equation (31) that parallel MPI is feasible in that situation. Secondly, we will not place any coils at angles $\theta = 90$ and $\theta = 180$ degrees, as the same equation tells us that these coils will yield zero signal and hence will deliver no information to us. Now, as we just wish to show the power of parallel MPI in this thesis, we choose the simplest yet powerful setup shown in Figure 13.

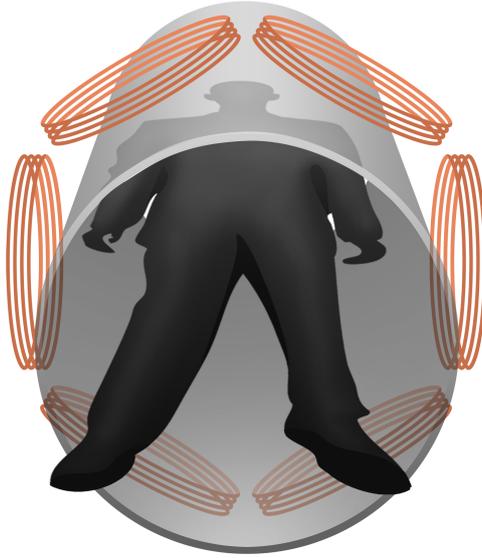


Figure 13: The receive coil setup we will consider in this project. The z -axis is along the bore, and the centers of the six receive coils are placed at $z=0$. The x -axis runs from left to right in this image, and the field free line will be parallel to the y -axis, exciting in the x -direction. In this way, an axial slice of the body can be imaged with high resolution in the xy -plane.

This hardware setup implies that we have six sensitivity maps S_i , shown in Figure 14. To calculate these sensitivity maps, we used a numerical Biot-Savart law simulator. The Biot-Savart law governs the spatial distribution of the magnetic field strength induced by a current through a wire (in our case a coil) [39]. By the reciprocity principle, this also governs the current induced by a change in magnetic field strength at different spatial locations, which yields the sensitivity maps in Figure 14.

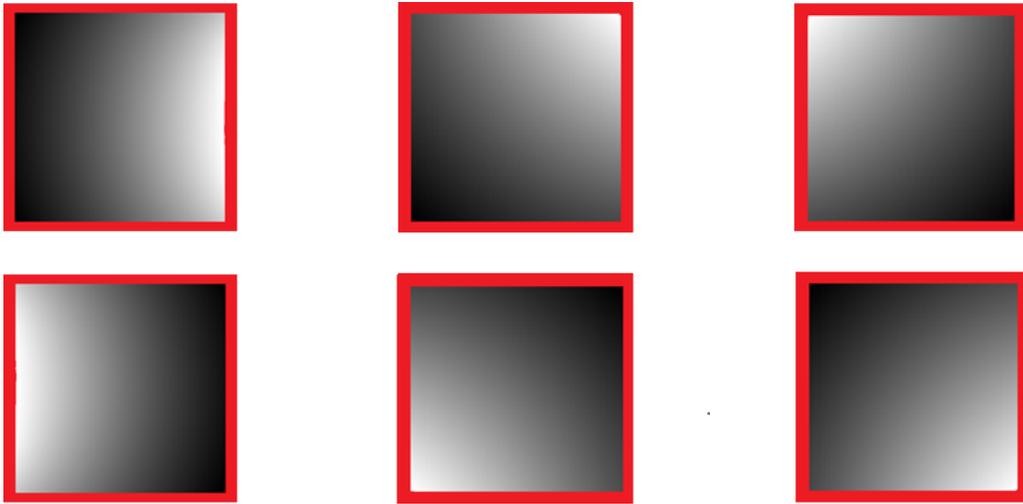


Figure 14: The six sensitivity maps that are inherent to the hardware setup shown in Figure 13. Here, the sensitivity maps are shown on a logarithmic scale so that the structure is better visible.

2.6 The parallel FFL MPI physics simulator

Currently, there is no MPI scanner in the world with a parallel coil hardware setup. Hence, in the time scope of this thesis, it is impossible to work with real data. To generate synthetic data, the easiest would be to use our operator E , that we will define later in equation (36). This can be a good first step, but will always yield better results than one can expect in real life. This is because the algorithm tries to find the best $\hat{\rho}$ such that $E\hat{\rho} = d$, and if d was created using $E\rho$ we can even expect to find back the exact solution $\hat{\rho} = \rho$. In real life however, the operator E is just an approximation of what physically happens, so that $E\rho$ will always differ from the actual data, even if there would be no noise involved.

Therefore, to really validate our model and reconstruction techniques, we have built a parallel FFL MPI simulator in Python. It takes the true nanoparticle density ρ as an input, after which it places the field free line over the first column of the matrix ρ , and records the change in magnetization when the field free line is moved to the next column. This mimics excitation in the x -direction, which is part of our chosen strategy from section 2. The recorded change in magnetization over the entire volume is then gridded onto the correct box in the sinogram. All the parameters, like gradient field strength and nanoparticle diameter are chosen at fixed, common values. To vary these parameters would not yield relevant information for this project. After forming the sinogram, we can use either filtered backprojection or parallel imaging to find $\hat{\rho}$. For the reader's illustration, we show the output of the simulator when ρ is the Cal phantom (logo of the University of California in Berkeley), and its reconstruction in Figure 17. Note that the output in (c) is blurred, inherently by MPI physics. We do see that resolution is still pretty good, which reflects the fact that FFL MPI gets rid of the 'bad' normal envelope of the point spread function in the xy -plane, like we showed in the previous section.

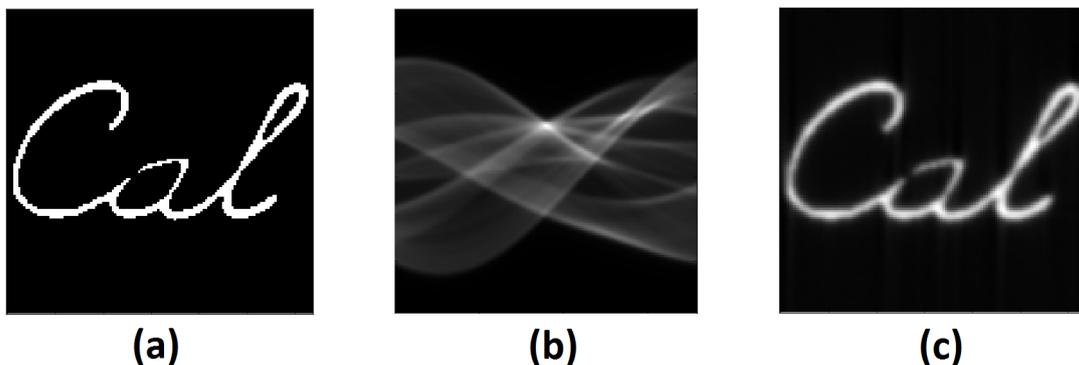


Figure 15: The sinogram (b) is formed after the cal phantom (a) has been fed to the FFL MPI simulator, using 180 projection angles. (c) is the filtered backprojection reconstruction. Note that it is blurred by the MPI physics.

3 Inverse problem formulation.

In section 2, we saw that parallel MPI is physically feasible. The next step, is to exploit the extra spatial information that we get from the multiple coils. We can learn from the two examples in section 2.1 that it makes sense to take a forward model approach. After all, by theorem 2.1 the Fourier transforms of the MPI projections can be understood as radially sampled spokes through the origin of the 2-D Fourier transform of the nanoparticle density. Undersampling in the number of spokes will yield untraceable aliasing, which is why a direct reconstruction approach will fail.

3.1 Notation and basics

In mathematics, an image u is treated as a function that maps values onto a domain Ω :

$$u : \Omega \rightarrow \mathbb{R}.$$

Now, whenever one wishes to visualize these functions, the domain Ω will have to be discretized, and u will be a matrix. For $\Omega \in R^N$, we define

$$U_{i_1, \dots, i_N} = \frac{\int_{\text{pixel}(i_1, \dots, i_N)} u(x) \, dx}{\int_{\text{pixel}(i_1, \dots, i_N)} 1 \, dx} \quad (32)$$

In every imaging modality, the received data of a scan - may it be a click on the button of your camera or an MRI scan - is used to form an image. How the transition from the received data to the eventual image takes place, is determined by the physical properties of the imaging system that makes the scan. In most modalities, this transition from signal to image can be described by an operator, i.e.

$$Eu = d \tag{33}$$

where d is the received data, u is the desired image and E is the operator that describes the relation between the signal and the image. Equation (33) is called the forward model the imaging system. An intuitive approach to solve u from (33) is to multiply both sides by the E^{-1} , i.e. the inverse operator. But here two main problems arise:

1. The inverse operator is not defined.
2. The received data d might not be equal to the theoretically expected data, due to for example noise differences caused by properties of the hardware in the imaging system. It is a hard task to nicely extract u from such a signal.

Problem 1 occurs in this thesis, and a good example of problem 2 occurs when one tries to reconstruct u when E is a convolution operator, i.e.

$$d = Eu = h * u. \tag{34}$$

Where h is the so called point spread function (PSF). From the convolution theorem we know that

$$\mathcal{F}(d) = \mathcal{F}(h * u) = \mathcal{F}(h) \cdot \mathcal{F}(u).$$

And thus

$$u = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(d)}{\mathcal{F}(h)} \right). \tag{35}$$

So we see that we have a problem when $\mathcal{F}(h)$ is (close to) zero. Noisy signals often contain high frequencies, whereas the point spread function usually contains little. Therefore, we see that the higher frequency parts of d are amplified using this direct Fourier inversion method, which explains the noise-amplifying and thus problematic property of this method. In section 3.3, we will look for ways to work around problems 1 and 2. First, we formulate the forward model for parallel field free line MPI.

3.2 Forward model formulation for parallel MPI

In field free line MPI, the desired image is a blurred 2-D nanoparticle density distribution ρ , that we aim to reconstruct from projection data d . We have to be careful in the definition of our forward operator E , because it is not our goal to deconvolve the blurred density with the point spread function. Hence, we will only put projection operations into E , and no convolutions. So from now on we will aim to reconstruct $\tilde{\rho}$, being the Langevin-blurred version of the true nanoparticle density distribution ρ .

Now, let L be the number of coils used in the data acquisition. The data set d after a parallel field free line MPI scan, will consist of L sinograms P_i , $i \in \{1, 2, \dots, L\}$. Let S_i be the sensitivity map of coil i , and R be the *Radon transform*, that performs the discrete version of equation (26). We can then write

$$E\tilde{\rho} = [RS_1\tilde{\rho}, RS_2\tilde{\rho}, \dots, RS_L\tilde{\rho}]^T = d, \quad (36)$$

so we will regard our data set d as a multidimensional array of size $N \times M \times L$. N is the number of projection angles and M is usually chosen to be about $n\sqrt{2}$ if we aim to reconstruct a $\tilde{\rho}$ of size $n \times n$. This choice follows from the Pythagorean theorem, as we have to be able to project along the diagonal of the square image.

3.3 Image reconstruction via minimization

In section 3.1 we saw that in general, problems arise in extracting the desired image from a data set. To work around this, we can rewrite the forward model to a minimization problem, and use algorithms that aim to get as close to the true solution as possible. Ideally, we can also include prior knowledge of the solution in this minimization problem. The objective function is made convex such that a minimum is guaranteed, and as a first step we consider:

$$\hat{u} = \arg \min_{u \in X} F(u) := \|Eu - d\|_2^2 \quad (37)$$

Setting $\frac{dF}{du} = 0$ yields $Eu = d$, which proves that the argmin of F is the desired image. Also, $\|Eu - d\|_2^2$, is the squared L^2 norm of the distance between Eu and d , makes sure that Eu is close to d . Conformly, this term is called the data fidelity term. Now, as in section 3.1, significant noise will cause problems if we try to find the minimum of F through an algorithm. Therefore, we can choose to extend F with a second term called a regularization term, that guides the algorithm in the right direction. We will talk more about this in section 4. Besides the forward operator, the algorithm needs another operator to find the minimum of the objective function: the adjoint operator E^* . Let $E : X \rightarrow Y$, then the adjoint operator is defined as the operator that satisfies the equation

$$\langle Ep, q \rangle_Y = \langle p, E^*q \rangle_X, \quad (38)$$

for all possible $p \in X$ and $q \in Y$. Here $\langle \cdot, \cdot \rangle_X$ and $\langle \cdot, \cdot \rangle_Y$ are the innerproducts on X and Y respectively. The adjoint operator is in practice more often defined and easier to find than the inverse operator E^{-1} . The reason that a minimization algorithm needs this operator, can be seen if we would analytically try to solve equation (37).

$$\frac{dF}{du} = E^*(Eu - d) = 0 \quad (39)$$

$$(E^*E)\rho = E^*d \quad (40)$$

We see two things here. Firstly that the adjoint operator plays a role in calculating the derivative of the objective function as we see in (39). Secondly, we see that if we can find E^* , we can solve (40) instead of (37). This can be convenient, because there are very efficient algorithms, e.g. the conjugate gradient algorithm, that exploit the fact that (E^*E) is self-adjoint and positive definite. Therefore, we need to find the adjoint operator E^* explicitly.

3.4 The adjoint operator for parallel MPI

To find the adjoint to the forward operator defined in equation (36), we can use the definition of the adjoint operator. The first observation we make, is that for the inner product in (38) to be defined, we need $E^* : Y \rightarrow X$, with $\rho \in X$ and $d \in Y$. Furthermore, we use the following: let $A, B : X \rightarrow Y$ be two operators. Then for all $x \in X$ and $y \in Y$, by definition of the adjoint operator we have

$$\langle ABx, y \rangle = \langle Bx, A^*y \rangle = \langle x, B^*A^*y \rangle.$$

So $(AB)^* = B^*A^*$. Hence, with E given in (36), for each RS_i we use $(RS_i)^* = S_i^*R^*$ to find E^* . If L receive coils are used, we obtain

$$E^*d = [S_1^*R^*, S_2^*R^*, \dots, S_L^*R^*] \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_L \end{bmatrix} \quad (41)$$

$$= \sum_{i=1}^L S_i^*R^*d_i, \quad (42)$$

where $S_i^* = S_i$ as it only performs a pointwise, real matrix multiplication. Furthermore, it is well known that backprojection is the adjoint operator of the Radon transform R . So, for R^* we use equation (26). The backprojection operation transforms each d_i into an $n \times n$ matrix, yielding L matrices. The pointwise multiplication with the sensitivity maps followed by the summation in (42) yields one final matrix of size $n \times n$, which is the desired size.

4 Convex optimization and regularizers

4.1 Convex functions

As discussed in section 3.3, the reconstruction problem for parallel MPI can be written as a minimization problem. Whenever an analytical solution to such a minimization problem is unavailable, numerical techniques have to be used. To guarantee the existence of a solution, it is essential that the objective function is convex. Let us start this section with the very definition of a convex function.

A function $F : X \rightarrow \mathbb{R}$ is called convex if for all $x_1, x_2 \in X$ and $\lambda \in [0, 1]$

$$F(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda F(x_1) + (1 - \lambda)F(x_2).$$

F is strictly convex if the inequality is strict. Convex functions have the nice property that every one of its minima is a global minimum, and if they are strictly convex, the minimum is unique. As we already saw in equation (37), an image reconstruction problem can often be written as a minimization of a convex objective function. In this chapter, we will first extend the objective function to our needs, and do a quick review on the theory of convex minimization that is most relevant for this project. With this knowledge, we will investigate algorithms that can compute the minimum of the convex objective function.

4.2 Regularization

In addition to the data fidelity term in (37), a regularization term can be included in the objective function to penalize certain undesired properties of the solution \hat{u} , and to improve the condition of the minimization problem. Most of the time, these two go hand-in-hand. In the objective function, one places a factor λ to adjust the severity of the regularization.

$$\hat{u} = \arg \min_{u \in X} F(u) := \underbrace{\lambda H(u)}_{\text{Data fidelity}} + \underbrace{J(Au)}_{\text{Regularization}}. \quad (43)$$

Before minimizing (43), a specific regularization term J has to be chosen. This is quite an important choice, because it imposes restrictions and properties on the solution \hat{u} . We will define the most important spaces in which these solutions can live and their (dis)advantages in section 4.2.1. We now start by regarding two examples that entail well-known spaces on continuous domains, so that the reader gets a feeling for what formulation (43) can do.

The data fidelity term we will regard is

$$H(u) = \frac{\lambda}{2} \|Eu - d\|_2^2,$$

which is a distance measure between Eu and d and minimizing this is the most fundamental requirement; if we leave $H(u)$ out of the objective function, we end up with a solution that has no connection to the data d at all. Note that we need $u \in L^2(\Omega)$ in order for $H(u)$ to be finite. Fortunately, $L^2(\Omega)$ is a very rich space, and we thus have a large pool of images to obtain a solution from. For the regularization term, a first choice would be $J(u) = 0$. To give an illustration of how this works in the framework of (43), we regard our convolution problem in (34). We have

$$\hat{u} = \arg \min_{u \in L^2(\Omega)} \frac{1}{2} \|h * u - f\|_2^2. \quad (44)$$

Taking the derivative w.r.t. u together with Plancherel's theorem and the convolution theorem yields that we end up with the same solution \hat{u} and thus problems as in (35). Hence, we must use a nonzero regularization term. Let us set $J(u) = \frac{1}{2} \|u\|_2^2$; as this is finite for all $u \in L^2$, we preserve the large amount of images that we can extract a solution from. So, we have an explicit minimization problem:

$$\hat{u} = \arg \min_{u \in L^2(\Omega)} \cdot \underbrace{\frac{\lambda}{2} \|h * u - f\|_2^2}_{\text{Data fidelity}} + \underbrace{\frac{1}{2} \|u\|_2^2}_{\text{Regularization}}. \quad (45)$$

Which, by convexity, has a unique solution, and it is

$$\hat{u} = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(h)} \mathcal{F}(f)}{|\mathcal{F}(h)|^2 + \frac{1}{\lambda}} \right). \quad (46)$$

We see that we no longer face the problem of division by zero. But still, we are using a penalty on the L^2 -norm of u , which does not necessarily have a physical meaning. So, when this penalty is severe (small values of λ), we will experience unwanted properties in our solution such as contrast loss. Moreover, noise is admitted in

$L^2(\Omega)$, which makes it unsuited for denoising purposes.

Hence, we look for a space in which we have an improved condition of the problem as in (46), but at the same time improve the signal to noise ratio (SNR) of \hat{u} . Noisy signals contain a lot of small jumps in signal intensity, and hence high derivatives. Therefore, we introduce $J(u) = \frac{1}{2} \|\nabla u\|_2^2$ to penalize derivatives, and end up with the minimization problem

$$\hat{u} = \arg \min_{u \in H^1(\Omega)} \underbrace{\frac{\lambda}{2} \|h * u - f\|_2^2}_{\text{Data fidelity}} + \underbrace{\frac{1}{2} \|\nabla u\|_2^2}_{\text{Regularization}}. \quad (47)$$

Using directional derivatives and Plancherel's theorem, one can deduce that there is a unique solution to this minimization problem:

$$\mathcal{F}(\hat{u})(\omega) = \frac{\overline{\mathcal{F}(h)(\omega)} \mathcal{F}(f)(\omega)}{|\mathcal{F}(h)(\omega)|^2 + \frac{1}{\lambda} |\omega|^2}. \quad (48)$$

The inverse Fourier transform gives then the best approximation \hat{u} to the true image. We see that division by zero is avoided and the high frequency parts are explicitly damped, so noise is reduced instead of amplified.

In Figure 16, we show the results of L^2 and H^1 regularization on a 1-D image that is subject to a convolution with a Gaussian kernel, and white noise. We omit the plot here, but we want to stress that without regularization, the deconvolution yields a completely blown-up signal that contains no information about the true image.

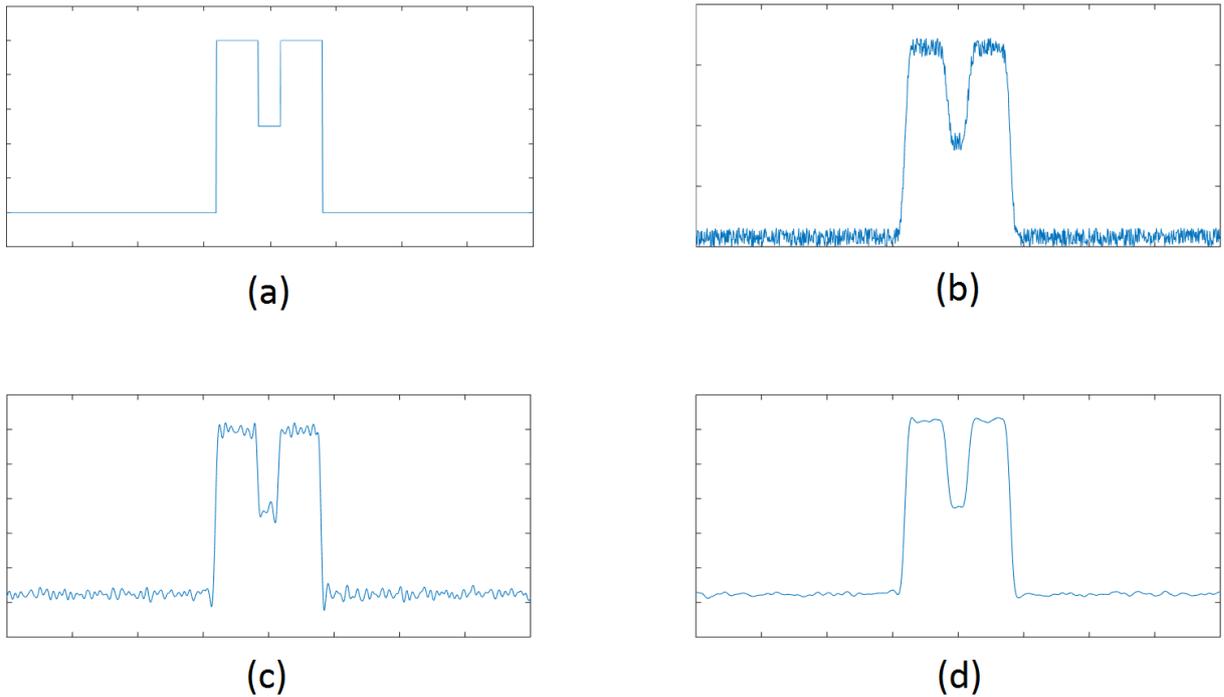


Figure 16: In (b) we see the signal in (a), convolved with a Gaussian kernel. Also, we added white noise with standard deviation $\sigma = 0.1$. In (c) and (d) we plotted the solutions to the deconvolution problem as given by (46) and (48) respectively. We see that the H^1 penalty is better at reducing noise. However, the smoothing behavior reduces resolution whereas L^2 does a slightly better job in keeping the sharp edges in place.

4.2.1 Spaces of images

The two examples in the previous section have shown us that regularization can help to prevent noise blow-ups in reconstruction techniques, but that caution is needed. Namely, we saw that the choice of the regularization term affects the space X in which we look for a solution. Firstly because we need the regularization term to be finite. Secondly, when we derive the optimality condition for the solution of the minimization problem, we can derive certain properties of the solution. More on this follows in section 4.2.2.

The relation between this requirement and the space X is found in the very definitions of these spaces. We start with the Lebesgue spaces L^p . For every $1 \leq p < \infty \in \mathbb{R}$ we have

$$L^p(\Omega) := \left\{ u : \Omega \rightarrow \mathbb{R} : \int_{\Omega} |u|^p dx < \infty \right\}. \quad (49)$$

Note that Ω can be any subset of \mathbb{R}^n . Another important space is

$$L^\infty(\Omega) := \left\{ u : \Omega \rightarrow \mathbb{R} : \operatorname{ess\,sup}_{x \in \Omega} u(x) < \infty \right\} \quad (50)$$

because we can denote a function's maximum through the L^∞ norm. Every Lebesgue space with $p \geq 1$ is a Banach space with norm

$$\begin{aligned} \|u\|_{L^p} &= \left(\int_{\Omega} |u|^p dx \right)^{\frac{1}{p}} \\ \|u\|_{L^\infty} &= \operatorname{ess\,sup}_{x \in \Omega} u(x) \end{aligned}$$

. For $p = 2$, L^p is a Hilbert space, defined by the inner product

$$\langle u, v \rangle_{L^2} := \int_{\Omega} u \cdot v dx. \quad (51)$$

A motivation for choosing a Lebesgue spaces as an image space, is that they are very large. For example, images with a countable number of discontinuities are allowed, which is great because every edge in an image is characterized by discontinuities. Moreover, we know that for $p > 1$, each Lebesgue space is a dual space, since

$$L^p(\Omega) = (L^q(\Omega))^* \iff \frac{1}{p} + \frac{1}{q} = 1. \quad (52)$$

As we shall see in section 4.3, dual spaces are especially nice in the sense of minimization problems for convex functionals.

Yet, Lebesgue spaces have their drawbacks. This is mainly due to the fact that they are not able to clearly distinguish between noise and signal. That is, the L^p norm of a signal with added Gaussian noise is not necessarily larger than that of the original signal. Therefore, we look for normed subspaces of L^p that are in fact able to detect noise through their norm. An obvious consideration is to look at the Sobolev spaces that involve the first-order (weak) derivatives:

$$W^{1,p}(\Omega) = \left\{ u \in L^p(\Omega) : \int_{\Omega} |\nabla u|^p dx < \infty \right\}. \quad (53)$$

So we see immediately that $W^{1,2}(\Omega) \subset L^2(\Omega)$. The restriction is caused by a property that we like for our imaging purposes: the weak derivative ∇u has to be p -integrable. This allows us to distinguish noise from signal because the Sobolev seminorm

$$|u|_{W^{1,p}}^p = \int_{\Omega} |\nabla u(x)|^p dx \quad (54)$$

of an image that is subject to noise is relatively large. Therefore, it is smart to use this Sobolev seminorm as the regularization term J , to create a penalty on noisy images. We will now dig a little deeper into the effects that these choices for J have on the properties of the solution of (43), see if disadvantages arise and how we can adapt to them.

4.2.2 Properties of the solution to the objective function.

As we saw, the fact that the regularization term should be finite already gives an insight in which space we look for our solution \hat{u} . In this section, we will show that the optimality condition for (43) gives a second and more

precise description of the properties of \hat{u} . Subsequently, we suggest a regularization term that does a better job at keeping sharp edges in place.

To give an illustration, let us start with the most simple case

$$\hat{u} = \arg \min_{u \in L^2(\Omega)} F(u) = \arg \min_{u \in L^2(\Omega)} \frac{\lambda}{2} \|Ku - d\|_2^2 + \frac{1}{2} \|u\|_2^2.$$

Where K is an integral operator, such as the convolution in (34). The optimality condition then yields

$$\begin{aligned} 0 &= F'(u) = \lambda K^*(Ku - d) + u \\ \implies u &= \lambda K^*(d - Ku) \end{aligned}$$

So u is in the range of K^* , which is again an integral operator. This implies that \hat{u} will be smooth and contains no sharp edges. So even for L^2 regularization on a signal that contains very little noise, we might not end up with the solution we want.

Next, for noisy signals we wish to use a penalty on the derivatives:

$$\hat{u} = \arg \min_{u \in H^1(\Omega)} F(u) = \arg \min_{u \in H^1(\Omega)} \frac{\lambda}{2} \|Ku - d\|_2^2 + \frac{1}{2} \|\nabla u\|_2^2.$$

The optimality condition yields

$$0 = F'(u) = \lambda K^*(Ku - d) + \nabla^* \nabla u \quad (55)$$

$$-(\nabla^* \nabla)u = \lambda K^*(Ku - d) \quad (56)$$

For simplicity consider $K = I$, and recall that $\nabla^* = -\text{div}$ and thus $\nabla^* \nabla = -\Delta$. Hence, (56) simplifies to

$$\Delta u = \lambda(u - d), \quad u - d \in L^2(\Omega)$$

which is an elliptic partial differential equation (PDE). From elliptic regularity [26], we know that the solution to this PDE will live in $W^{2,2}(\Omega)$ and thus be oversmoothed. This analysis gives rise to a key question: can we design a regularization functional $J(u)$ that penalizes derivatives and at the same time allows for solutions \hat{u} to (43) that contain sharp edges?

To answer this question, we look at the properties of a more general regularization term

$$J(u) = \frac{1}{p} \int_{\Omega} |\nabla u|^p dx. \quad (57)$$

Then, the optimality condition yields

$$0 = F'(u) = \lambda K^*(Ku - d) - \nabla \cdot (|\nabla u|^{p-2} \nabla u). \quad (58)$$

The Euclidean norm of $|\nabla u|^{p-2} \nabla u$ is $|\nabla u|^{p-1}$. So when $p > 1$, ∇u will have such a big impact on the value of $\nabla^* (|\nabla u|^{p-2} \nabla u)$ for images u that contain large “gradients”, that the data fitting is too heavily distorted. Remember that very large gradients are present at the edges of images. Therefore, for $p > 1$ the solution to (58) will not contain any sharp edges even though the data fidelity term tells us so. Interestingly, we see a shift in this behavior when $p \downarrow 1$. The Euclidean norm of $|\nabla u|^{-1} \nabla u$ is equal to 1, no matter how large ∇u . Hence, for $p = 1$, images with large (infinite) “gradients” can still be a solution to (58). We have to be careful, though. The reason we write “gradients”, is that for functions that contain discontinuities, the gradient is not defined. Therefore, we introduce the concept of total variation.

4.2.3 Total variation

The total variation (TV) of a function is defined as

$$TV(u) := \int_{\Omega} |\nabla u| dx, \quad (59)$$

which makes sense only if $u \in W^{1,1}(\Omega)$. Now, piecewise constant functions that contain discontinuities, are not in $W^{1,1}(\Omega)$. Therefore, we consider a larger space: the space of functions of bounded variation $BV(\Omega)$. To define this space, we must first give the exact definition of the total variation:

$$TV(u) := \sup_{\substack{g \in C_0^\infty(\Omega; \mathbb{R}^d) \\ \|g\|_\infty \leq 1}} \int_{\Omega} u \nabla \cdot g dx \quad (60)$$

For now we use

$$\|g\|_\infty := \operatorname{ess\,sup}_{x \in \Omega} \sqrt{g_1(x)^2 + \dots + g_d(x)^2},$$

but variants of this choice are possible as well, leading to different properties of solutions. The space of functions of bounded variation is then defined as

$$BV(\Omega) := \{u \in L^1(\Omega) : TV(u) < \infty\} \quad (61)$$

i.e. the space of functions for which the total variation is well-defined. A short summary: We saw that for $p \downarrow 1$, the solutions to the convex minimization problem

$$\min_u \quad \frac{\lambda}{2} \|Eu - d\|_2^2 + \frac{1}{2} \int_\Omega \frac{1}{p} |\nabla u|^p \, dx$$

are not oversmoothed. Hence we like to use $p = 1$, but the gradient operator ∇ is only defined for $u \in W^{1,1}(\Omega)$, which does not allow for sharp edges. Therefore, we defined the concept of total variation, since $TV(u)$ is finite for piecewise constant functions u , and is equal to $|\nabla u|$ for $u \in W^{1,1}(\Omega)$. So, we look for the minimum of the convex functional

$$F(u) = \frac{\lambda}{2} \|Eu - d\|_2^2 + TV(u) \quad (62)$$

which has a solution \hat{u} in the space $BV(\Omega)$, i.e. the space of absolutely integrable functions that have finite total variation. It is an extension of $W^{1,1}(\Omega)$ with the nice property that it contains piecewise constant functions. In the next sections, we will introduce the concept of convex minimization and define an algorithm to actually find the minimum of (62).

4.3 From theory to practice

In this project, we are going to consider the following objective functions:

$$\min_u \lambda \|Eu - d\|_2^2 + \|Au\|_2^2, \quad A = 0, \text{ } I \text{ or } \nabla, \quad (63)$$

$$\min_u \lambda \|Eu - d\|_2^2 + \|\nabla u\|_1 \quad (64)$$

For (63), the optimality condition yields that it is a necessity to solve

$$(\lambda E^*E + A^*A)u = \lambda E^*d,$$

which we will do using the conjugate gradient algorithm that we introduce in section 4.4.1 as the operator $(\lambda E^*E + A^*A)$ is self-adjoint. For the objective function in (64) with the L^1 norm present, we have to put in some more effort. The optimality condition yields

$$(\lambda E^*E)u - \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = E^*d,$$

which brings about two problems. Firstly, we identify a possible division by zero. Secondly, if one would somehow circumvent this division by zero (for example by adding a small number ϵ), we are again left with a PDE that has smooth solutions, while we wish to preserve discontinuities (edges). There are two ways to proceed. One is to approximate $J = |\nabla u|$ by $J_\epsilon = \sqrt{(\nabla u)^2 + \epsilon}$, because it tends to favor diffusion along edges rather than across, so that edges are smoothed out less than other structures [42]. Most of the time, it performs sufficiently well but it is a trick and an approximation. We take the second, more general and accurate approach. It makes use of the *Fenchel duality theorem*. This states that to every primal problem

$$P = \min_{u \in X} H(u) + J(Au), \quad (65)$$

there exists a dual problem

$$D = \max_{p \in Y^*} -H^*(A^*p) - J^*(-p) \quad (66)$$

such that $P = D$. Note that $H : X \rightarrow [0, \infty]$ and $J : Y \rightarrow [0, \infty]$ are assumed to be convex functions. The functions H^* and J^* that are found in the dual problem, are called the *convex conjugates* of H and J . They can be found explicitly, and are defined as follows. Let F be a convex function of a vector $z \in Z$, then the convex conjugate is defined as the Legendre transform of F :

$$F^*(\tilde{z}) = \max_z \{ \langle z, \tilde{z} \rangle - F(z) \}. \quad (67)$$

The original function F can be recovered by applying conjugation again:

$$F(z) = \max_{\tilde{z}} \{ \langle \tilde{z}, z \rangle - F^*(\tilde{z}) \}. \quad (68)$$

Now, to see why this helps us solve (64), we will circumvent the problems with the minimization of the L^1 term by substituting it with its convex conjugate, to create a *saddle point problem*. By definition,

$$J(Au) = \max_p \{ \langle Au, p \rangle - J^*(p) \},$$

so we can rewrite (65) into the saddle point problem

$$\min_{u \in X} \max_{p \in Y^*} S(u, p) := \langle Au, p \rangle + H(u) - J^*(p). \quad (69)$$

In section 4.4.2, we introduce a primal dual method that solves (69). It uses an optimality condition based on the concept of subdifferentials, and the difficult L^1 minimization we had in the primal problem transforms into a simple projection onto a convex set in the dual space, characterized by one update rule in the algorithm.

Subdifferentials and the optimality condition

To obtain the solution to (69), we need to know which properties this solution should have. Usually, the optimality condition gives us this property and is found by setting the derivative of the objective functional equal to zero. But, since the functional might not be differentiable in the classical sense, we must rely on the concept of subdifferentials.

Let $F : X \rightarrow \mathbb{R}$ be a convex functional and X^* be the dual space of X . Then, the subdifferential of F at $u \in X$ is defined as

$$\partial F(u) = \{p \in X^* : \langle p, v - u \rangle \leq F(v) - F(u), \forall v \in X\} \quad (70)$$

where $\langle \cdot, \cdot \rangle$ denotes the standard duality product between X and X^* . An element $p \in \partial F$ is called a subgradient and can be identified with the slope of a plane in $X \times \mathbb{R}$ through $(u, F(u))$ that lies under the graph of F . From the definition we can see immediately that \hat{u} is a minimizer of F if and only if $0 \in \partial F(\hat{u})$, and due to convexity this first-order optimality condition is not only necessary but also sufficient. Another important note on subdifferentials is that for Fréchet-differentiable functionals, the subdifferential coincides with the classical derivative $F'(u)$.

4.4 Algorithms

Up to now, we have formulated the inverse problem for parallel MPI and all its components. Also, we showed that it can be rewritten as the minimization of a convex objective function, to which we can add convex regularizers to improve the condition of the problem, especially when the data is noisy. We selected two algorithms that can solve these minimization problems: the Conjugate Gradient (CG) algorithm, and a primal dual hybrid gradient (PDHG) method. The CG algorithm is very efficient, but problems arise when we add the TV regularizer to the objective function. The PDHG algorithm is perfectly suited for the latter, and very flexible.

4.4.1 Conjugate gradients

This efficient algorithm solves any problem

$$Ku = d$$

as long as K is a self-adjoint and positive definite operator, and it is so efficient because it exploits both of these properties. As discussed in section 4.3, we can write the objective functions in (63) such that a specific operator K is formed that is positive definite and self adjoint. We use the operator K and the data d in the following iteration scheme:

Algorithm 1 Conjugate gradient algorithm

Choose $u^0 \in X$, let $r^0 = d - Ku^0$ and $p^0 = r^0$.

```

for n = 0, 1, 2, ... do
   $\alpha^n = \frac{(r^n)^T r^n}{(p^n)^T K p^n}$ 
   $u^{n+1} = u^n + \alpha^n p^n$ 
   $r^{n+1} = r^n - \alpha^n K p^n$ 
   $\beta^n = \frac{(r^{n+1})^T r^{n+1}}{(r^n)^T K r^n}$ 
   $p^{n+1} = r^{n+1} + \beta^n p^n$ 
end for

```

4.4.2 Primal-dual hybrid gradients

To solve the saddle point problem (69), we use a so called first-order primal-dual hybrid gradient method, introduced by Chambolle and Pock [27]. This algorithm is great in its simplicity, and still has a good convergence rate.

Assume that there is a solution (\hat{u}, \hat{p}) to (69), then we know that we must have

$$0 \in \partial S(\hat{u}) \quad (71)$$

$$0 \in \partial S(\hat{p}). \quad (72)$$

Now, as the inner product $\langle Au, p \rangle$ is Fréchet differentiable, we can use the classical derivative for this term in S , and see that (71) and (72) are equivalent with

$$A\hat{u} \in \partial J^*(\hat{p}), \quad (73)$$

$$-A^*p \in \partial H(\hat{u}) \quad (74)$$

The algorithm in [27] iteratively makes sure that these conditions are fulfilled. In these iterations, the algorithm makes use of proximal operators:

$$(I + \sigma \partial J^*)^{-1} \quad \text{and} \quad (I + \tau \partial H)^{-1}$$

For a convex function F , the proximal operator

$(I + \tau \partial F)^{-1}(y)$ is

$$(I + \tau \partial F)^{-1}(y) := \arg \min_x \left\{ \frac{\|x - y\|^2}{2\tau} + F(x) \right\} \quad (75)$$

Now let's have a look at the algorithm analyzed in [27].

Algorithm 2 General Primal-Dual Hybrid Gradient (PDHG) algorithm

Let $A : X \rightarrow Y$. Choose $(u^0, p^0) \in X \times Y^*$, $\tau, \sigma > 0$ s.t. $\sigma\tau L^2 < 1$, with $L = \|A\|$ and set $\bar{u}^0 = u^0$

for $n=0,1,2,\dots$ **do**

$$\begin{cases} p^{n+1} = (I + \sigma \partial J^*)^{-1}(p^n + \sigma A \bar{u}^n) \\ u^{n+1} = (I + \tau \partial H)^{-1}(u^n - \tau A^* p^{n+1}) \\ \bar{u}^{n+1} = u^{n+1} + \xi(u^{n+1} - u^n) \end{cases}$$

end for

The parameters ξ, σ and τ influence the convergence. Usually, the value of these parameters are $\sigma = \tau = L^{-1}$ and $\xi = 1$. In [27], Chambolle and Pock prove that the primal and dual variables u^n and p^n converge as $n \rightarrow \infty$. With this given, we can see that the algorithm does what we want: for the dual variable, the update in the algorithm is

$$\begin{aligned} p^{n+1} &= (I + \sigma \partial J^*)^{-1}(p^n + \sigma A \bar{u}^n) \\ \implies p^{n+1} &= p^n + \sigma A \bar{u}^n - \sigma \partial J^* p^{n+1} \end{aligned}$$

Now, assume $p^n \rightarrow \hat{p}$ and $u^n \rightarrow \hat{u}$. Then, as $n \rightarrow \infty$,

$$\begin{aligned} \hat{p} &= \hat{p} + \sigma A \hat{u} - \sigma \partial J^* \hat{p} \\ \implies A \hat{u} &\in \partial J^* \hat{p} \end{aligned}$$

which is the optimality condition given in (73). In the same way, one can derive that in the case of convergence, the algorithm makes sure that

$$-A^* \hat{p} \in \partial H \hat{u}.$$

PDHG algorithm for TV minimization

We regard 2-D images, so we consider Ω to be a regular Cartesian grid of size $M \times N$:

$$\Omega := \{(ih, jh) : 1 \leq i \leq M, 1 \leq j \leq N\},$$

where h denotes the size of the spacing and (i, j) denote the indices of the discrete locations (ih, jh) in the image domain. We search for $u \in X = \mathbb{R}^{M \times N}$. X is equipped with the standard scalar product

$$\langle u, v \rangle_X = \sum_{i,j} u_{i,j} v_{i,j}, \quad u, v \in X$$

The last step in our transition from the continuous to the discrete case, is to find a discretization for the objective functional $H(u) + J(Au)$. For $H(u)$ we can just use the l^2 -norm. To discretize $J(Au) = TV(u)$, define

$$(\nabla u)_{i,j} = \begin{pmatrix} (\nabla u)_{i,j}^1 \\ (\nabla u)_{i,j}^2 \end{pmatrix}, \quad (76)$$

where

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N, \\ 0 & \text{if } i = N, \end{cases} \quad (77)$$

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < M, \\ 0 & \text{if } j = M. \end{cases} \quad (78)$$

Then, the total variation of $u \in \mathbb{R}^{M \times N}$ is defined by

$$TV(u) = \|\nabla u\|_1 = \sum_{i,j} |(\nabla u)_{i,j}|, \quad (79)$$

with $|x| := \sqrt{x_1^2 + x_2^2}$ for all $x = (x_1, x_2) \in \mathbb{R}^2$. So we see that in $J(Au)$, we take $J(\cdot) = \|\cdot\|_1$ and $A = \nabla$ defined as (76). Hence, since $A : X \rightarrow Y$, we identify Y with $\mathbb{R}^{M \times N \times 2}$, equipped with the inner product

$$\langle p, q \rangle_Y := \sum_{i,j} (p_{i,j}^1 q_{i,j}^1 + p_{i,j}^2 q_{i,j}^2). \quad (80)$$

In [28], the connection between this discrete version of TV and the continuous, formal definition given in (60) is shown. One can prove that for $M = N$ on $\Omega = (0, 1) \times (0, 1)$ with $h = 1/N$,

$$TV_h := h \sum_{i,j} |(\nabla u)_{i,j}|$$

“ Γ -converges” to (60) as $N \rightarrow \infty$. To see the analogy, we can dig a little deeper into convex analysis. For one-homogeneous convex functions like $TV(u)$, it is a well-known result in convex analysis [29] that the convex conjugate

$$TV^*(v) = \sup_u \langle u, v \rangle - TV(u)$$

is given by

$$TV^*(v) = \chi_K(v) = \begin{cases} 0 & \text{if } v \in K \\ \infty & \text{if } v \notin K, \end{cases} \quad (81)$$

for some closed convex set K . Now remember from section 4.3 that $TV^{**} = TV$, so we recover

$$TV(u) = \sup_{v \in K} \langle u, v \rangle. \quad (82)$$

We can identify K as follows. Recall the innerproduct for Y given in (80), and the discretization of $TV(u)$ in (79). Then we see that we can write

$$TV(u) = \sup_{p \in P} \langle p, \nabla u \rangle_Y, \quad (83)$$

where P is defined as

$$P = \{p \in Y : \|p\|_\infty \leq 1\}. \quad (84)$$

where $\|p\|_\infty = \max_{i,j} |p_{i,j}|$ with $|p_{i,j}| = \sqrt{(p_{i,j}^1)^2 + (p_{i,j}^2)^2}$. Now K has to make sure (82) is equivalent to (83), and $K \subset X$ because the inner product has to be defined. Therefore, we introduce the adjoint operator of the gradient: the divergence operator $\text{div} : Y \rightarrow X$. By definition,

$$\langle \nabla u, p \rangle_Y = \langle u, -\text{div } p \rangle_X$$

for every $p \in Y$ and $u \in X$ which is exactly what we need. Specifically, the X - and Y -inner products that we defined imply that

$$\begin{aligned} (\text{div } p)_{i,j} &= \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{if } 1 < i < M \\ p_{i,j}^1 & \text{if } i = 1, \\ -p_{i-1,j}^1 & \text{if } i = M \end{cases} \\ &+ \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{if } 1 < j < N \\ p_{i,j}^2 & \text{if } j = 1, \\ -p_{i,j-1}^2 & \text{if } j = N \end{cases} \end{aligned}$$

is the adjoint operator to (76). Hence, for discrete images $u \in \mathbb{R}^{M \times N}$ we can write the total variation as (82) with

$$K = \{\text{div } p : p \in P\}$$

where the analogy with the continuous setting is obvious; in (60) we see that

$$K = \{\text{div } g : g \in C_0^\infty(\Omega; \mathbb{R}^2), \|g\|_\infty \leq 1\}.$$

Now that we have discretized the problem, we can form the specific algorithm to solve it.

4.4.3 Simple case

We start off with showing how the PDHG algorithm can solve a simple minimization problem, i.e. the L_2 data fidelity problem

$$\min_{u \in L^2(\Omega)} \frac{1}{2} \|Eu - d\|_2^2, \quad (85)$$

which can be written as

$$\begin{aligned} \min_{u,y} \quad & H(u) + J_1(y), \text{ with} \\ H(u) = \quad & 0 \\ J_1(y) = \quad & \frac{1}{2} \|y - d\|_2^2 \\ y = \quad & Au \end{aligned}$$

Note that we have treated the data fidelity term as if it were a regularization term here; this is merely a choice, the outcome will be the same.

The proximal operator

Now, as we saw in the beginning of this section, we need to find the proximal operators $(I + \tau\partial H)^{-1}$ and $(I + \sigma\partial J_1^*)^{-1}$. So first, we need to find J_1^* :

$$J_1^*(p) = \max_y \langle p, y \rangle_X - J_1(y) := \max_y F(y, p). \quad (86)$$

Substituting the solution of $\frac{\partial F}{\partial y}$ into (86) gives

$$J_1^*(p) = \frac{1}{2} \|p\|_2^2 + \langle p, d \rangle \quad (87)$$

Now, using definition (75), we find

$$(I + \tau\partial H)^{-1}(\xi) = \arg \min_u \left\{ \frac{\|u - \xi\|^2}{2\tau} \right\} \quad (88)$$

$$(I + \sigma\partial J_1^*)^{-1}(\xi) = \arg \min_p \left\{ \frac{\|p - \xi\|^2}{2\tau} + \frac{1}{2} \|p\|_2^2 + \langle p, d \rangle \right\}, \quad (89)$$

Setting the first derivative to zero yields

$$(I + \tau\partial H)^{-1}(\xi) = \xi \quad (90)$$

$$(I + \sigma\partial J_1^*)^{-1}(\xi) = \frac{\xi - \sigma d}{1 + \sigma}. \quad (91)$$

4.4.4 Adding the total variation regularizer

In the objective function, we now add the total variation term:

$$\min_{u \in BV(\Omega)} \frac{\lambda}{2} \|Eu - d\|_2^2 + \|\nabla u\|_1 \quad (92)$$

and we will see how this affects the algorithm. (92) can be written as

$$\begin{aligned} \min_{u,y,z} \quad & H(u) + J_1(y) + J_2(z), \text{ with} \\ H(u) = \quad & 0 \\ J_1(y) = \quad & \frac{1}{2} \|y - d\|_2^2 \\ J_2(z) = \quad & \|z\|_1 \\ y = \quad & Eu \\ z = \quad & \nabla u. \end{aligned}$$

Proximal operators

We already computed $(I + \tau\partial H)^{-1}$ and $(I + \sigma\partial J_1^*)^{-1}$ in equation (91), so we're left to find $(I + \sigma\partial J_2^*)^{-1}$. Note that the proximal operator for J_1^* is slightly different because of the parameter λ in (92):

$$(I + \sigma\partial J_1^*)^{-1}(\xi) = \frac{\xi - \sigma f}{1 + \frac{\sigma}{\lambda}}.$$

To find J_2^* , we compute

$$J_2^*(p) = \max_z \langle p, z \rangle_Y - \|z\|_1 := \max_z G(z), \quad z = \nabla u \in Y$$

There are two cases to be considered: $p \in P$ or $p \notin P$, with P as in (84). It can be proven, and intuition tells us as well, that for $p \in P$, the maximum of $\langle p, z \rangle_Y - \|z\|_1$ is 0. On the other hand, if p has one pixel $p_{i,j}$ with $|p_{i,j}| > 1$, we see that $\langle p, z \rangle_Y$ can get arbitrarily large. Therefore,

$$J^*(p) = \chi_P(p) = \begin{cases} 0 & \text{if } p \in P \\ \infty & \text{if } p \notin P \end{cases} \quad (93)$$

Hence,

$$(I + \sigma \partial J_2^*)^{-1}(\xi) = \arg \min_p \left\{ \frac{\|p - \xi\|^2}{2\tau} + \chi_P(p) \right\}. \quad (94)$$

The quadratic term is minimized when $p = \xi$, but χ_P excludes this minimizer when $\xi \notin P$. To solve this problem, we write the quadratic term as a sum over pixels:

$$\frac{\|p - \xi\|^2}{2\tau} = \frac{\sum_{i,j} |p_{i,j} - \xi_{i,j}|^2}{2\tau}.$$

χ_P places an upper bound on the magnitude of each pixel, which is why we will consider the proximal operator of J_2^* pixel-wise:

$$(I + \sigma \partial J_2^*)^{-1}(\xi_{i,j}) = \begin{cases} \xi_{i,j} & \text{if } |\xi_{i,j}| \leq 1 \\ \frac{\xi_{i,j}}{|\xi_{i,j}|} & \text{if } |\xi_{i,j}| > 1 \end{cases} \quad (95)$$

We write this concisely as

$$(I + \tau \partial J_2^*)^{-1}(\xi) = \frac{\xi}{\max(1, |\xi|)}.$$

Algorithm

With all the information above, we can formulate the explicit update steps of algorithm 2 for TV minimization. To write it concisely, we use

$$f(\xi) := \frac{\xi - \sigma d}{1 + \frac{\sigma}{\lambda}}$$

$$g(\xi) := \frac{\xi}{\max(1, |\xi|)}.$$

Algorithm 3 PDHG algorithm for TV minimization

Choose $(u^0, p^0, q^0) \in X \times Y^* \times Z^*$, $\tau, \sigma > 0$ s.t. $\sigma\tau L^2 < 1$, with $L = \|[E, \nabla]\|$ and set $\bar{u}^0 = u^0$

for $n=0,1,2,\dots$ **do**

$$\begin{cases} p^{n+1} = f(p^n + \sigma E \bar{u}^n) \\ q^{n+1} = g(q^n + \sigma \nabla \bar{u}^n) \\ u^{n+1} = u^n - \tau E^* p^{n+1} + \operatorname{div} q^{n+1} \\ \bar{u}^{n+1} = 2u^{n+1} - u^n \end{cases}$$

end for

5 Results

The results section will be divided into three parts. The first two serve to show the power of parallel imaging and its realization in MPI, without regarding regularization techniques. In the third part, we will investigate how regularization can help to improve the reconstruction quality.

So, in 5.1, parallel MPI reconstructions will be shown where we did the ‘inverse crime’. That is, the data that we reconstruct the image from, was generated by the same operator E that we use for the reconstruction in the CG algorithm. In 5.2, we will reconstruct parallel MPI images from data that is generated by the simulator from section 2.6. This simulator is very much physics driven, and will accurately resemble data from an actual MPI scan. Finally, in 5.3 we will address reconstruction performance under noisy data, and see if the regularization techniques discussed in section 4.2 can help to improve the performance. Throughout these sections, as a measure of quality we will use the *relative error* to a “ground truth” image. Let $\tilde{\rho}_g$ be the ground truth image, and $\hat{\rho}$ the undersampled reconstruction. Be it via parallel imaging using the CG or PDHG algorithm introduced in the previous section, or via the filtered backprojection method discussed in section 2.4. Then, to quantify quality we will use the relative error:

$$e = \frac{\|\tilde{\rho}_g - \hat{\rho}\|_2}{\|\tilde{\rho}_g\|_2}. \quad (96)$$

To also judge qualitatively, we will show the images on a color scale. After all, the human eye is better at distinguishing colors than gray scales.

5.1 Parallel MPI reconstructions from data generated by the forward operator

We will present our first results here, using the 'Cal' phantom shown in Figure 17. We deleted a part of the letter a, as a detail we can judge in the reconstructions. The pixel values of this cal phantom are stored into a matrix ρ of size 128×128 , and given to the operator E we defined in equation (36) to generate the data, using the sensitivity maps from Figure 14. This data is then multiplied by E^* , and given as input to the conjugate gradient algorithm defined in section 4.4.1, together with the operator (E^*E) .



Figure 17: The 'Cal' phantom. It is the unofficial logo of the University of California in Berkeley. We will use it for the first validation of our model. Also, on the right we show the color map that we will use throughout the results.

In Figure 18, we present the output of our algorithm for different numbers of projections. The Figure is meant to give the reader the first visual comparison of the currently used filtered backprojection reconstruction approach and our parallel imaging method. Despite these low numbers of projection angles, our parallel MPI reconstructions still yield good results. To show how the relative error behaves w.r.t. the number of projection angles, we plotted it in Figure 19.

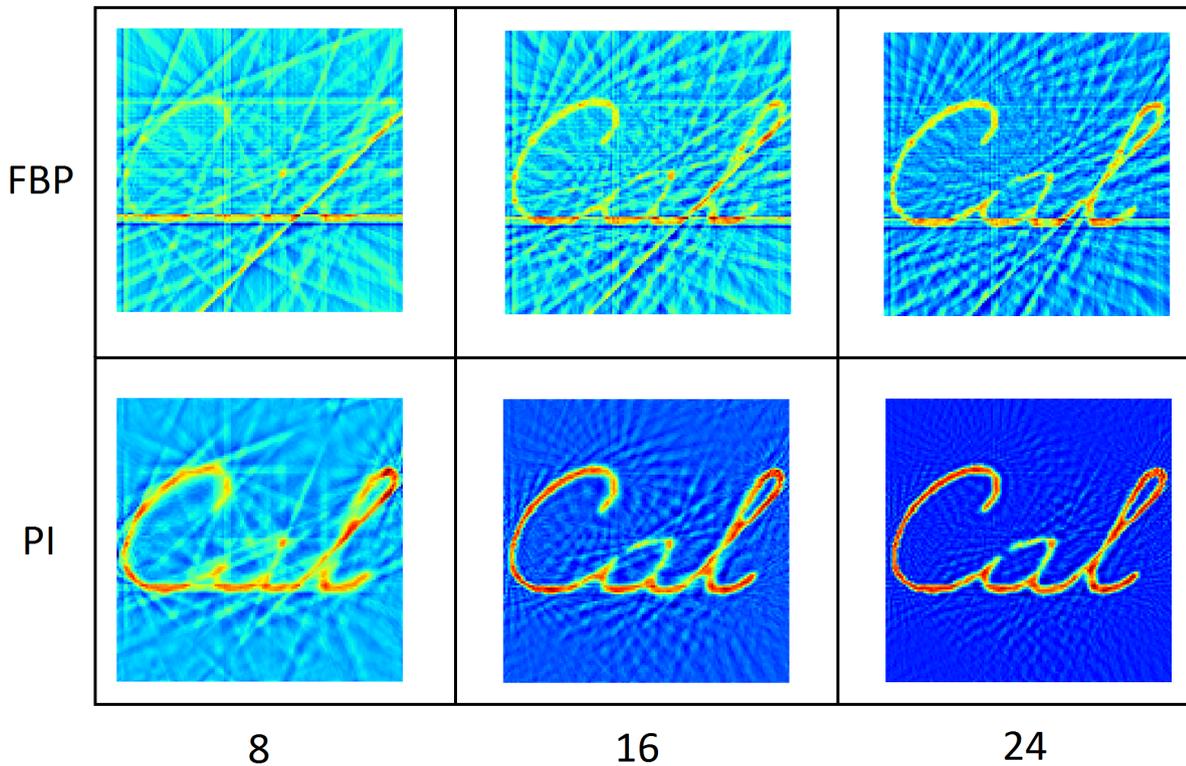


Figure 18: Visual comparison of reconstruction quality for filtered backprojection (FBP) and parallel imaging (PI). We see the reconstructions for 8, 12 and 16 projection angles, which are relatively low amounts.

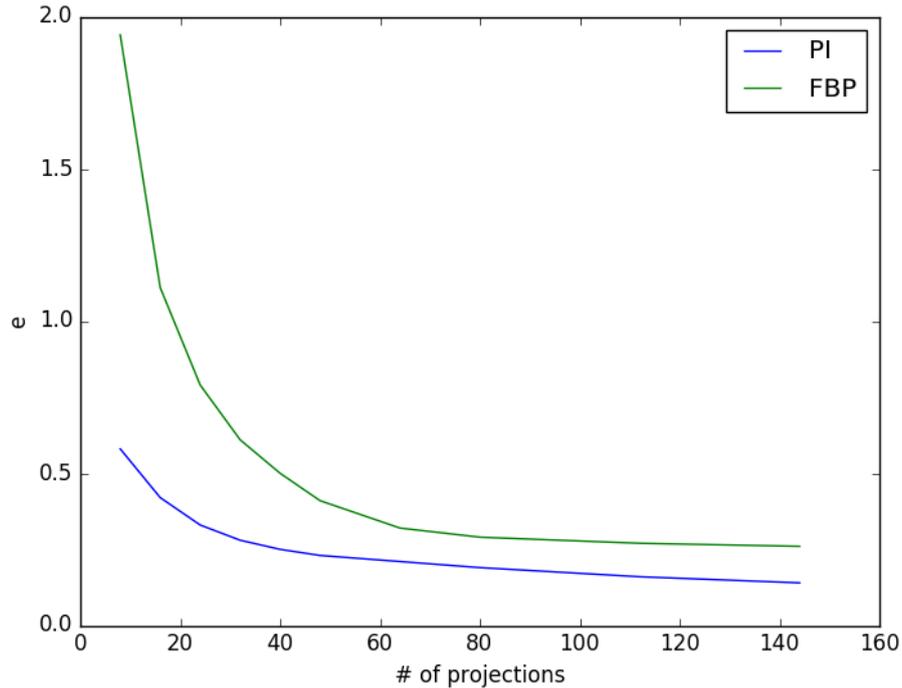


Figure 19: Comparison of parallel imaging (PI) vs. filtered backprojection (FBP) through relative error to the ground truth image. We see that PI performs especially better in the lower number of projection angles regime.

5.2 Parallel MPI reconstructions from simulated data

In the previous section, we saw that our reconstruction algorithms for parallel MPI work, and that parallel MPI has high potential. To show that it will work in practice, we will review the performance of parallel MPI on data that is generated with the physics simulator from section 2.6. As opposed to the previous section, our operator E will now only be an approximation to how the data is generated, making it more difficult for the reconstruction algorithm to find the right $\hat{\rho}$. To measure the performance, we will again need a ‘ground truth’ image, for which we choose the filtered backprojection reconstruction of the simulated data with 180 projection angles. We will use two different phantoms, that we created on the computer. The first one mimics a network of blood vessels, which is an important structure for MPI applications, and the second will show us the performance under several signal strengths.

Blood vessels phantom

In Figure 20 we see the blood vessels phantom and the ground truth reconstruction. As a ground truth we choose the FBP reconstruction with 180 projections. Note that it is blurred, because of the Langevin physics described in section 2.3.2.

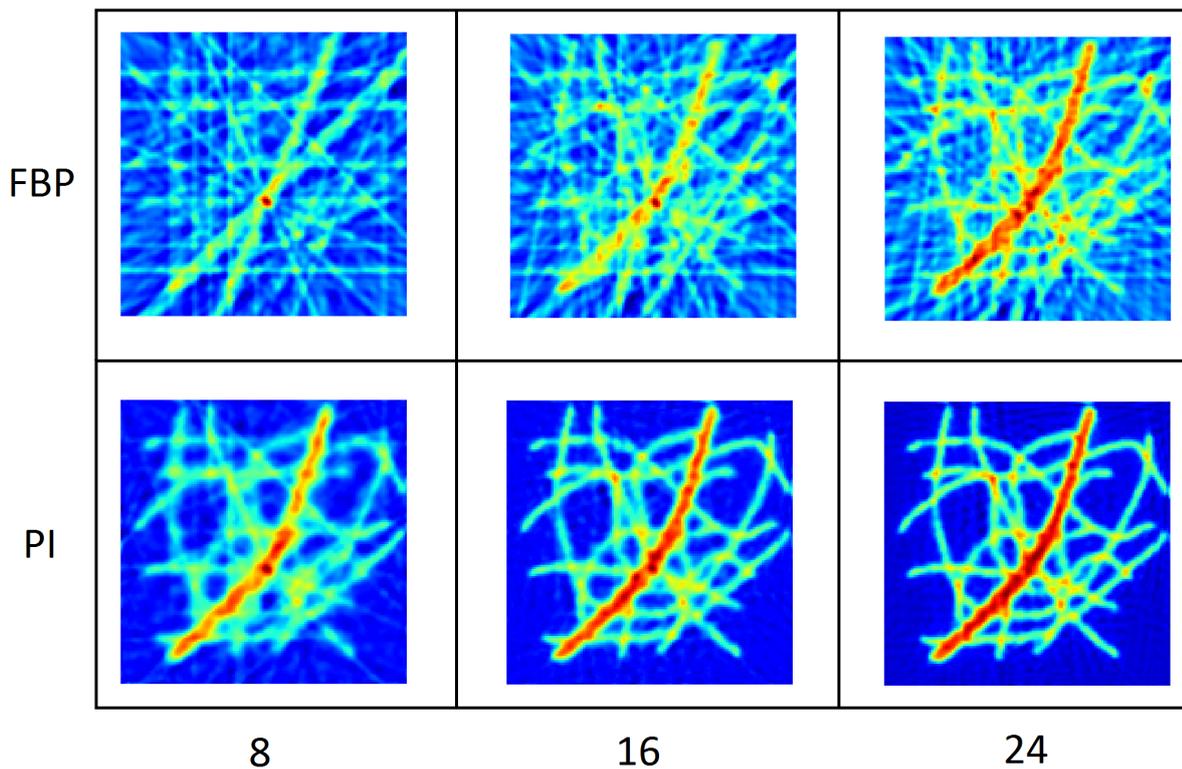


Figure 21: Visual comparison of reconstruction quality for filtered backprojection (FBP) and parallel imaging (PI). We see the reconstructions for 8, 16 and 24 projection angles, which are relatively low amounts.

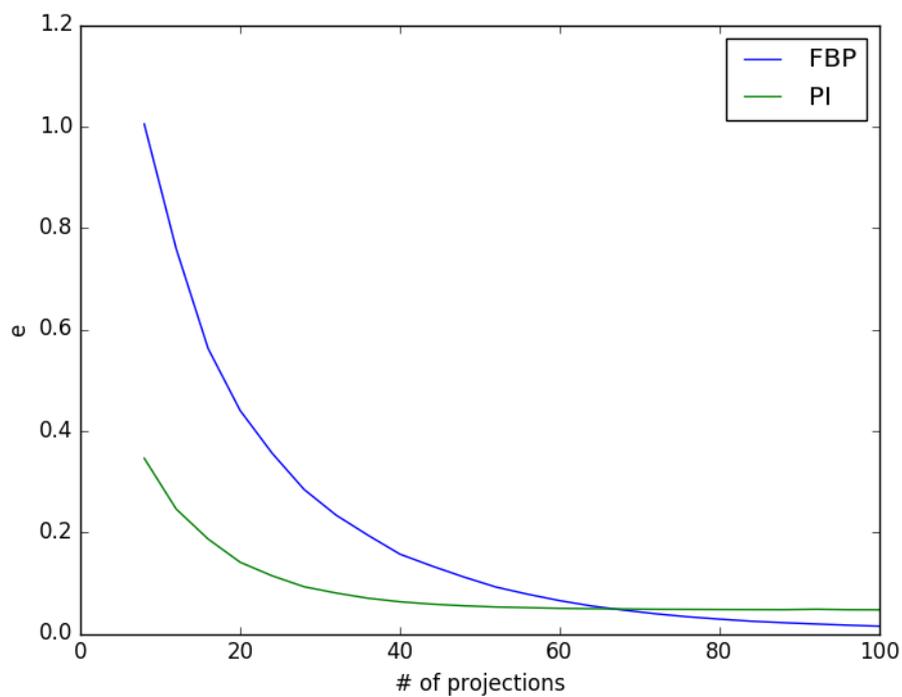


Figure 22: Comparison of parallel imaging (PI) vs. filtered backprojection (FBP) through the relative error to the ground truth image.

Balls phantom

The next phantom we will use for a performance review is the balls phantom shown in Figure 23. Again, in this Figure we also show the filtered backprojection reconstruction from simulated data using 180 projection angles.

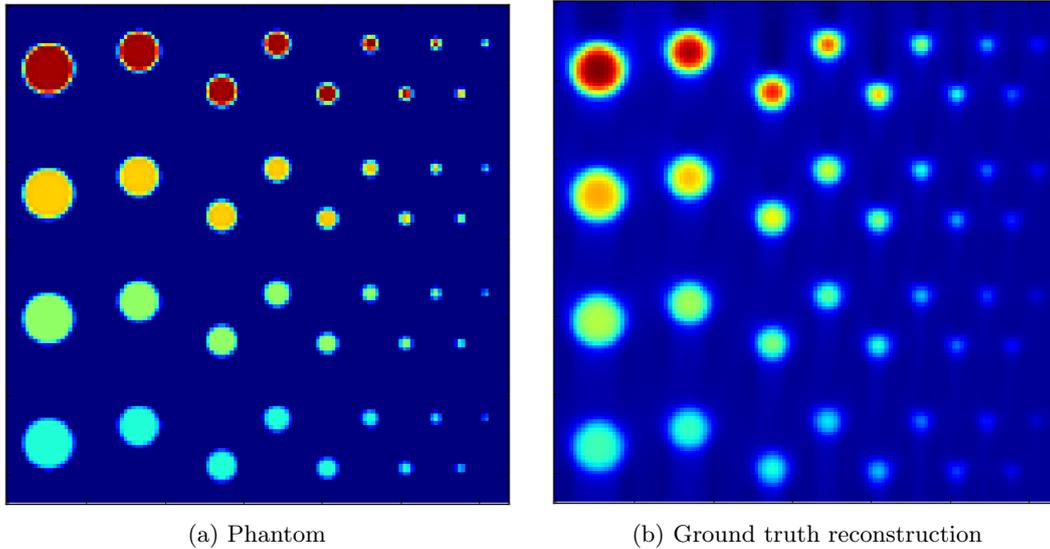


Figure 23: **(a)** A phantom that is sparse, with degrading signal strength from top to bottom. **(b)**: The filtered backprojection reconstruction of the data set generated by our Field Free Line MPI simulator. The data set was formed using 180 projection angles, hence the good quality of the reconstruction.

Next, we are going to undersample Fourier space by using less projection angles, and show the performance of our parallel imaging reconstruction algorithm versus the standard filtered backprojection method. For this phantom, we would particularly like to see how the algorithm is able to reconstruct the lowest signals in the bottom right. We see that already for 12 or more projection angles, the small, low signal balls on the right are reconstructed and can be distinguished from the aliasing. For the FBP reconstructions at these low projection numbers, we see that the balls on the right are completely lost. Also, the shape of the bigger balls are deformed by the aliasing.

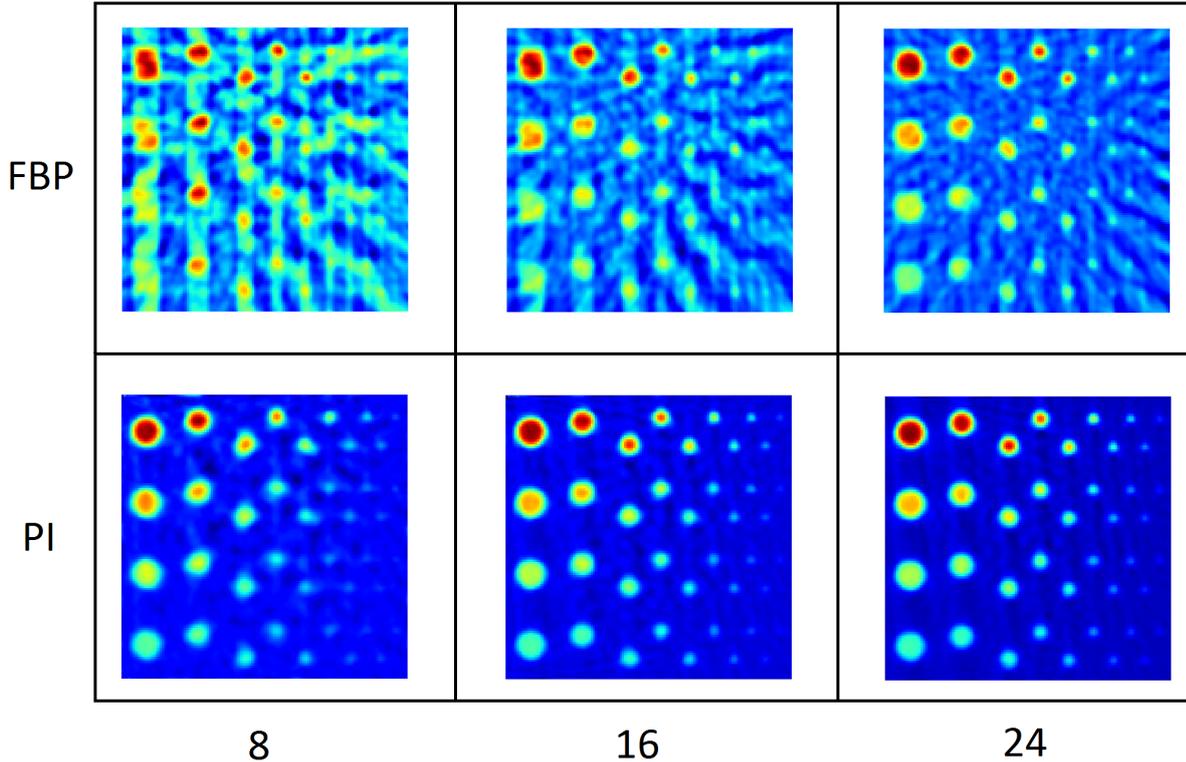


Figure 24: Visual comparison of reconstruction quality for filtered backprojection (FBP) and parallel imaging (PI). We see the reconstructions of the balls phantom from Figure 23, for 8, 16 and 24 projection angles. Note how nicely we are able to reconstruct the balls on the far most right, that are completely lost in the FBP reconstruction.

5.3 Noise and regularization performance

In the previous section, we made the first step towards real data, as we used a simulator that closely resembles the physics of FFL MPI to generate the data. To make the final step, we have to add noise to the data, as this is always present in real life data acquisitions. Like we discussed in sections 3.1 and 4.2, this noise can cause big problems in the reconstruction, especially when a problem is tackled as an inverse problem. We also saw that regularization terms in the objective function can help to improve the condition of the problem, and prevent a ‘blow-up’ of the final solution. In this section, we will first investigate the performance of our parallel MPI reconstructions with noisy data, using only the data fidelity term in the objective function. Then, we add regularizers and analyze if we see better behavior. For the analysis, we will use the balls phantom. This choice is natural, as it has different signal strengths so that we can judge whether the algorithm is able to reconstruct the lower signals despite the noise. Throughout the results, we will again use e as in equation (96), with the FBP reconstruction of 180 projections of the balls phantom as the ground truth.

5.3.1 Added noise, no regularization

To simulate noisy data, we first generate the data d in the parallel FFL MPI simulator, and add a zero mean normally distributed signal with standard deviation σ to d . In Figure 25 we show the results for different noise levels. We need a measure for noise level, and choose it to be σ , where we add

$$d_n \sim N(0, \max(d) \cdot \sigma) \quad (97)$$

to the data d . In Figure 25 we show the unregularized reconstructions for $\sigma = 0.005$, $\sigma = 0.01$ and $\sigma = 0.02$, using 16 projection angles. We see that for $\sigma \geq 0.01$, things get problematic and we get worse reconstructions than with filtered backprojection.

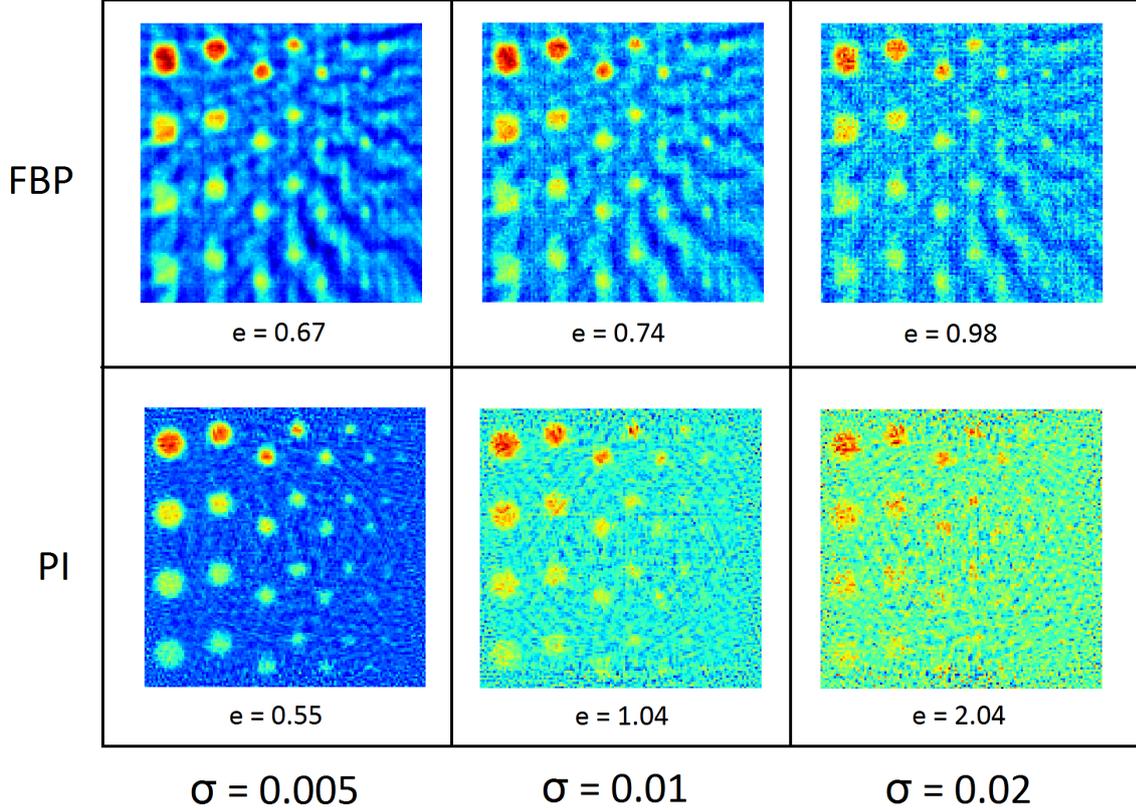


Figure 25: Comparison of FBP and PI reconstructions for different noise levels, using 16 projection angles. To quantify the comparison, we have written the relative error e to the ground truth under each reconstruction. We clearly see that for $\sigma \geq 0.01$, we will have to add regularizers to the objective function of the PI reconstruction model; without it PI performs worse than FBP.

5.3.2 Regularization

To avoid the noise blow-up we saw in section 5.3.1, we try three different regularization terms in the objective function, and compare their performance. One is L^2 regularization, also known as Tikhonov regularization. Like we addressed in section 4.2, this regularizer does avoid the blow-up, but does not put a penalty on the noise. Hence, we expect improved performance, but no denoising. Then, to penalize noise, we moved to H^1 regularization, where we expect to see an SNR improvement, at the cost of resolution. Also, it will only smooth out the noise and not get rid of it. Finally, we applied the theory on Total Variation minimization and used the PDHG algorithm to find the solution. The ballsphantom is sparse, hence we expect this regularization to be able to totally get rid of the noise in the regions where the signal is zero. A known disadvantage of the Total Variation regularizer is contrast loss; it will be interesting to see which of the small balls on the right will still be visible using this technique. For the tests we will use 16 projection angles, as we saw in section 5.2 that without noise, this was sufficient for good parallel MPI reconstructions. The images are shown in Figure 26. Under each image, we give the relative error to the ground truth, and our choice for the regularization parameter λ . Note that in our objective functions in (63) and (64), the data fidelity term is multiplied by λ . Hence, the lower the value for λ , the more influence the regularization penalty has on the final solution. Based on the noise level, we ran some tests for λ in a certain range, and chose the value for which the relative error approached a minimum.

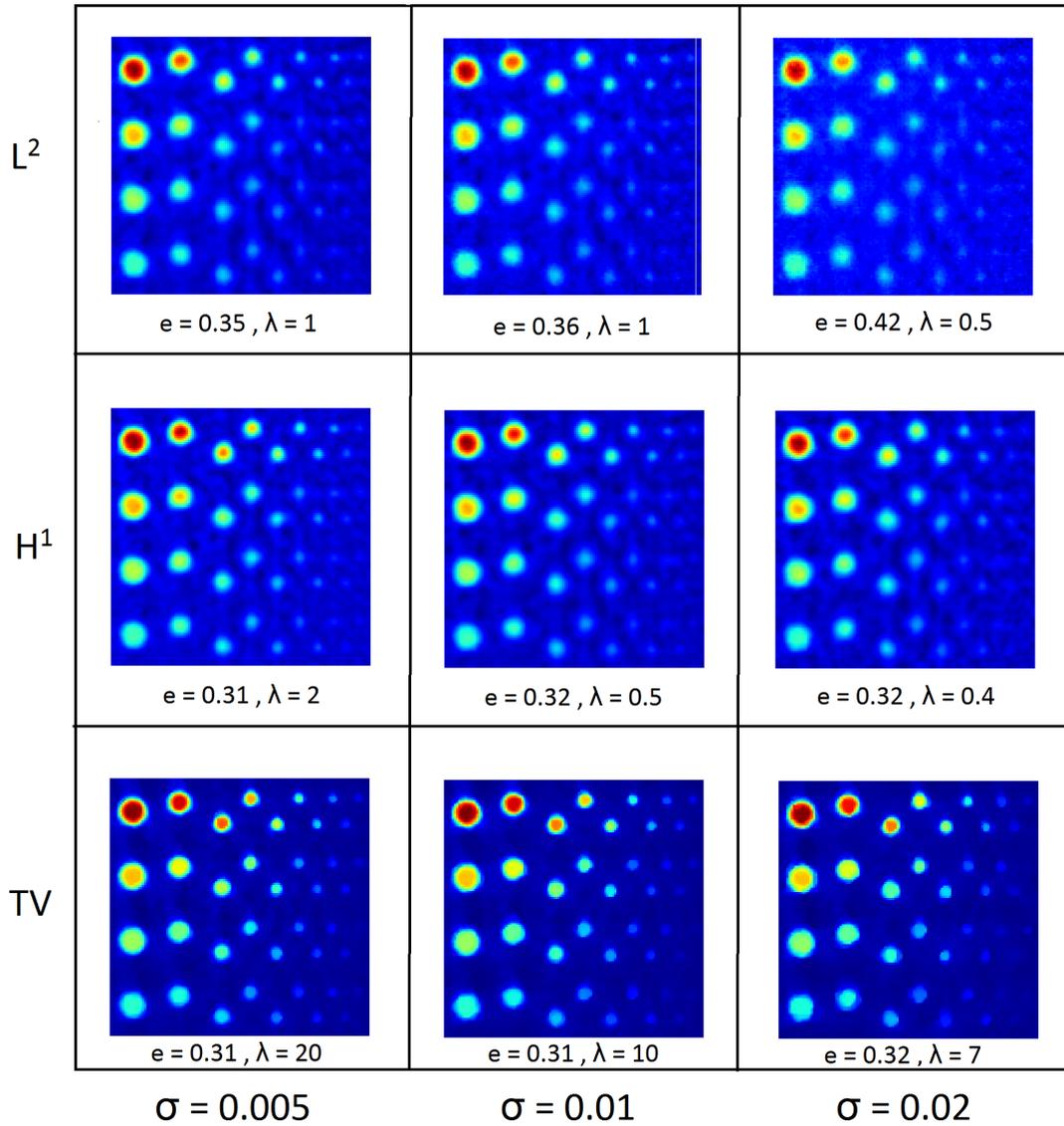


Figure 26: Comparison of regularization techniques at different noise levels. For each reconstruction, the data was generated by the simulator using 16 projection angles. The noise was added as in equation (97).

First of all, we observe that each of our chosen regularization techniques does a really good job at preventing the noise blow-up we saw in Figure 25 for $\sigma \geq 0.01$. Also, the bigger balls are being reconstructed pretty well in each case. However, when the balls tend to get smaller, especially in the bottom right where the signal strength is lower, problems arise. For L^2 and H^1 regularization, we see spots in the image that could be a small ball, but also some high-amplitude, smoothed noise. This can be very problematic in clinical practice, as doctors won't be able to decide if the signal is diagnostic or just noise. As expected, for the TV regularization we see that we can get rid of all the noise in the background, making it way more easy to differentiate between real signal and noise. Although the contrast loss is also visible, especially for $\sigma = 0.02$ where we had to use $\lambda = 7$, the phantom is reconstructed really well. Also, bear in mind that we only used 16 projection angles.

6 Conclusion and future scope

We presented the first parallel Magnetic Particle Imaging method, and showed that it is both theoretically and practically feasible. Although there was no time or money to build an actual parallel MPI scanner, we were able to approximate a real setup by building a simulator to generate the data. This simulator is based on knowledge of the behavior of the nanoparticles, and the scanning strategy we chose: field free line excitations and six parallel receive coils to pick up the resulting signal. To find the parallel MPI image we built a forward model that relates the data d to the desired nanoparticle density distribution ρ . This inverse problem is then rewritten into a minimization problem and solved algorithmically, using the efficient conjugate gradients algorithm. Dependent on the structure of the phantom, we can push the number of projections needed as low as 8 and still obtain good quality. Currently, in field free line MPI at least 60 projections are made to obtain diagnostically relevant images. We do see that problems arise when noise is added to our data, which is why we added regularization terms to the objective function to improve the condition of the problem. All of these had success, but the total variation regularizer performs best in both the noise reduction and reconstruction quality.

For future work, a logical next step is to write a proposal to get funding for the construction of an actual parallel MPI scanner, and obtain real images. However, we would like to recommend a very important step that has to be made first before such a proposal is written. The quality of the parallel MPI results can have a very close relationship to the choice of the hardware, specifically w.r.t. the number of receive coils and their geometrical orientation. To optimize for this was beyond the scope of this thesis, but can be very useful. The construction of a scanner will be quite costly, so it has to be done right, i.e. the best number of receive coils with an optimal geometric orientation. To find this optimum, we recommend to capture all the operations that are performed by the operator E into one matrix. Then, one can analyze the condition number of this matrix for different hardware setups. Also, a signal to noise (SNR) analysis can be done, with Biot-Savart law simulators. Given a certain hardware setup, the sensitivity (governed by the Biot-Savart law and reciprocity) to each spatial location is unique, yielding different SNR properties. Also, one can look at compressed sensing techniques in combination with random angle projections or golden ratio angle projections. L^1 wavelet regularizers can then be used to decrease the number of projections even further. These techniques have been tried for MRI (see [40],[43]), and are suited for our parallel MPI method because of the forward model approach, and the fact that we are (indirectly) sampling radially in k-space. Finally, it would be nice to investigate if the operator E can be extended with the point spread function convolution operation. The resolution of the reconstructed image will improve, but we expect to see a lot more difficulty in handling noise.

A practical issue that will arise once the scanner has been built, is the estimation of the sensitivity map of each receive coil. Each object, animal or person has a different susceptibility distribution across its body, leading to a different magnetic susceptibility distribution inside the entire imaging bore. Hence, for each scan the sensitivity maps of the receive coils will differ. In our opinion, the ESPiRiT algorithm in [38] is the best way to deal with this. It estimates the sensitivity maps directly based on the data, so that no time has to be spent on calibrating scans, etc. Moreover, it is suited for parallel MPI, as the ESPiRiT method relies on an oversampling at the center of k-space. By the Fourier Slice Theorem, FFL MPI samples k-space radially, leading to dense sampling at the center of k-space.

References

- [1] Pankhurst, Q.A. and Connolly, J. Applications of magnetic nanoparticles in biomedicine. *J. Phys. D: Appl. Phys.* 2003; 36 R167.
- [2] Bach-Gransmo, T. Ferrimagnetic susceptibility contrast agents. *Acta Radiology* 1993; 387:1-30 .
- [3] Romanus, E. and Weber, P. Magnetic nanoparticle relaxation measurement as a novel tool for in vivo diagnostics. *Journal of Magnetism and Magnetic Materials* 2002; 252:387-389.
- [4] Kodama, R.H. Magnetic Nanoparticles. *Journal of Magnetism and Magnetic Materials* 1999; 200:359-372.
- [5] Wang, Y.X.J. and Krestin, G.P. Superparamagnetic iron oxide contrast agents: physicochemical characteristics and applications in MR imaging *Eur. Radiol.* 2001; 11:2319-2331.
- [6] Pankhurst, Q.A. Progress in applications of magnetic nanoparticles in biomedicine. *J. Phys. D: Appl. Phys.* 2009; 224001
- [7] Weizenecker, J. and Gleich, B. Tomographic Imaging using the Nonlinear Response of Magnetic Nanoparticles. *Nature.* 2005;435(7046):1214-1217
- [8] Sun, C. and Zhang, M. Magnetic nanoparticles in MR imaging and drug delivery. *Advanced Drug Delivery Reviews* 2008; 60:1252-1265.
- [9] Baumgarten, D. Magnetic nanoparticle imaging by means of minimum norm estimates from remanence measurements. *Med. Biol. Eng. Comput.* 2008; 46:1177-1185
- [10] Visscher, M. and Ten Haken, B. Selective detection of magnetic nanoparticles in biomedical applications using differential magnetometry *Journal of Magnetism and Magnetic Materials* 2014;365:31-39. .
- [11] Gleich, B. Principles and Applications of Magnetic Particle Imaging. *Springer Vieweg* 2014, Lübeck, Germany.
- [12] Center for disease control and prevention. <https://www.cdc.gov/diabetes/pubs/pdf/kidneyfactsheet.pdf>. 2014.
- [13] Nunn, A.D. and Tweedle, M.F. Can receptors be imaged with MRI agents? *Quarterly Journal of Nuclear Medicine* 1997; 41:155-162.
- [14] Heyn, C. and Foster, P.J. Detection threshold of single SPIO-labelled cells with FIESTA. *Magn. Reson. Med.* 2005; 53:312-320.
- [15] Weizenecker, J. and Gleich, B. Three-dimensional real-time in vivo magnetic particle imaging. *Phys. Med. Biol.* 2009;54(5):L1-L10. .
- [16] Weizenecker, J. and Gleich, B. A simulation study on the resolution and sensitivity of magnetic particle imaging. *Phys. Med. Biol.* 2007; 52:6363-6374.
- [17] Rahmer, J., Weizenecker, J., Gleich, B. and Borgert, J. Signal encoding in magnetic particle imaging properties of the system function *BMC medical imaging* 2009; 9:4.
- [18] Knopp, T. and Buzug, T.M. 2-D Model based reconstruction for magnetic particle imaging. *Medical Physics* 37, 485 (2010) 2010; 37 , 485.
- [19] Rahmer, J., Weizenecker, J., Gleich, B. and Borgert, J. Analysis of a 3-D system function measured for magnetic particle imaging. *IEEE Transactions on Medical Imaging* 2012; vol.31 no.6.
- [20] B. Gleich, J. Weizenecker, and J. Borgert. Experimental results on fast 2D-encoded magnetic particle imaging. 536, N81 *Phys. Med. Biol.* 2008; 53(6):N18-N84.
- [21] Goodwill, W. and Conolly, S. The X-Space formulation of the magnetic particle imaging process: 1-D signal, resolution, bandwidth, SNR, SAR, and magnetostimulation. *IEEE Transactions on Medical Imaging* 2010; vol.29, no.11 1851-1859.
- [22] Conolly, S. and Goodwill, W. Multidimensional x-space magnetic particle imaging *IEEE Transactions on Medical Imaging* 2011; vol.30, no.9 1581-1589.
- [23] Chikazumi, S. Physics of magnetism. *Wiley press* 1964. .

- [24] Den Dekker, A.J. and Van den Bos, A. Resolution: a survey. *J. Opt. Soc. Am.* 1997; vol. 15 no.3
- [25] Rosensweig, R. Ferrohydrodynamics. *Dover Publications, New York* 1985.
- [26] Evans, L.C. Partial Differential Equations. *Graduate Studies in Mathematics* Second Edition, 2010.
- [27] Pock, T. and Chambolle, A. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision* 2011; vol. 40 no. 1 120-145
- [28] Braides, A. Gamma-convergence for beginners. *Oxford lecture series in Mathematics and its applications No. 22.* Oxford University Press, 2002.
- [29] Ekeland, I. and Teman, R. *Convex Analysis and Variational Problems.* Amsterdam, 1976.
- [30] J. Weizenecker, B. Gleich, and J. Borgert. Magnetic particle imaging using a field free line. *J. Phys. D: Appl. Phys.* 2008; vol. 41, no. 10.
- [31] J. Kunckle and P. Goodwill. Projection reconstruction magnetic particle imaging. *IEEE transactions on med. imaging* 2013; vol.32, no.2.
- [32] Larkman, D. and Nunes, R. Parallel magnetic resonance imaging. *Physics in Medicine and Biology.* 2007; 52 R15-R55.
- [33] Beatty, P.J. , Nishimura, D. and Pauly, J.M. . Rapid gridding reconstruction with a minimal oversampling ratio. *IEEE Transactions on Medical Imaging.* 2005; vol. 24, No.6 .
- [34] Sodickson, D.K. and Manning, W.J. Simultaneous acquisition of spatial harmonics (SMASH): fast imaging with radiofrequency coil arrays *Magn. Reson. Med.*. 1997; Vol. 38 ,591603
- [35] Pruessmann, K.P. . SENSE: Sensitivity Encoding for fast MRI *Magn. Reson. Med.*. 1999; Vol. 42, 952-962.
- [36] Griswold, M.A. Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA) *Magn. Reson. Med.*. 2002, Vol. 47, 1202-1210.
- [37] Lustig, M. and Pauly, J.M. . SPIRiT: Iterative Self-consistent Parallel Imaging Reconstruction from Arbitrary k-Space. *Magn. Reson. Med.*. 2010; Vol. 64, 457-471.
- [38] Uecker, M. and Lustig, M. ESPIRiT - An Eigenvalue Approach to Autocalibrating Parallel MRI: Where SENSE meets GRAPPA. *Magn. Reson. Med.*. 2014; Vol. 71, 990-1001.
- [39] Griffiths, D.J. Introduction to Electrodynamics, third edition.
- [40] Lustig, M. , Donoho, D. and Pauly, J.M. Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging. *Magn. Reson. Med.*. 2007; Vol. 58, 1182-1195.
- [41] Esser, E. , Zhang, X. and Chan, T.F. A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science. *SIAM J. Imaging Sciences.* 2010; Vol. 3, No. 4, pp. 1015-1046.
- [42] Acar, R. and Vogel, C.R. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse problems.* 1994; Vol. 10 , pp 1217-1229.
- [43] Winkelmann, S. An optimal radial profile order based on the golden ratio for time-resolved MRI. *IEEE Transactions on medical imaging.* 2007; Vol. 26, No. 1 .