

Compressed Sensing accelerated radial acquisitions for dynamic Magnetic Resonance Imaging

Ian Zwaan

Master's Thesis
Department of Mathematics
Utrecht University

December 2012

Supervisors:
Dr. Gerard L.G. Sleijpen
Peter R. Seevinck, PhD

Contents

Introduction	3
1 Magnetic Resonance Imaging	5
1.1 Introduction	5
1.2 Physical principles	5
1.3 Fourier imaging	9
1.4 Basic k -space traversal	11
1.4.1 Slice selection	11
1.4.2 Phase and frequency encoding	11
1.4.3 Sampling density	12
1.5 Radial k -space coverage	13
References	14
2 Numerical methods for Compressed Sensing	15
2.1 Introduction	15
2.2 Mathematical background	15
2.3 Numerical methods	18
2.3.1 Key ingredients	18
2.3.2 Greedy algorithms	19
2.3.2.1 Orthogonal Matching Pursuit	19
2.3.2.2 Stagewise Orthogonal Matching Pursuit	20
2.3.2.3 Compressive Sampling Matching Pursuit	21
2.3.2.4 In-Crowd	22
2.3.3 Convex relaxation algorithms	23
2.3.3.1 Normalised Iterative Hard Thresholding	23
2.3.3.2 Fixed-Point Continuation	25
2.3.3.3 Fast Iterative Shrinkage-Thresholding Algorithm	27
2.3.3.4 Spectral Projected Gradient for ℓ_1 Minimization	29
2.3.3.5 Nonlinear Conjugate Gradients	31
2.4 Numerical experiments	32
2.4.1 Recovery from noise-free and noised measurements	32
2.4.2 A practical example	34
2.4.3 A note about runtime performance	35
2.5 Conclusion	36
References	42

3 Accelerated Radial Compressed Sensing	44
3.1 Introduction	44
3.2 Theory	45
3.2.1 Background	45
3.2.2 Exploiting periodicity	47
3.2.3 Implementation	49
3.3 Motivation	52
3.4 Numerical experiments	54
3.4.1 Test setup	54
3.4.2 Results and discussion	56
3.5 Conclusion	57
References	74

Introduction

High speed image reconstruction from undersampled data is crucial for (near) real-time dynamic *Magnetic Resonance Imaging*. This particular type of imaging could facilitate accurate intraprocedural targeting, monitoring and control for, e.g., minimally invasive image-guided diagnosis and treatment of oncological disease. When MRI can be used for guidance of these minimally invasive interventions, exploiting its unprecedented visualization of human anatomy and physiological processes, a broad range of new treatment options will become possible.

Up to now only limited use of MRI for guidance of minimally invasive interventions has been reported, caused by two tightly coupled limitations of conventional MRI. First, dephasing and low spatial resolution result in inaccurate depiction of small metallic objects typically encountered in interventional devices. And second, slow image acquisition and/or reconstruction with conventional MR imaging methods result in limited temporal resolution (precision in time). Imaging methods generally used to increase the temporal resolution (e.g., echo planar imaging) worsen the spatial accuracy even more, rendering current-day MRI unsuitable for these procedures.

The recently developed imaging method coRASOR¹ has overcome the first of the two limitations just mentioned. CoRASOR is a radial sampling image acquisition strategy which enables depiction of both small metallic objects as well as needles. However, the limited temporal resolution remains a challenge to be addressed.

The goal of this project is to improve image reconstruction speed of radially acquired and undersampled data, by using *Compressed Sensing* to exploit structure in medical images. Development of a Compressed Sensing platform for 2D and 3D radial data ensures future compatibility with coRASOR. In addition, experience shows that radial sampling strategies are particularly suitable for dynamic imaging.

This thesis consists of three parts. **Chapter 1** is a short overview of Magnetic Resonance Imaging intended for mathematicians. Of particular interest is the mathematics behind radial acquisitions in **Section 1.5**.

Chapter 2 introduces Compressed Sensing and studies and compares a number of reconstruction algorithms. Be sure to read about the *Fast Iterative Shrinkage-Thresholding Algorithm* in **Section 2.3.3.3**, since the reconstruction algorithm used in **Chapter 3** derives from it.

Finally, in **Chapter 3** we present a flexible method dubbed *Accelerated Radial Compressed Sensing* (ARCS) which uses Compressed Sensing to reconstruct 2D and 3D radial data. Our tests on 2D radial data show that ARCS is competitive in quality

¹Peter R. Seevinck et al. “Highly localized positive contrast of small paramagnetic objects using 3D center-out radial sampling with off-resonance reception”. In: *Magnetic Resonance in Medicine* 65.1 (Aug. 2011), pp. 146–156. ISSN: 1522-2594. DOI: [10.1002/mrm.22594](https://doi.org/10.1002/mrm.22594). URL: <http://dx.doi.org/10.1002/mrm.22594>.

with traditional CS reconstruction methods (which reconstruct Cartesian data) and is five to twenty times as fast at the same time. Therefore, we believe that ARCS is a novel approach that warrants additional research.

Chapter 1

Magnetic Resonance Imaging

1.1 Introduction

Common medical imaging methods include X-ray techniques and ultrasound techniques. X-ray techniques such as conventional X-ray and computerized tomography visualize differences in tissue density, but require the use of ionizing radiation to do so. Another downside is the fact that differences in tissue densities are often very small. Ultrasound techniques are safe and use the reflection signature of focused sound waves to image soft tissues. However, ultrasound methods require a clear acoustic window and provide poor spatial resolution.

Magnetic resonance imaging (MRI) does not require ionizing radiation, can highlight small differences between tissues, provides high spatial resolution and is noninvasive. MRI can provide highly detailed morphological images as a result. The flexibility of MRI techniques allow different applications as well, typical examples include angiography (visualization of blood vessels) and functional imaging (visualization of brain activity).

The main drawback of MRI is its low imaging speed. Even with (current) fast imaging techniques and undersampling methods, imaging speed severely limits the use of MRI for interventional purposes. Recent research has shown that the upcoming theory of *compressed sensing* can be used to obtain significant improvements in undersampling. Such improvements directly translate to faster imaging.

The following treatment of MRI is a simplified and high-level overview; many details are omitted for the sake of clarity. The first part provides an overview of the physical principles underlying MRI, the second part describes how the measured signal translates to a Fourier transform, the third part introduces basic k -space traversal and the last part describes radial k -space traversal, a fast imaging method with good undersampling properties. Much of the following material can be found in more detail in [1]. Another introduction to MRI can be found in [2].

1.2 Physical principles

The nucleus of a hydrogen atom, a proton, is both NMR active and abundant in the human body. As a result, study of particles in MRI is usually restricted to the proton. The proton is a charged particle that possesses a natural angular momentum, or spin. This spinning charge can be viewed as a current loop which

in its turn induces a magnetic field aligned with the axis of the spin. The direction and magnitude of the magnetic field are represented by the magnetic moment μ . Under normal circumstances the magnetic moment is randomly oriented. When an external magnetic field \mathbf{B}_0 is applied the magnetic moment tends to align with that field. However, the moment does not align exactly with the external field. A complex combination of forces such as gravity, spin and \mathbf{B}_0 cause μ to rotate around \mathbf{B}_0 . This type of motion is called *precession*, see [Figure 1.1](#). The rate of precession is given by the Larmor (angular) frequency

$$\omega_0 = \gamma B_0, \quad (1.1)$$

where γ is a constant, called the gyromagnetic ratio, which depends on the nucleus involved. In case of a proton $\gamma \approx 2.675 \cdot 10^8$.

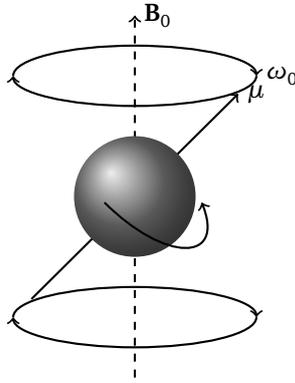


Figure 1.1: Precession of a spinning nucleus about the axis of an applied magnetic field.

A proton can either align parallel (with) or anti-parallel (against) to the static field. The former alignment has lower energy than the latter. As a result of the thermal energy at room temperature, there is only a slight excess of protons favouring parallel (low energy) alignment in a collection of protons. This so called *spin excess* creates a net magnetization, or longitudinal equilibrium magnetization, M_0 parallel to the static field. It can be shown that M_0 is given by

$$M_0 = \frac{B_0 \gamma^2 \hbar^2}{4kT} \rho_0, \quad (1.2)$$

where ρ_0 is the number of protons per unit volume (spin density), k the Boltzmann constant (the Boltzmann constant only appears in formulas (1.1) and (1.16) and should not be confused with the k from k -space), T the absolute temperature and \hbar Planck's constant over 2π . There is no transverse magnetization component since the phase of precession is random with respect to individual protons. As a result, the net magnetization vector is static and does not induce a current. In order to create a measurable signal it is necessary to *excite* the protons.

Consider a frame of reference rotating at the Larmor frequency and with its z -axis aligned with \mathbf{B}_0 . Remark that individual magnetic moments of protons appear stationary in this rotating frame. Suppose it is possible to apply an external magnetic field such that the effective magnetic field experienced in the rotating

frame of reference is aligned with the x -axis. In this case the net magnetization vector is rotated about the x -axis through the angle

$$\Delta\theta = \gamma B_1 \tau, \quad (1.3)$$

where B_1 is the effective field strength and τ the time interval the field is applied. The motion is precession about the \mathbf{B}_1 axis, compare (1.1) and (1.3). The maximum traverse magnetization is achieved when $\Delta\theta = \pi/2$. In this case the net magnetization vector aligns with the rotating y -axis. In a fixed frame of reference, the *laboratory frame*, this means that the net magnetization vector rotates in the traverse plane. This magnetic ‘flux’ can be measured.

It turns out that the magnetic field required above can be created using two¹ perpendicular magnetic fields oscillating in a fixed direction. The frequencies of the magnetic fields need to match the Larmor frequency precisely to achieve the desired effect, this is called the *on-resonance condition*. Virtually no energy is absorbed at off-resonance frequencies. Since the Larmor frequency usually resides in the range of radio waves, the superposition of magnetic fields used to rotate the net magnetization vector is often referred to as a radio frequency (RF) pulse.

After excitation the strength of the signal decays as the spins return to equilibrium. There are two main sources of decay, also called *relaxation*. The first is *spin-lattice* relaxation, the excited spin loses energy and returns to equilibrium. The spontaneous emission of energy is negligible and most energy is exchanged with the environment, the lattice, of the spin. The second is *spin-spin* relaxation, random fluctuations of the local magnetic field affect precession frequencies of different spins. The spins dephase and the traverse net magnetization is lost.

Both spin-lattice and spin-spin relaxation occur at an exponential rate. Spin-lattice relaxation is characterised by the *spin-lattice relaxation time* T_1 , the time scale of the (constant) growth rate of the longitudinal magnetization M_z proportional to $M_0 - M_z$. Spin-spin relaxation is characterised by the *spin-spin relaxation time* T_2 , the time scale of the decay rate of the traverse magnetization \mathbf{M}_\perp proportional to \mathbf{M}_\perp . Spin-lattice and spin-spin relaxation are independent and usually T_1 is (much) larger than T_2 . A graphical interpretation of the flip, precession and decay is shown in [Figure 1.2](#).

Combining precession, spin-lattice relaxation and spin-spin relaxation leads to the following equation of motion

$$\frac{d\mathbf{M}}{dt} = \gamma \mathbf{M} \times \mathbf{B} + \frac{1}{T_1} (M_0 - M_z) \hat{\mathbf{z}} - \frac{1}{T_2} \mathbf{M}_\perp. \quad (1.4)$$

This equation is known as the *Bloch equation* with the solution given by

$$M_x(\mathbf{r}, t) = e^{-t/T_2(\mathbf{r})} (M_x(\mathbf{r}, 0) \cos \omega_0 t + M_y(\mathbf{r}, 0) \sin \omega_0 t) \quad (1.5)$$

$$M_y(\mathbf{r}, t) = e^{-t/T_2(\mathbf{r})} (M_y(\mathbf{r}, 0) \cos \omega_0 t - M_x(\mathbf{r}, 0) \sin \omega_0 t) \quad (1.6)$$

$$M_z(\mathbf{r}, t) = M_z(\mathbf{r}, 0) e^{-t/T_1(\mathbf{r})} + M_0 (1 - e^{-t/T_1(\mathbf{r})}). \quad (1.7)$$

The terms \mathbf{r} above are not required since the Bloch equation is not space dependant, however, they are useful in the discussion below. It is often useful to use the complex

¹Strictly speaking, one oscillating magnetic field would be sufficient. However, there are numerous reasons why using two fields is advantageous, see e.g. [1, Section 3.2.1-3.2.2].

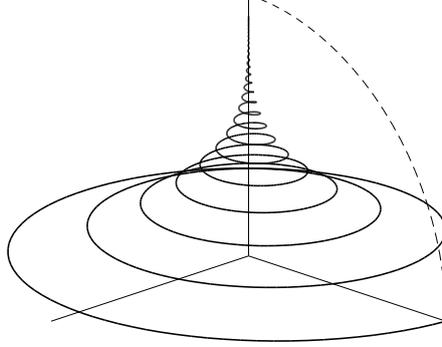


Figure 1.2: Precession of the net magnetization after a $\pi/2$ flip. The flip is depicted by the dashed line and the precession by the solid line. Longitudinal and traverse magnetization are lost due to T_1 and T_2 decay respectively. In normal circumstances the precession frequency would be much higher than shown here.

notation for the traverse magnetization component $\mathbf{M}_\perp = M_x \hat{\mathbf{x}} + M_y \hat{\mathbf{y}}$. With the notation $M_+(\mathbf{r}, t) \equiv M_x(\mathbf{r}, t) + iM_y(\mathbf{r}, t)$ it is easy to see that

$$M_+(\mathbf{r}, t) = e^{-t/T_2(\mathbf{r})} e^{-i\omega_0 t} M_+(\mathbf{r}, 0) = e^{-t/T_2(\mathbf{r})} e^{-i\omega_0 t + i\phi_0(\mathbf{r})} M_\perp(\mathbf{r}, 0). \quad (1.8)$$

Using electromagnetic theory one can show that the magnetic flux through a coil can be written as

$$\Phi_M = \int_{\text{sample}} \mathbf{M}(\mathbf{r}, t) \cdot \mathcal{B}^{\text{receive}}(\mathbf{r}) d\mathbf{r}, \quad (1.9)$$

where $\mathcal{B}^{\text{receive}}(\mathbf{r})$ is the magnetic field per unit current that would be produced by the coil at point \mathbf{r} . The current, or *electromotive force* (emf), induced in a coil is given by the change in the magnetic flux over time: $\text{emf} = d\Phi_M/dt$. The measured ‘signal’ is proportional to the emf. In typical MRI experiments the Larmor frequency ω_0 is at least four orders of magnitude larger than $1/T_1$ and $1/T_2$. Therefore the derivatives of e^{-t/T_1} and e^{-t/T_2} can be neglected compared to the derivative of $e^{-i\omega_0 t}$. As a result the derivative of M_z is neglected entirely. Using the complex representation of M it is easy to show that (1.9) becomes

$$\begin{aligned} \text{signal} \propto \omega_0 \int e^{-t/T_2(\mathbf{r})} M_\perp(\mathbf{r}, 0) [\mathcal{B}_x^{\text{receive}}(\mathbf{r}) \sin(\omega_0 t - \phi_0(\mathbf{r})) \\ + \mathcal{B}_y^{\text{receive}}(\mathbf{r}) \cos(\omega_0 t - \phi_0(\mathbf{r}))] d\mathbf{r}. \end{aligned} \quad (1.10)$$

By using the parameterization

$$\mathcal{B}_x^{\text{receive}} \equiv \mathcal{B}_\perp \cos \theta_\mathcal{B}, \quad \mathcal{B}_y^{\text{receive}} \equiv \mathcal{B}_\perp \sin \theta_\mathcal{B}, \quad (1.11)$$

the signal can be reduced to

$$\text{signal} \propto \omega_0 \int e^{-t/T_2(\mathbf{r})} M_\perp(\mathbf{r}, 0) \mathcal{B}_\perp(\mathbf{r}) \sin(\omega_0 t + \theta_\mathcal{B} - \phi_0(\mathbf{r})) d\mathbf{r}. \quad (1.12)$$

By using a second receive coil placed perpendicular to the first one, the signal can be measured with a $\pi/2$ phase difference, or *phase lag*. In this case the signal received by the second coil is similar to the one above, but the sine is replaced by a cosine. Combining both signals gives

$$s(t) \propto \omega_0 \int e^{-t/T_2(\mathbf{r})} M_{\perp}(\mathbf{r}, 0) \mathcal{B}_{\perp}(\mathbf{r}) e^{-i(\omega_0 t + \theta_{\mathcal{B}} - \phi_0(\mathbf{r}))} d\mathbf{r}, \quad (1.13)$$

where $s(t) = s_{re}(t) + is_{im}(t)$ is the complex signal. For clarity and educational purposes, assume that the static magnetic field and transmitting and receiving rf coils are uniform, such that ϕ_0 , $\theta_{\mathcal{B}}$ and \mathcal{B}_{\perp} are independent of position. Let Λ be a constant of proportionality and neglect the relaxation effects, then

$$s(t) = \omega_0 \Lambda \mathcal{B}_{\perp} \int M_{\perp}(\mathbf{r}, 0) e^{-i\omega_0 t} d\mathbf{r}. \quad (1.14)$$

In practice, T_1 decay can be ignored if the time between excitations is sufficiently long, e.g. if the net magnetization can fully recover. Furthermore, traverse relaxation usually dominated by dephasing resulting from static field inhomogenities, characterised by T_2' .² Using a special combination of RF pulses T_2' dephasing can be reversed and the signal is temporarily recovered (an *echo*). As long as the sampling time T_s and/or the echo time T_E are sufficiently short the T_2 relaxation can be neglected as well.

Finally, notice that if the equilibrium magnetization M_0 is independent of \mathbf{r} and if V_s is the sample volume then

$$|s(t)| = \omega_0 M_0 \Lambda \mathcal{B}_{\perp} V_s. \quad (1.15)$$

Since both ω_0 and M_0 are proportional to B_0 , the above equality shows that the signal strength is quadratic in the magnetic field strength. It turns out that the noise is approximately linear in B_0 for $B_0 > 0.5\text{T}$, such that the *signal-to-noise* ratio (SNR) is linear in the field strength. This is one of the many reasons why high-field MRI scanners are of interest.

1.3 Fourier imaging

We have seen that the precession frequency of spins is proportional to field strength. The idea of magnetic resonance imaging is to manipulate the magnetic field in such a way that spatial information is encoded in the precession frequency of the spins. Let us formalize this notion in the following section.

Let us assume, in addition to the assumptions made in the previous section, that we can achieve a perfect $\pi/2$ flip angle and that excitation is instantaneous (e.g. no relaxation occurs during the application of the pulse). Then $M_{\perp}(\mathbf{r}, 0) = M_0(\mathbf{r})$ and we can generalize (1.2) in order to define the *effective spin density*

$$\rho(\mathbf{r}) \equiv \omega_0 \Lambda \mathcal{B}_{\perp} M_0(\mathbf{r}) = \omega_0 \Lambda \mathcal{B}_{\perp} \frac{B_0 \gamma^2 \hbar^2}{4kT} \rho_0(\mathbf{r}). \quad (1.16)$$

² T_2 and T_2' relaxation “add up” to T_2^* relaxation, such that $e^{-t/T_2^*} \equiv e^{-t/T_2} e^{-t/T_2'}$ where $1/T_2^* \equiv 1/T_2 + 1/T_2'$.

In practice one may need to solve the Bloch equation numerically in order to determine the exact value of $M_{\perp}(\mathbf{r}, 0)$.

Let $\alpha \in \{x, y, z\}$ in the following part. Suppose we have the spatially linear varying field

$$B(\mathbf{r}, t) = B_0 + xG_x(t) + yG_y(t) + zG_z(t). \quad (1.17)$$

The quantity G_{α} is the (spatially) constant gradient in de α direction, i.e. $G_{\alpha} \equiv \partial B / \partial \alpha$. Define the angular frequency of the spins by

$$\omega(\mathbf{r}, t) \equiv \gamma B(\mathbf{r}, t) = \omega_0 + \omega_G(\mathbf{r}, t), \quad (1.18)$$

where ω_0 is the Larmor frequency from (1.1). The accumulated phase $\phi(\mathbf{r}, t)$ at position \mathbf{r} and up to time t is given by

$$\phi(\mathbf{r}, t) \equiv - \int_0^t \omega(\mathbf{r}, \tau) d\tau. \quad (1.19)$$

Using the effective spin density and the accumulated phase, (1.14) can be adapted to find

$$s(t) = \int \rho(\mathbf{r}) e^{i\phi(\mathbf{r}, t)} d\mathbf{r}, \quad (1.20)$$

where small deviations of ω_0 in front of the integral, but not in the phase, are ignored.

Denote the spatial frequency by $\mathbf{k} = \mathbf{k}(t) = (k_x(t), k_y(t), k_z(t))$, where k_{α} is defined as

$$k_{\alpha}(t) \equiv \frac{\gamma}{2\pi} \int_0^t G_{\alpha}(\tau) d\tau. \quad (1.21)$$

Then the accumulated phase up to time t , due to the applied gradient can be written as

$$\phi_G(\mathbf{r}, t) = - \int_0^t \omega_G(\mathbf{r}, \tau) d\tau \quad (1.22)$$

$$= -2\pi [xk_x(t) + yk_y(t) + zk_z(t)] \quad (1.23)$$

$$= -2\pi \mathbf{k} \cdot \mathbf{r}. \quad (1.24)$$

We can use the identity above to write (1.20) as

$$s(\mathbf{k}) = e^{-i\omega_0 t} \int \rho(\mathbf{r}) e^{-i2\pi \mathbf{k} \cdot \mathbf{r}} d\mathbf{r}. \quad (1.25)$$

Remember that t is hidden in \mathbf{k} . The rapid oscillations associated with the Larmor frequency can be removed by considering the signal from the rotating frame of reference:

$$\sigma(\mathbf{k}) \equiv e^{i\omega_0 t} s(\mathbf{k}) = \int \rho(\mathbf{r}) e^{-i2\pi \mathbf{k} \cdot \mathbf{r}} d\mathbf{r}. \quad (1.26)$$

Now it is easy to see that the signal is a Fourier transform of the effective spin density. Hence

$$\rho(\mathbf{r}) = \int \sigma(\mathbf{k}) e^{+i2\pi \mathbf{k} \cdot \mathbf{r}} d\mathbf{k}. \quad (1.27)$$

The next section will show how the gradients can be used to encode spatial information in a desired way.

1.4 Basic k -space traversal

The k -space can be covered in virtually any way one can think of. In fact, there are entire books devoted to k -space trajectories, their properties, advantages and disadvantages. However, this section is only about the basics. Let us focus on Cartesian k -space coverage, e.g. the k -space samples are taken on an rectangular grid. Roughly speaking, spatial information along the z -axis is encoded using *slice selection*, along the y -axis using *phase encoding* and along the x -axis using *frequency encoding*. These subjects will be discussed next.

1.4.1 Slice selection

Slice selection exploits the on-resonance condition to excite spins in specific locations. Recall that $\omega(\mathbf{r}) = \gamma B(\mathbf{r})$. During excitation a linear gradient is superimposed on the static field such that $\omega(\mathbf{r})$ varies linearly along the selection axis, usually the z -axis. When the RF pulse is applied, only the spins precessing with the frequency of the spin are excited. Therefore, a pulse with a sufficiently narrow bandwidth will only excite a slice of the sample.

On a side note, the desired profile of the RF pulse is a top hat function in frequency space. This translates to a sinc in time domain. Since the sinc has unbounded support, truncation of the signal is necessary, leading to Gibbs' phenomenon. In practice this truncation results in (some) excitation of spins beyond the desired slice and can result in *cross-talk* between imaged slices. Cross-talk can be avoided, for example, by leaving some room between slices, replacing the top hat function by a different window such as the Hann window or by imaging slices in a non sequential order.

The careful reader may have noticed that spins within a single slice precess at different frequencies as a result of the applied gradient and the finite thickness of a slice. This difference is problematic when we try to encode spatial information in other directions and leads to dephasing and signal loss. Consider the following example³: a constant slice selection gradient $G_{ss} < 0$ is applied for duration T_{ss} and spins are instantaneously excited at time $T_{ss}/2$. Suppose we select a slice of thickness Δz and that there are two spins, spin s_- at $z_0 - \Delta z/2$ and spin s_+ at $z_0 + \Delta z/2$. Then according to (1.22) $\phi_G(s_{\pm}, t) = (z_0 \pm \Delta z)G_{ss}t$ and at time T_{ss} the two spins have a phase difference of $\Delta z G_{ss} T_{ss}$. The spins can be *rephased* by applying a positive gradient after the first gradient. For example, constant gradient $-G_{ss}$ is applied for a duration of $T_{ss}/2$. Remark that the same result is achieved by applying a stronger/weaker gradient for a shorter/longer duration respectively.

1.4.2 Phase and frequency encoding

Phase encoding occurs between slice selection and frequency encoding. After slice selection a constant gradient G_y is applied for time τ_y along the y -axis. Phase accumulates along the y -axis and once the gradient is turned off we have $k_y = \gamma/(2\pi) G_y \tau_y$. It is common to keep τ_y fixed and to vary G_y with steps ΔG_y between measurements.

Now that k_z and k_y are fixed with slice selection and phase encoding respectively, it is possible to obtain samples along k_x using frequency encoding. The idea is to

³This example also gives intuitive insight in dephasing caused by static field inhomogeneities (T_2' relaxation).

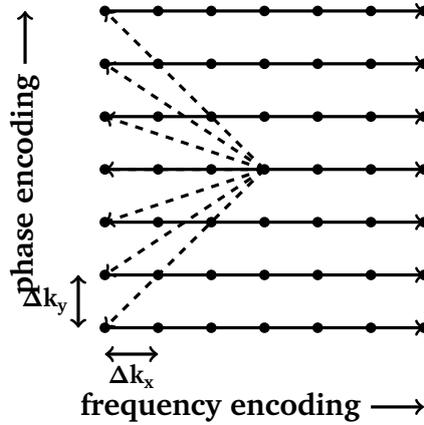


Figure 1.3: It is common to traverse a slice one line at a time. After excitation the phase encoding gradient is used to move up or down and a negative frequency encoding gradient is used to move left in the k -space. Once arrived at the desired starting point, previous gradients are turned off, a positive frequency encoding gradient is applied and the sampling begins.

apply a gradient along the x -axis during sampling. This way k_x advances between each sample point. The process of obtaining k_x samples is often called *read-out* and the frequency encoding gradient is often called the *read* gradient.

To obtain samples from $-k_{x,max}$ to $+k_{x,max}$ a negative gradient is applied in order to move from $k_x = 0$ to $-k_{x,max}$. During this time no samples are taken and spins along the x -axis dephase. Once the $-k_{x,max}$ is reached a positive read gradient is applied until $k_x = +k_{x,max}$ and samples are taken. Spins along the x -axis rephase and at some point all spins are in phase, e.g. $k_x = 0$. When $k_x = 0$ for a second time the *gradient echo* is obtained. This point in time is usually called T_E . If T_s is the sample time, e.g. the duration the positive read gradient is applied, and we define $t' \equiv t - T_E$, then $k_x = \gamma/(2\pi) G_x t'$. Putting everything together the measured signal can be written as

$$\sigma(t', G_y) = \iint \left[\int_{z_0 - \frac{\Delta z}{2}}^{z_0 + \frac{\Delta z}{2}} \rho(x, y, z) dz \right] e^{-2\pi i \frac{\gamma}{2\pi} G_y \tau_y y} e^{-2\pi i \frac{\gamma}{2\pi} G_x t' x} dy dx \quad (1.28)$$

for $-T_s/2 < t' < T_s/2$. See also [Figure 1.3](#).

It is interesting to note that volumes can be imaged directly, e.g. without slice selection, using phase encoding along both the z and y axes. In fact it is possible to use phase encoding in all three dimensions. This type of encoding is of interest in *single point imaging* and allows sample points to be acquired in the absence of artifacts (such as shifts or blurring in the image) caused by the read-out gradient or the magnetic properties of the samples.

1.4.3 Sampling density

It is important to consider the sample density and the number of samples necessary for reconstruction. Consider first the one dimensional case. Suppose ρ describes an object of width A centered at the origin. A can also be the width of the sensitivity of

the experiment. Clearly σ has bandwidth $A/2$. By the Shannon-Whittaker theorem, ρ can be reconstructed if $\Delta k \leq 1/A$. If $L \equiv 1/\Delta k$ is the *field-of-view* (FOV), then the previous statement can be written as $L \geq A$. If we approximate the signal using N samples then $\Delta x \equiv L/N = 1/(N\Delta k) = 1/(2k_{max})$ is the spatial *resolution*. Usually samples are indexed asymmetrically from $-n$ to $n - 1$ in order to keep an even number of points. In the two dimensional case similar results hold and $\Delta k_y = \gamma/(2\pi)\Delta G_y\tau_y$ and $\Delta x = \gamma/(2\pi)G_x\Delta t$, where Δt is the time between samples during readout.

1.5 Radial k -space coverage

As previously stated there are various ways of covering the k -space. One of these ways which is promising for high speed MRI is radial coverage. In radial k -space coverage samples are read along lines originating from the origin. In three dimensions these lines make up a pincushion/kooshball, but radial coverage can be combined with slice selection as well.

With the x , y and z gradients turned on, the combined gradient can be derived from (1.17) to be

$$\mathbf{G}(t) = D\mathbf{B}(\mathbf{r}, t) = (G_x(t), G_y(t), G_z(t)) \quad (1.29)$$

for any \mathbf{r} . Thus, in order to orient the read gradient \mathbf{G} in the desired direction we can simply use spherical coordinates, e.g.

$$G_x(t) = G(t) \cos \theta \sin \phi \quad (1.30)$$

$$G_y(t) = G(t) \sin \theta \sin \phi \quad (1.31)$$

$$G_z(t) = G(t) \cos \phi, \quad (1.32)$$

with $G(t) \in [0, G_{max})$, $\theta \in [0, 2\pi)$ and $\phi \in [0, \pi)$.

Requirement for Δk_r , Δk_θ and Δk_ϕ can be derived in a similar manner as requirements for Δk_x , Δk_y and Δk_z . First notice that $\Delta k_\alpha = k\Delta\alpha \leq k_{max}\Delta\alpha$ for $\alpha \in \{\theta, \phi\}$. To prevent aliasing, the largest angular step $\Delta\alpha$ must obey $k_{max}\Delta\alpha \equiv 1/L$. The radial step must satisfy $\Delta k_r = 1/L$.

Reconstruction in radial k -space coverage is closely related to reconstruction in computerized tomography. Define $s_\alpha = \sin \alpha$ and $c_\alpha = \cos \alpha$ and the rotated basis vectors

$$\mathbf{u} = \begin{bmatrix} c_\theta s_\phi \\ s_\theta s_\phi \\ c_\phi \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} s_{-\theta} \\ c_{-\theta} \\ 0 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} -c_\theta c_\phi \\ -s_\theta c_\phi \\ s_\phi \end{bmatrix}. \quad (1.33)$$

Then the three dimensional *Radon transform* is given by

$$p_{\theta,\phi}(\xi) = \int \int \rho(\xi\mathbf{u} + \eta\mathbf{v} + \zeta\mathbf{w}) d\eta d\zeta. \quad (1.34)$$

If point on a line through the origin is written as $\xi\mathbf{u}$, then $P_{\theta,\phi}$ is the projection of the plane through $\xi\mathbf{u}$ with its normal parallel to $\xi\mathbf{u}$. Substituting \mathbf{k} by $\xi\mathbf{u} + \eta\mathbf{v} + \zeta\mathbf{w}$, \mathbf{r} by $\mu_1\mathbf{u} + \mu_2\mathbf{v} + \mu_3\mathbf{w}$ and using the fact that \mathbf{u} , \mathbf{v} and \mathbf{w} are mutually orthonormal

we obtain from (1.26)

$$\sigma(\mu_1 \mathbf{u} + \mu_2 \mathbf{v} + \mu_3 \mathbf{w}) = \int \int \int \rho(\xi \mathbf{u} + \eta \mathbf{v} + \zeta \mathbf{w}) e^{-2\pi i(\xi \mu_1 + \eta \mu_2 + \zeta \mu_3)} d\xi d\eta d\zeta. \quad (1.35)$$

In particular if $\mu_2 = \mu_3 = 0$ and $\mu \equiv \mu_1$ then

$$\sigma(\mu \mathbf{u}) = \int p_{\theta, \phi}(\xi) e^{-2\pi i \xi \mu} d\xi = \widehat{P}_{\theta, \phi}(\mu). \quad (1.36)$$

Therefore, in radial k -space coverage the signal is the one dimensional Fourier transform of the Radon transform. This result is closely related to computerized tomography where the measured quantity is (logarithmically) proportional to the two dimensional Radon transform $P_{\theta, 0}$. In this case $P_{\theta, 0}(\xi)$ is the projection of the energy absorption from an X-ray through $\xi \mathbf{u}$ and σ would be the two dimensional Fourier transform of the tissue density ρ , see [3, Chapter 9].

In radial imaging θ and ϕ determine the direction of the read gradient and μ determines a point along the read gradient. Clearly, k -space points are obtained on a grid that is rectangular in spherical coordinates whereas the images we would like to have are rectangular in Cartesian coordinates. A popular approach is to interpolate measured k -space points to a Cartesian grid and apply an inverse Fourier transform.

A more accurate result may be obtained by expressing ρ as the inverse Fourier transform of σ in spherical coordinates. With the substitution of variables $\mathbf{k} = k\mathbf{u}$ (1.27) can be written as

$$\rho(x, y, z) = \int_0^\pi \int_0^\pi \int_{-\infty}^\infty k^2 \sin \phi \sigma(k\mathbf{u}) e^{+2\pi i k \mathbf{u} \cdot \mathbf{r}} dk d\phi d\theta. \quad (1.37)$$

Observe that $\theta \in [0, \pi)$ and $k \in (-\infty, \infty)$. Let $\mathbf{r} = r(c_\alpha s_\beta, s_\alpha s_\beta, c_\beta)$ and define $\xi \equiv \mathbf{u} \cdot \mathbf{r}$. Notice that this definition of ξ is consistent with the use of ξ above; $\xi \mathbf{u}$ is the projection of \mathbf{r} onto \mathbf{u} . The above equation can now be written as

$$\rho(rc_\alpha s_\beta, rs_\alpha s_\beta, rc_\beta) = \int_0^\pi \int_0^\pi \left[\int_{-\infty}^\infty k^2 \sin \phi \widehat{p}_{\theta, \phi}(k) e^{+2\pi i k \xi} dk \right] d\phi d\theta \quad (1.38)$$

Denote the expression between the square brackets as $\widetilde{p}_{\theta, \phi}$. The map $p_{\theta, \phi} \rightsquigarrow \widetilde{p}_{\theta, \phi}$ can be seen as a filter operation: $\widetilde{p}_{\theta, \phi}$ is the inverse Fourier transform of $\widehat{p}_{\theta, \phi} H_\phi$, with k -space function $H_\phi(k) = k^2 \sin \phi$.

References

- [1] E. Mark Haacke et al. *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. John Wiley & Sons, Inc., 1999 (cit. on pp. 5, 7).
- [2] Philips. *Basic principles of MR imaging*. Koninklijke Philips Electronics N.V., 2010 (cit. on p. 5).
- [3] G.L.G. Sleijpen. Tech. rep. Utrecht University: Department of Mathematics, Aug. 2008 (cit. on p. 14).

Chapter 2

Numerical methods for Compressed Sensing

2.1 Introduction

In many applications sampling by the Nyquist rate is impractical or infeasible as a result of, e.g., hardware, software or time constraints. Recently the theory of *compressed sensing* (CS), also known as *compressive sensing*, has gained a lot of attention since it allows sub-Nyquist sampling under mild constraints. In a single slogan CS is to “do more with less”. In case of Magnetic Resonance Imaging, sub-Nyquist sampling means shorter scan duration and therefore increased patient comfort. Ideally (near) real-time imaging would be possible with CS, enabling the use of MRI in interventional applications.

Section 2.2 introduces common notation, describes key terms and provides a more formal statement of CS. **Section 2.3** gives an (incomplete) overview of existing methods and techniques. The overview is followed by numerical tests and an interpretation of the results.

2.2 Mathematical background

Let $x \in \mathbb{C}^n$. Define the support of x as $\text{supp}(x) \equiv \{i : x_i \neq 0\}$ and introduce $\|x\|_0 \equiv |\text{supp}(x)|$ as shorthand for the cardinality of $\text{supp}(x)$. The function $\|\cdot\|_0$ is commonly referred to as the ℓ_0 -norm, despite the fact it is not a true norm. x is said to be s -sparse for some $s \in \mathbb{N}_0$ if $\|x\|_0 \leq s$ and $\Sigma_s \equiv \{x \in \mathbb{C}^n : \|x\|_0 \leq s\}$ is the set of all s -sparse vectors. Let $x_s \in \Sigma_s$ denote an s -term approximation of x where all but the s largest absolute coordinates of x are set to zero and ties arbitrarily broken. Note that x_s is the best s -term approximation to x for any ℓ_p -norm and $0 < p \leq \infty$. If x is s -sparse for some $s \ll n$ then x is called *sparse* otherwise x is called *compressible* if $\|x - x_s\|_p$ for some $0 < p \leq \infty$ quickly decays to zero as s increases.

Let $x \in \mathbb{C}^n$ be the signal we wish to reconstruct. Suppose we perform $m < n$ measurements such that each measurement is an inproduct with x . In matrix notation we obtain

$$b = Ax, \tag{2.1}$$

where $b \in \mathbb{C}^m$ are the measurements and $A \in \mathbb{C}^{m \times n}$ is the measurement matrix. Since $m < n$ the problem is underdetermined and no unique solution exists. However, if x is sparse then a unique solution might be obtained by solving

$$\operatorname{argmin}_x \|x\|_0 \quad \text{subject to} \quad b = Ax. \quad (2.2)$$

The only constraint is that x and A satisfy the conditions in the following lemma.

Theorem 1. *Let $A \in \mathbb{C}^{m \times n}$ and $s \in \mathbb{N}$ with $2s \leq m$. Suppose that every set of $2s$ columns of A are linearly independent. If there exists an s -sparse solution to $Ax = b$, it is unique.*

Proof. Suppose two s -sparse solutions $x \neq x'$ exist such that $b = Ax = Ax'$. It follows that $2s$ columns of A are linearly dependent since $A(x - x') = 0$ and $x - x' \neq 0$ is at most $2s$ -sparse. Contradiction. \square

Solving the ℓ_0 -minimization problem in (2.2), and thus finding the nonzero coordinates, is a combinatorial problem in general. However, if A satisfies the conditions in **Theorem 2** below then the ℓ_0 -minimization problem is equivalent to the ℓ_1 -minimization problem

$$\operatorname{argmin}_x \|x\|_1 \quad \text{subject to} \quad b = Ax. \quad (BP)$$

The problem above is often called *Basis Pursuit* (BP) and is a convex optimization problem. The name Basis Pursuit has its roots in linear programming, though, in our setting it may be more appropriate to call the problem *Support Pursuit* since we wish to find the support, the non-zero coefficients, of a signal.

Before stating **Theorem 2** let us introduce the *restricted isometry property* (RIP). With A as before the s -restricted isometry constant δ_s is defined as

$$\delta_s \equiv \sup \left\{ \left| 1 - \frac{\|Ax\|_2^2}{\|x\|_2^2} \right| : x \in \Sigma_s \text{ and } x \neq 0 \right\}. \quad (2.3)$$

The constant δ_s can be interpreted as a the ‘distance’ from orthonormality for every set of at most s columns of A . If $\delta_s < 1$ then A is said to satisfy the RIP of order s . If $\delta_{2s} < 1$ then by **Theorem 1** any s -sparse solution is unique. We are now ready for the following theorem, for which a proof can be found in Candès et. al. [9].

Theorem 2. *Assume that $\delta_{2s} < \sqrt{2} - 1$. Then the solution x_* to (BP) obeys*

$$\|x_* - x\|_1 \leq C_0 \|x - x_s\|_1$$

and

$$\|x_* - x\|_2 \leq C_0 s^{-1/2} \|x - x_s\|_1$$

with $C_0 = 2(1 + \rho)(1 - \rho)^{-1}$ and $\rho \equiv \sqrt{2}\delta_{2s}(1 - \delta_{2s})^{-1}$. In particular, if x is s -sparse, the recovery is exact.

In many applications the measurements are contaminated with noise, e.g.

$$b = Ax + z, \quad (2.4)$$

for some unknown noise vector z . In this case, it may be more suitable to solve the convex optimization problem

$$\operatorname{argmin}_x \|x\|_1 \quad \text{subject to} \quad \|b - Ax\|_2 \leq \sigma, \quad (BP_\sigma)$$

for some $\sigma \geq \|z\|_2$. This problem is also known as *basis pursuit denoising* and results similar to the ones for basis pursuit hold, see the following theorem. Refer again to Candès et. al. [9] for a proof.

Theorem 3. Assume that $\delta_{2s} < \sqrt{2} - 1$ and $\|z\|_2 \leq \sigma$. Then the solution x_* to (BP_σ) obeys

$$\|x_* - x\|_2 \leq C_0 s^{-1/2} \|x - x_s\|_1 + C_1 \sigma, \quad (2.5)$$

with C_0 and ρ as in [Theorem 2](#), $C_1 = 2\alpha(1 - \rho)^{-1}$ and $\alpha \equiv 2\sqrt{1 + \delta_{2s}}(1 - \delta_{2s})^{-1}$.

Although C_0 and C_1 grow asymptotically if δ_{2s} approaches $\sqrt{2} - 1$, the constants C_0 and C_1 are generally reasonably small. For example if $\delta_{2s} = 0.2$ then $C_0 \approx 4.2$ and $C_1 \approx 8.5$.

Ideally one would like to construct a matrix A such that the bounds in [Theorem 2](#) and [Theorem 3](#) are satisfied for large values of s in the order of m . It turns out to be difficult to design matrices that provably satisfy the former properties. It can be shown that certain types of random matrices are highly likely to satisfy the desired properties under certain conditions, e.g., see [8, 11] and references therein. The following are examples.

- *Gaussian matrices.* Entries of A are independently sampled from the normal distribution with mean zero and variance $1/m$.
- *Bernoulli matrices.* The entries of a Bernoulli matrix are independently chosen from $\{-1/\sqrt{m}, +1/\sqrt{m}\}$ with uniform probability.
- *Fourier matrices.* The partial Fourier matrix A obtained by randomly choosing m rows from the full Fourier matrix.
- *Incoherent matrices.* The random selection of m rows of an orthonormal matrix.

Observe that a partial Fourier matrix is an incoherent matrix, another example is a partial wavelet matrix (if an orthonormal wavelet is used). In theory partial Fourier matrices and incoherent matrices are sometimes rescaled or their columns are renormalized. This is not always done, or necessary, in practice.

Experiments repeatedly show that the RIP is too rigid and restrictive, despite strong results. Recent research includes a RIPless theory for CS [7], based on concepts such as a local RIP and a weak RIP. Many researchers are more pragmatic in practice and use measurement matrices loosely based on theory and with a particular “structure” that experiments show works well (e.g., randomised or incoherent measurements). We will use the same approach for our numerical experiments in [Section 2.4](#).

There are effective CS reconstruction algorithms based on the ℓ_1 -regularized minimization problem

$$\operatorname{argmin}_x \left\{ \frac{1}{2} \|b - Ax\|_2^2 + \lambda \|x\|_1 \right\}, \quad (QP_\lambda)$$

which is closely connected to quadratic programming. Incidentally this problem is sometimes also called basis pursuit denoising. Another related problem is *Least Absolute Shrinkage and Selection Operator* (LASSO):

$$\operatorname{argmin}_x \|b - Ax\|_2 \quad \text{subject to} \quad \|x\|_1 \leq \tau. \quad (LS_\tau)$$

The latter is important for a reconstruction algorithm called *Spectral Projected Gradient* for ℓ_1 minimization, see [Section 2.3.3.4](#).

2.3 Numerical methods

This section gives an overview of existing CS reconstruction algorithms. Please keep in mind that this is by no means an in-depth or complete overview. Notable absent algorithms include “the” *Homotopy* algorithm, its “little brother” *Least Angle Regression* (LARS) and *Iteratively Reweighted Least Squares* (IRLS), a nice overview of these three can be found in [11]. In addition, CS is an active research area and it is almost impossible to keep up with the newest developments.

In this section we will see two classes of algorithms for CS reconstruction. The first class consists of *greedy* algorithms, which are further described in [Section 2.3.2](#). At each stage a greedy algorithm always takes the “best” choice based on current information. Most of these algorithms are variations of *Orthogonal Matching Pursuit*, see [Section 2.3.2.1](#) through [Section 2.3.2.3](#). The second class consists of *convex relaxation* algorithms, which is the subject of [Section 2.3.3](#). A relaxation algorithm is an iterative method for solving a system of equations. The algorithms are called convex since (regularized) ℓ_1 -minimization is a convex problem. Examples include *Nonlinear Conjugate Gradients* in [Section 2.3.3.5](#) and the *Fast Iterative Shrinkage-Thresholding* in [Section 2.3.3.3](#).

The line between both classes is inherently fuzzy. We will see that, generally speaking, greedy algorithms try to find nonzero coordinates and solve a reduced problem, whereas convex relaxation methods are usually gradient based methods that compute elaborate step-sizes and use specific projections to enforce sparsity.

2.3.1 Key ingredients

We introduce the following useful notation. For an n -dimensional vector x let $\Gamma \subset \{1, 2, \dots, n\}$ be a subset. Then x_Γ is defined as the vector consisting of coordinates of x indexed by Γ . Define A_Γ as the matrix consisting of columns of the $m \times n$ matrix A indexed by Γ . In MATLAB notation: $x_\Gamma \equiv x(\Gamma)$ and $A_\Gamma \equiv A(:, \Gamma)$. And in the text a variable of the form $x^{(k)}$ shall mean “ x as computed in iteration k of the algorithm”.

Many greedy solvers use the minimum norm least squares solution of the reduced system $A_\Gamma x_\Gamma = b$ for $|\Gamma| \leq m$; it should be clear that if $\Gamma = \text{supp}(x_*)$, then the solution x is to $A_\Gamma x_\Gamma = b$ is sparse and minimizes $\|b - Ax\|_2$. Most greedy methods try to identify non-zero coordinates and iteratively grow or improve a set Γ of indices. Since the actual non-zero coordinates are unknown, algorithms choose candidates based on $A^T r^{(k)}$, where $r^{(k)} \equiv b - Ax^{(k)}$ is some intermediate residual. Assuming there exists a sparse solution, an elegant motivation is provided by Needell and Tropp 2008 [15]. Suppose we have an s -sparse signal x and measurement matrix A with a small RIP constant, e.g. $\delta_s \ll 1$. Then the energy in each set of s coordinates of $A^T b = A^T Ax$ approximates the energy in the corresponding s coordinates of x since $x^T A^T b = x^T A^T Ax = \|Ax\|_2^2 \approx \|x\|_2^2$. Hopefully, the s largest coordinates of $A^T b$ indicate the s largest coordinates of x . We can apply the same idea to the residuals $r^{(k)} = b - Ax^{(k)} = A(x - x^{(k)})$ in an attempt to find the largest non-zero coordinates of $x - x^{(k)}$ based on $r^{(k)}$. In addition, $\|b - Ax\|_2^2$ is minimal if and only if $b - Ax \perp AC^n$ which implies $A^T(b - Ax) = A^T r = 0$. As an alternatively

one can see $A^T r^{(k)}$ as the gradient of $1/2 \|b - Ax^{(k)}\|_2^2$ and the largest coordinate as an approximation to the direction of steepest descent.

Another key ingredient is the *hard-thresholding* operator $H_s(x)$ that keeps the s largest (in magnitude) coordinates of x and replaces the rest by zero. There is also the *soft-thresholding* or *shrinkage* operator $s_\lambda(y)$ defined by

$$s_\lambda(y)_j \equiv e^{i(\text{Arg } y_j)} \max\{|y_j| - \lambda, 0\}, \quad (2.6)$$

for a complex vector $y \in \mathbb{C}^n$ and $j = 1, \dots, n$. Application of $s_\lambda(y)$ “shrinks” the magnitude of every nonzero coordinate in y by λ . It turns out that $s_\lambda(y)$ is the unique solution of

$$s_\lambda(y) \equiv \underset{x}{\operatorname{argmin}} \left\{ \frac{1}{2} \|x - y\|_2^2 + \lambda \|x\|_1 \right\}. \quad (2.7)$$

The shrinkage operator is a special class of the *proximal map*

$$\operatorname{prox}_L(g)(y) \equiv \underset{x}{\operatorname{argmin}} \left\{ \frac{L}{2} \|x - y\|_2^2 + g(x) \right\}, \quad (2.8)$$

such that $s_\lambda(y) = \operatorname{prox}_1(\lambda \|x\|_1)(y)$. Although we define the shrinkage operator and proximal map for complex input, the algorithms that use them (e.g., Fixed Point Continuation in [Section 2.3.3.2](#) and Fast Iterative-Shrinkage Thresholding in [Section 2.3.3.3](#)) are designed for real valued signals and it is unclear if these algorithms can successfully reconstruct complex valued signals.

2.3.2 Greedy algorithms

2.3.2.1 Orthogonal Matching Pursuit

The first algorithm we will look at for solving *(BP)* is a straightforward algorithm called *Orthogonal Matching Pursuit* (OMP). Simply put, in each iteration OMP expands the set of nonzero coordinates, selects the corresponding columns from A and solves a reduced linear system in a least squares sense.

Algorithm 1: Orthogonal Matching Pursuit

Input: Measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m < n$, measurements $b \in \mathbb{C}^m$, maximum number of iterations $k_{max} \leq m$ and tolerance $\epsilon \geq 0$.

Output: x such that $x \approx \operatorname{argmin}_{\hat{x}} \|\hat{x}\|_0$ subject to $A\hat{x} = b$.

begin

```

     $\Gamma \leftarrow \emptyset, x \leftarrow 0, r \leftarrow b$ 
    for  $k \leftarrow 1, 2, \dots, k_{max}$  and  $\|r\|_2 > \epsilon \|b\|_2$  do
         $a \leftarrow A^T r$ 
        if  $a = 0$  then break
         $i_{max} \leftarrow \operatorname{argmax}_{i \notin \Gamma} |a_i|$ 
         $\Gamma \leftarrow \Gamma \cup \{i_{max}\}$ 
         $x_\Gamma \leftarrow$  least squares solution of  $A_\Gamma x_\Gamma = b$ 
         $r \leftarrow b - Ax$ 

```

[Algorithm 1](#) gives a precise description of OMP. Needless to say, solving the least squares sub-problem can be expensive. Possible approaches include a QR

factorization that is updated for each coordinate added to the support, or the use of iterative approximation algorithms such as LSQR. Reasonable values for tolerance ϵ inherently depend on properties of the underlying problem. For example, A could be well conditioned (e.g. a partial Fourier matrix) or have small singular values, b might be contaminated with any amount of noise and x can have millions or only hundreds of coordinates.¹

OMP is based on a sparse approximation algorithm and was developed by Tropp and Gilbert [17]. They state and prove the following relevant theorem.

Theorem 4. Fix $\delta \in (0, 0.36)$, and choose $m \geq Ks \log(n/\delta)$. Suppose that x is an arbitrary s -sparse signal in \mathbb{R}^n . Draw the m rows of A randomly from the standard Gaussian distribution on \mathbb{R}^n . Given the data $b = Ax$, Orthogonal Matching Pursuit can reconstruct the signal with probability exceeding $1 - 2\delta$. For this theoretical result, it suffices that $K = 20$. When m approaches infinity, it suffices to take $K \approx 4$.

In short, this theorem states that for sufficiently many measurements m an s -sparse signal can be recovered with high probability if A is a Gaussian matrix. Tropp and Gilbert also state the following theorem that shows a similar result holds when the measurement matrix is a partial Fourier matrix, though, they attribute the proof to Kunis and Rauhut (see [17] for proper references).

Theorem 5. Fix $\delta \in (0, 1)$. Let $m \geq Ks \log(n/\delta)$, and suppose that Γ is a subset of $\{1, 2, \dots, n\}$ with cardinality s . Independent from Γ , draw a matrix A consisting of m random rows from the n -dimensional DFT matrix. With probability at least $1 - \delta$, OMP can recover every vector x supported on Γ from the data $b = Ax$.

If there is noise or if the signal is compressible instead of sparse, the results above no longer hold. However, similar behaviour is expected to hold in practice. Finally, Tropp and Gilbert remark the following important difference between OMP and basis pursuit. If A satisfies the conditions in Theorem 2, then for all s -sparse x the solution to $Ax = b$ can be found by solving (BP). Theorem 4 and Theorem 5 make no such guarantees for all s -sparse x , not even if A satisfies the conditions of Theorem 2. Instead the two theorems state that OMP can reconstruct each signal with high probability (but will likely fail to recover all sparse signals).

2.3.2.2 Stagewise Orthogonal Matching Pursuit

While OMP is a straightforward algorithm, it is also a slow one. The main downside of OMP is that only a single coordinate is added to the support in each iteration. To overcome this downside, Donoho proposed *Stagewise Orthogonal Matching Pursuit* (StOMP) [10]. Donoho calls iterations *stages* in StOMP and in each stage multiple coordinates can enter the support of x . Only a small number of stages, e.g., 10, are required as a result.

In the derivation of StOMP, Donoho considers the vector $A^T b$ as a noised representation of x and $A^T r$ is the noised part of the signal that has not yet been approximated. In fact StOMP's slogan is: "noiseless underdetermined problems behave like noisy well-determined problems".

StOMP is described in Algorithm 2. The performance of StOMP highly depends on the strategy for the selection of the threshold. If the threshold is too high, the algorithm will converge slowly. If the threshold is too low, a solution with too many

¹For the noise free phase diagrams in Section 2.4.1 $\epsilon = 10^{-5}$ was used by default.

Algorithm 2: Stagewise Orthogonal Matching Pursuit

Input: Measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m < n$, measurements $b \in \mathbb{C}^m$, maximum number of stages k_{max} and tolerance $\epsilon \geq 0$.

Output: x such that $x \approx \operatorname{argmin}_{\hat{x}} \|\hat{x}\|_0$ subject to $A\hat{x} = b$.

begin

```
 $\Gamma \leftarrow \emptyset, x \leftarrow 0, r \leftarrow b$   
for  $k \leftarrow 1, 2, \dots, k_{max}$  and  $\|r\|_2 > \epsilon \|b\|_2$  do  
   $a \leftarrow A^T r$   
   $t \leftarrow \text{threshold}$   
   $\sigma \leftarrow \|r\|_2 / \sqrt{m}$   
   $\Gamma' \leftarrow \Gamma \cup \{i : |a_i| > t\sigma\}$   
  if  $\Gamma' = \Gamma$  then break else  $\Gamma \leftarrow \Gamma'$   
   $x_\Gamma \leftarrow \text{least squares solution of } A_\Gamma x_\Gamma = b$   
   $r \leftarrow b - Ax$ 
```

nonzeros may be found. Ideally, the strategy used to select the threshold should guarantee that StOMP arrives in the state where all nonzeros of the true signal x_* are selected in the last Γ and the last Γ has no more than m entries. Donoho motivates this criterion with the following lemma. Intuitively, it means the correct support has been identified and that the true signal can be found by computing the minimum norm solution to the least squares sub-problem.

Lemma 1. *Let A be a matrix with its n columns taken randomly from the unit sphere \mathbb{S}^{m-1} . Let x_* denote the result of StOMP and Γ_* its support. Suppose that the support Γ of x with $Ax = b$ is a subset of Γ_* . Suppose in addition that $|\Gamma_*| \leq m$. Then, with high probability, we have perfect recovery: $x = x_*$.*

There are multiple good methods for threshold selection, unfortunately, these are non-trivial statistical methods but were not clearly described in Donoho's paper.² As a concluding remark, it should be noted that the statistical nature of StOMP and the underlying idea that "noiseless underdetermined problems behave like noisy well-determined problems" naturally allows noised measurements in StOMP's theoretical framework.

2.3.2.3 Compressive Sampling Matching Pursuit

StOMP represents an advance over OMP in terms of computational performance. However, threshold selection in StOMP is difficult and no rigorous error bounds are available for StOMP. *Compressive Sampling Matching Pursuit* (CoSaMP), introduced by Needell and Tropp [15], uses a straightforward method to select a suitable support for the solution vector and provides rigorous error bounds. As opposed to OMP and StOMP, the error bounds of CoSaMP are based on the restricted isometry property. In particular this means that CoSaMP succeeds to find the solution to every sample with high probability.

At the most basic level the approach taken by CoSaMP is to select a support larger than necessary, compute a least squares solution and then remove the smallest coordinates. The last step is called *pruning* and reduces the support of the solution

²It is possible that the threshold selection methods might be "common" methods in statistics.

vector. See [Algorithm 3](#) for more details. Observe that selected coordinates can leave the support in later iterations. Also notice that if both x and $x^{(k)}$ are s -sparse, then $x - x^{(k)}$ is at most $2s$ -sparse, which explains the occurrence of H_{2s} in the algorithm.

Algorithm 3: Compressive Sampling Matching Pursuit

Input: Measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m < n$, measurements $b \in \mathbb{C}^m$, sparsity s , maximum number of iterations k_{max} and tolerance $\epsilon \geq 0$.

Output: x such that $x \approx \operatorname{argmin}_{\hat{x}} \|b - A\hat{x}\|_2$ subject to $\|\hat{x}\|_0 \leq s$.

begin

$\Gamma \leftarrow \emptyset, x \leftarrow 0, r \leftarrow b$

for $k \leftarrow 1, 2, \dots, k_{max}$ **and** $\|r\|_2 > \epsilon \|b\|_2$ **do**

$a \leftarrow A^T r$

$\Gamma' \leftarrow \operatorname{supp} x \cup \operatorname{supp} H_{2s}(a)$

if $\Gamma' = \Gamma$ **then break** **else** $\Gamma \leftarrow \Gamma'$

$x_\Gamma \leftarrow H_s(\text{least squares solution of } A_\Gamma x_\Gamma = b)$

$r \leftarrow b - Ax$

Of course, the sparsity of the signal is often unknown in advance. If the sparsity of the signal cannot be approximated reliably, Needell and Tropp suggest to use the estimation $s \approx m/(2 \log n)$ or to run CoSaMP for various values of s , e.g. $s = 1, 2, 4, 8, \dots, m$ while keeping track of the residual norm.

The main results of Needell and Tropp about CoSaMP are summarized in the following theorem.

Theorem 6. *Suppose that A is an $m \times n$ measurement matrix with restricted isometry constant $\delta_{2s} \leq 0.025$. Let $u = Ax + e$ be a vector of measurements of a real or complex signal of length n , contaminated with noise e . for a given precision parameter η , the algorithm CoSaMP produces an s -sparse approximation x_* that satisfies*

$$\|x - x_*\|_2 \leq C \cdot \max \left\{ \eta, \frac{1}{\sqrt{s}} \|x - x_{s/2}\|_1 + \|e\|_2 \right\} \quad (2.9)$$

where $x_{s/2}$ is a best $(s/2)$ -sparse approximation to x . The running time is $\mathcal{O}(\mathcal{L} \cdot \log(\|x\|_2/\eta))$, where \mathcal{L} bounds the cost of a matrix-vector multiply with A or A^* .

The running time results hinge on the assumption that a good approximation to the solution of the least squares sub-problem can be obtained in time \mathcal{L} . For example, iterative methods such as LSQR could be used. This is also the reason they need a very well conditioned matrix with $\delta_{2s} \leq 0.025$.

A corollary of the theorem is that performance of CoSaMP should decrease gracefully as the energy in the noise increases or if the signal is compressible. Because of this and the favorable runtime performance, CoSaMP often shows up as a comparison for new methods.

2.3.2.4 In-Crowd

Instead of solving (2.2) it may be desirable to solve the related ℓ_1 -regularized minimization problem in (QP_λ) instead. In fact, the latter problem is convex and can be solved with quadratic programming (second order cone programming in the complex case). Unfortunately solvers for general quadratic programming problems

are non-trivial to implement and more importantly, become too slow for large and dense measurement matrices. The In-Crowd algorithm [12] by Gill et al. attempts to address the previous problems for a compressed sensing setting. In essence, In-Crowd tries to find the non-zero coordinates of x and solve a reduced version of (QP_λ) .

Algorithm 4: In-Crowd

Input: Measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m < n$, measurements $b \in \mathbb{C}^m$, regularization parameter λ , tuning parameter L , maximum number of iterations $k_{max} \leq \lceil n/L \rceil$ and tolerance $\epsilon \geq 0$.

Output: x such that $x \approx \operatorname{argmin}_{\hat{x}} 1/2 \|b - A\hat{x}\|_2^2 + \lambda \|\hat{x}\|_1$.

begin

$\Gamma \leftarrow \emptyset, x \leftarrow 0, r \leftarrow b$

for $k \leftarrow 1, 2, \dots, k_{max}$ **and** $\|r\|_2 > \epsilon \|b\|_2$ **do**

$a \leftarrow A^T r$

$\Gamma' \leftarrow \operatorname{supp} x \cup \{j \in \operatorname{supp} H_L(a) : |a_j| > \lambda\}$

if $\Gamma' = \Gamma$ **then break** **else** $\Gamma \leftarrow \Gamma'$

$x_\Gamma \leftarrow \operatorname{argmin}_z 1/2 \|b - A_\Gamma z\|_2^2 + \lambda \|z\|_1$

$r \leftarrow b - Ax$

The In-Crowd algorithm closely follows OMP, StOMP and CoSaMP. The support of the solution vector is expanded by taking the L largest components from $A^T r$. A reduced quadratic system corresponding to the active support is created and solved using quadratic programming. And finally, coordinates corresponding to zeros are removed from the active support. See Algorithm 4. Note that if $\epsilon \neq 0$ then the algorithm gives some preference to a sparse solution that satisfies $\|b - Ax\|_2 \leq \epsilon \|b\|_2$ over a solution that exactly solves (QP_λ) . Like the previous algorithms, solving the sub-problem can potentially be expensive. Observe that In-Crowd reduces to OMP when $\lambda = 0$ and $L = 1$.

2.3.3 Convex relaxation algorithms

2.3.3.1 Normalised Iterative Hard Thresholding

Like previous methods, *Iterative Hard Thresholding* (IHT) by Blumensath and Davies [6, 5] is centered around the idea that $A^T b$ provides a “good” approximation x . IHT attempts to solve

$$\operatorname{argmin} \|b - Ax\|_2 \quad \text{subject to} \quad \|x\|_0 \leq s \quad (2.10)$$

by using the iteration

$$x^{(k+1)} = H_s(x^{(k)} + A^T(b - Ax^{(k)})), \quad (2.11)$$

with $x^{(0)} = 0$ (note: we will not provide an algorithm for IHT, but rather for an improved version of IHT for reasons stated below). Although IHT is not derived this way, we can intuitively view the method as a gradient step followed by a selection of the “most important” coefficients. The following theorem summarizes the necessary conditions for convergence.

Theorem 7. *If the columns of A span \mathbb{R}^m and if $\|A\|_2 < 1$, then the Iterative Hard Thresholding algorithm converges to a local minimum of the cost function $\|b - Ax\|_2$ under the constraint that x is s -sparse.*

Unfortunately, the IHT algorithm is sensitive to a re-scaling of A and the algorithm becomes unstable if the conditions in the theorem are not met. This is where *Normalized Iterative Hard Thresholding* (NIHT) comes into play. NIHT was developed by the same authors as IHT and described in [4]. The idea behind NIHT is to choose a special step-size μ in each iteration and compute

$$x^{(k+1)} = H_s(x^{(k)} + \mu A^T(b - Ax^{(k)})). \quad (2.12)$$

The step-size μ must be chosen in such a way that the algorithm becomes scale independent. Let $g = A^T(b - Ax)$ be the negative gradient of $1/2 \|b - Ax\|_2$. Suppose $\Gamma = \text{supp } x_s$ is the support of the best s term approximation to x . In this case the sparse solution can be found by minimizing $1/2 \|b - A_\Gamma x_\Gamma\|_2^2$. Using a gradient descent method the following iteration is obtained

$$x_\Gamma^{(k+1)} = x_\Gamma^{(k)} + \mu A_\Gamma^T(b - A_\Gamma x_\Gamma^{(k)}), \quad (2.13)$$

where it can be shown that the optimal stepsize μ is given by

$$\mu = \|g_\Gamma^{(k)}\|_2^2 / \|A_\Gamma g_\Gamma^{(k)}\|_2^2, \quad (2.14)$$

with $g^{(k)} \equiv A^T(b - Ax^{(k)})$. Back to NIHT. If $\text{supp } x^{(k+1)} = \text{supp } x^{(k)}$ then the choice of μ above is optimal. However, if $\text{supp } x^{(k+1)} \neq \text{supp } x^{(k)}$ then μ needs no longer be optimal. Blumensath and Davies show that in this case, a sufficient condition to guarantee convergence is that $\mu \leq \omega$, with

$$\omega = (1 - c) \frac{\|\tilde{x}^{(k+1)} - x^{(k)}\|_2^2}{\|A(\tilde{x}^{(k+1)} - x^{(k)})\|_2^2} \quad (2.15)$$

for a small fixed constant c . In the formula above $\tilde{x}^{(k+1)}$ is a ‘proposed candidate’, computed using the NIHT iteration and the previous μ . In case $\mu > \omega$ a line search is proposed. Specifically, use the update $\mu \leftarrow \mu / (\kappa(1 - c))$, for some $\kappa > 1/(1 - c)$ until $\mu < \omega$. A large $\kappa(1 - c)$ makes the line search fast while a small $\kappa(1 - c)$ makes the line search accurate. Put everything together and Algorithm 5 is obtained.

The following two interesting theorems conclude the discussion of this section.

Theorem 8. *If $\text{rank}(A) = m$ and $\text{rank}(A_\Gamma) = s$ for all Γ such that $|\Gamma| = s$, then the normalized Iterative Hard Thresholding algorithm converges to a local minimum of the optimization problem (2.10).*

Theorem 9. *Assume A satisfies the non-symmetric restricted isometry property*

$$0 < \alpha_{2s} \leq \frac{\|Ax\|_2}{\|x\|_2} \leq \beta_{2s}$$

for all x with $\|x\|_0 \leq 2s$. Given a noisy observation $b = Ax + e$, where x is an arbitrary vector; let x_s be the best s -term approximation to x , that is $x_s = H_s(x)$.

If the normalized Iterative Hard Thresholding algorithm had $\Gamma' = \Gamma$ in each iteration, then let $\gamma_{2s} = \beta_{2s}^2 / \alpha_{2s}^2 - 1$ else let $\gamma_{2s} = \max\{1 - \alpha_{2s}^2 / (\kappa \beta_{2s}^2), \beta_{2s}^2 / \alpha_{2s}^2 - 1\}$. If $\gamma_{2s} < 1/8$, then, at iteration k , the approximation $x^{(k)}$ satisfies

$$\|x - x^{(k)}\|_2 \leq 2^{-k} \|x_s\|_2 + 8\tilde{\epsilon}_s,$$

Algorithm 5: Normalized Iterative Hard Thresholding

Input: Measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m < n$, measurements $b \in \mathbb{C}^m$, sparsity s , maximum number of iterations k_{max} , tolerance $\epsilon \geq 0$ and tweaking parameters κ and c .

Output: x such that $x \approx \operatorname{argmin}_{\hat{x}} \|b - A\hat{x}\|_2$ subject to $\|\hat{x}\|_0 \leq s$.

begin

```

 $x \leftarrow 0, r \leftarrow b, \Gamma \leftarrow H_s(A^T b)$ 
for  $k \leftarrow 1, 2, \dots, k_{max}$  and  $\|r\|_2 > \epsilon \|b\|_2$  do
     $g \leftarrow A^T r$ 
     $\mu \leftarrow \|g\|_2^2 / \|A_\Gamma g_\Gamma\|_2^2$ 
     $x' \leftarrow H_s(x + \mu g)$ 
     $\Gamma' \leftarrow \operatorname{supp} x'$ 
    if  $\Gamma' \neq \Gamma$  then
        Recompute  $x' \leftarrow H_s(x + \mu g)$  with decreasing  $\mu \leftarrow \mu / (\kappa(1 - c))$ 
        until  $\mu \leq (1 - c) \|x' - x\|_2^2 / \|A(x' - x)\|_2^2$ .
     $x \leftarrow x', \Gamma \leftarrow \operatorname{supp} x$ 
 $r \leftarrow b - Ax$ 

```

where

$$\tilde{\epsilon}_s = \|x - x_s\|_2 + \frac{\|x - x_s\|_1}{\sqrt{s}} + \frac{1}{\beta_{2s}} \|e\|_2.$$

Furthermore, after at most $k^* = \lceil \log_2(\|x^s\|_2 / \tilde{\epsilon}_s) \rceil$ iterations, IHT_s estimates x with accuracy given by

$$\|x - x^{(k^*)}\|_2 \leq 9\tilde{\epsilon}_s.$$

More generally, if $\gamma_{2s} < 1/4$, then there exists a finite constant c , such that

$$\|x - x^{(k^*)}\|_2 \leq c\tilde{\epsilon}_s,$$

with $c \rightarrow \infty$ as $\gamma_{2s} \rightarrow 1/4$.

2.3.3.2 Fixed-Point Continuation

The shrinkage operator plays a central role in this section and the next. The focus of this section is a method known as *Fixed Point Continuation* (FPC) by Hale et al. [13]. Consider a more general minimization problem

$$\operatorname{argmin}_{x \in \mathbb{R}^n} \{\|x\|_1 + \mu f(x)\}, \quad (2.16)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable and convex, but not necessarily strictly convex and $\mu > 0$. For a scalar $\tau > 0$ define $h_\tau : \mathbb{R}^n \mapsto \mathbb{R}^n$ as

$$h_\tau(\cdot) \equiv I(\cdot) - \tau \nabla f(\cdot), \quad (2.17)$$

the value τ is a step size that will be tuned below. In a compressed sensing setting where $f(x) \equiv 1/2 \|b - Ax\|_2^2$ the gradient step h_τ enforces *data consistency*, while the shrinkage operator s_λ enforces sparsity. Thus it makes sense to try the iteration

$$x^{(k+1)} = s_\lambda \circ h_\tau(x^{(k)}) \quad (2.18)$$

It is shown in [13] that x_* is a solution of (2.16) if and only if x_* is a fixed point of $s_\lambda \circ h$ with $\lambda = \tau/\mu$. This further motivates the iteration above. The map $s_\lambda \circ h$ is not a contraction, but it is non-expansive ($\|s_\lambda(h_\tau(x)) - s_\lambda(h_\tau(y))\|_2 \leq \|x - y\|_2$) for $\tau \in (0, 2/\lambda_{\max}(H))$, where H is the Hessian of f and $\lambda_{\max}(H)$ is the largest eigenvalue of H . And in fact, the following theorem holds.

Theorem 10. *Let f be a convex quadratic function that is bounded below, H be its Hessian, and τ satisfy $0 < \tau < 2/\lambda_{\max}(H)$. Then the sequence $\{x^{(k)}\}$, generated by the fixed point iterations (2.18) from any starting point x^0 converges to an optimal solution of (2.16).*

Algorithm 6: Fixed-Point Continuation

Input: Measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m < n$, measurements $b \in \mathbb{C}^m$, number values L for μ , maximum number of iterations k_{\max} for each μ , regularization parameter $\bar{\mu}$, tuning parameters γ and β , tolerance ϵ_1 and complementary tolerance ϵ_2 .

Output: x such that $x \approx \operatorname{argmin}_{\hat{x}} \bar{\mu}/2, \|b - A\hat{x}\|_2^2 + \|\hat{x}\|_1$.

begin

$\tau \leftarrow \min\{1 + 1665(1 - m/n), 1.999\}/\lambda_{\max}(A^T A)$

$k \leftarrow 0, x \leftarrow \mathbf{0}, x' \leftarrow \tau A^T b$

$\mu \leftarrow \tau/(\gamma\|x'\|_\infty), \beta \leftarrow (\bar{\mu}/\mu)^{1/L}$

for $i \leftarrow 1, 2, \dots, L$ **do**

$\lambda \leftarrow \tau/\mu$

for at most k_{\max} steps and $\frac{\|x' - x\|_2}{\max\{\|x\|_2, 1\}} \geq \epsilon_1$ and $\|\mu \nabla f(x)\|_\infty - 1 \geq \epsilon_2$ **do**

$x \leftarrow x', x' \leftarrow s_\lambda(h_\tau(x))$

$k \leftarrow k + 1$

$\mu \leftarrow \beta \mu$

See Algorithm 6 for an algorithmic description of FPC. Experimentation by the original authors showed that for fixed τ , τ should be chosen in the interval $[1/\lambda_{\max}(A^T A), 2/\lambda_{\max}(A^T A)]$. Furthermore, τ should be close to $2/\lambda_{\max}(A^T A)$ when approximately $m/n \leq 0.4$ and should decay to $1/\lambda_{\max}(A^T A)$ as m/n increases. It is proposed to take

$$\tau = \min\{1 + 1.665(1 - m/n), 1.999\}/\lambda_{\max}(A^T A). \quad (2.19)$$

Although the choice for τ is somewhat restricted, there is a lot of freedom when choosing μ . To promote sparsity one can start the iterations with a small μ , e.g. large λ , and gradually increase μ up to the $\bar{\mu}$ for which the minimization problem is to be solved. Repeatedly solving for increasing μ is an example of a *continuation strategy*.

The iterations are initiated with $x^{(0)} = \tau A^T b$ and $\mu^{(1)}$ such that $\tau/\mu^{(1)} = \gamma\|x^{(0)}\|_\infty$, for some specified $0 < \gamma < 1$. For this choice $s_{\tau/\mu^{(1)}}$ will set all coordinates $\leq \gamma\|x^{(0)}\|_\infty$ to zero. Subsequent values for $\mu^{(i)}$ with $i = 2, 3, \dots, L$, are chosen as $\mu^{(i)} = \min\{\mu_1 \beta^{i-1}, \bar{\mu}\}$, where $\beta > 1$ and L is the first integer i for which $\mu_i = \bar{\mu}$. Hale et al. found the pairings $\gamma = 0.99$ with $\beta = 4$ and $\gamma = 0.9$ with $\beta = 2$ to be effective pairings and suggest to choose γ and β such to keep L approximately constant.

The iteration corresponding to a single μ_i is terminated when

$$\frac{\|x^{(k+1)} - x^{(k)}\|_2}{\max\{\|x^{(k)}\|_2, 1\}} < \epsilon_1 \quad \text{and} \quad \mu_i \|g(x^{(k)})\|_\infty - 1 < \epsilon_2. \quad (2.20)$$

The first check requires no explanation. In case $\nabla f(x) \equiv A^T(Ax - b)$ the second check can be rewritten as $\|A^T r^{(k)}\|_\infty < (1 + \epsilon_2)/\mu$ and understood as a measure of the part of the signal that has not yet been approximated. Hale et al. remark that ϵ_2 is allowed to be fairly large, e.g. $\epsilon_2 = 0.2$.

Finally, remark that $\lambda_{\max}(A^T A) = 1$ if A is a partial orthonormal matrix. In other cases it might be possible to approximate the largest eigenvalue using the power method. If computation of $\lambda_{\max}(A^T A)$ is a problem: the algorithms in the next section provide an approach similar to FPC without an absolute need of any eigenvalue.

2.3.3.3 Fast Iterative Shrinkage-Thresholding Algorithm

In this section consider the minimization problem

$$\operatorname{argmin}_{x \in \mathbb{R}^n} \{F(x) \equiv f(x) + g(x)\} \quad (2.21)$$

where $g \in C^0(\mathbb{R}^n, \mathbb{R})$ is a convex function and $f \in C^{1,1}(\mathbb{R}^n, \mathbb{R})$ is a convex function with $L(f)$ -Lipschitz continuous gradient:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L(f)\|x - y\|_2. \quad (2.22)$$

For example, in case of (BP_g) we would have $f(x) \equiv 1/2 \|b - Ax\|_2^2$ and $g(x) \equiv \lambda \|x\|_1$. Assume there exists at least one solution x_* with $F_* \equiv F(x_*)$.

In [1] Beck and Teboulle develop the *Fast Iterative Shrinkage-Thresholding Algorithm* (FISTA) as an improved version of *Iterative Shrinkage-Thresholding Algorithm* (ISTA). The latter is actually a class of algorithms, but one instance in particular relates to FISTA and it is this one that we shall call ISTA. Both ISTA and FISTA are based on a fixed point iteration, just like FPC.

Define $Q_L(x, y)$ as a quadratic approximation of $F(x)$ around y

$$Q_L(x, y) \equiv f(y) + \langle x - y, \nabla f(y) \rangle + \frac{L}{2} \|x - y\|_2^2 + g(x) \quad (2.23)$$

and its minimizer

$$p_L(y) \equiv \operatorname{argmin}_x \{Q_L(x, y)\} \quad (2.24)$$

$$= \operatorname{argmin}_x \left\{ \langle x - y, \nabla f(y) \rangle + \frac{L}{2} \|x - y\|_2^2 + g(x) \right\} \quad (2.25)$$

$$= \operatorname{argmin}_x \left\{ g(x) + \frac{L}{2} \left(\left\| \frac{1}{L} \nabla f(y) \right\|_2^2 + 2 \langle x - y, \frac{1}{L} \nabla f(y) \rangle + \|x - y\|_2^2 \right) \right\} \quad (2.26)$$

$$= \operatorname{argmin}_x \left\{ g(x) + \frac{L}{2} \left\| x - \left(y - \frac{1}{L} \nabla f(y) \right) \right\|_2^2 \right\} \quad (2.27)$$

$$= \operatorname{prox}_L(g) \left(y - \frac{1}{L} \nabla f(y) \right). \quad (2.28)$$

If indeed $g(x) \equiv \lambda \|x\|_1$ then the terminology of FPC applies and $p_L(y) = s_\nu \circ h_\tau$ with $\tau = 1/L$, $\nu = \tau/\bar{\mu}$ and $\bar{\mu} = 1/\lambda$, see [Section 2.3.3.2](#). The choices for μ and τ are consistent with the use of the same parameters in FPC. And indeed, ISTA uses the same iteration as FPC:

$$x^{(k+1)} = s_\nu \circ h(x^{(k)}) = p_L(x^{(k)}). \quad (2.29)$$

An integral part of the iteration above is the shrinkage operator to which ISTA thanks its name.

Beck and Teboulle motivate and prove their results for general f and g as defined at the beginning of the section. However, an alternative argument can be used when $f(x) \equiv 1/2 \|b - Ax\|_2^2$ and $g(x) \equiv \lambda \|x\|_1$. Using the fact that f is a quadratic function and using a multidimensional Taylor expansion³ we see that

$$f(x) = f(y) + (x - y)^T Df(y) + \frac{1}{2!} (x - y)^T \{D^2 f(y)\} (x - y) \quad (2.30)$$

$$= f(y) + \langle x - y, \nabla f(y) \rangle + \frac{1}{2} (x - y)^T A^T A (x - y) \quad (2.31)$$

$$\leq f(y) + \langle x - y, \nabla f(y) \rangle + \frac{\lambda_{\max}(A^T A)}{2} \|x - y\|_2^2. \quad (2.32)$$

Let $L \equiv \lambda_{\max}(A^T A)$ then $F(x) \leq Q_L(x, y)$ for all $x, y \in \mathbb{R}^n$. Furthermore, $p_L(y)$ is the minimizer of a convex function and a strictly convex function, hence $p_L(y)$ is well defined. As a result, if $x^{(k+1)} = p_L(x^{(k)}) \neq x^{(k)}$ then

$$F(x^{(k+1)}) \leq Q(p_L(x^{(k)}), x^{(k)}) < Q(x^{(k)}, x^{(k)}) = F(x^{(k)}). \quad (2.33)$$

On the other hand if $x^{(k+1)} = x^{(k)}$ then we know from FPC that $x^{(k)}$ solves (2.21).

Algorithm 7: Iterative Shrinkage-Thresholding Algorithm (with backtracking)

Input: $L > 0$, growth rate $\eta > 1$ and initial guess $x \in \mathbb{R}^n$.

Output: x such that $F(x) \approx F_*$.

begin

for $k = 1, 2, \dots, k_{\max}$ **do**

$i \leftarrow \operatorname{argmin}_{j \in \mathbb{N}} \{F(p_{\bar{L}}(x)) \leq Q_{\bar{L}}(p_{\bar{L}}(x), x)\}$, where $\bar{L} \equiv \eta^j L$

$L \leftarrow \eta^i L$

$x \leftarrow p_L(x)$

[Algorithm 7](#) is an implementation of ISTA that uses backtracking in case an upperbound for $L(f)$ is not easily computable. The following theorem shows that ISTA can obtain an ϵ -optimal solution, that is, an \tilde{x} such that $F(\tilde{x}) - F_* \leq \epsilon$, in $\mathcal{O}(1/\epsilon)$ iterations. A proof of this theorem and the next one can be found in Beck and Teboulle [1].

Theorem 11. *Let $\{x^{(k)}\}$ be the sequence generated by ISTA. Then for any $k \geq 1$*

$$F(x^{(k)}) - F(x_*) \leq \frac{\alpha L(f) \|x_0 - x_*\|_2^2}{2k},$$

³The current f is a quadratic function and it might not make sense to approximate it by another quadratic function. But remember that Beck and Teboulle motivate and prove their results for $f \in C^{1,1}(\mathbb{R}^n, \mathbb{R})$.

for every solution x_* of (2.21) and where $\alpha = 1$ for the constant stepsize setting and $\alpha = \eta$ for the backtracking stepsize setting.

Algorithm 8: Fast Iterative Shrinkage-Thresholding Algorithm (without backtracking)

Input: $L \geq L(f)$ and initial guess $x \in \mathbb{R}^n$.

Output: x such that $F(x) \approx F_*$.

begin

```

     $y \leftarrow x, t \leftarrow 1$ 
    for  $k = 1, 2, \dots, k_{max}$  do
         $x' \leftarrow p_L(y)$ 
         $t' \leftarrow (1 + \sqrt{1 + 4t^2}) / 2$ 
         $y \leftarrow x' + (t - 1) / t' (x' - x)$ 
         $x \leftarrow x', t \leftarrow t'$ 

```

FISTA improves on ISTA by applying p_L to a special linear combination of the previous two points $x^{(k-1)}$ and $x^{(k-2)}$. See Algorithm 8. Although the changes between ISTA and FISTA are a straightforward, the worst-case convergence rate improves from $\mathcal{O}(1/\epsilon)$ to $\mathcal{O}(1/\sqrt{\epsilon})$. See the following theorem.

Theorem 12. Let $\{x^{(k)}\}$ and $\{y^{(k)}\}$ be generated by FISTA. Then for any $k \geq 1$

$$F(x^{(k)}) - F(x_*) \leq \frac{2\alpha L(f) \|x_0 - x_*\|_2^2}{(k+1)^2},$$

for every solution x_* of (2.21) and where $\alpha = 1$ for the constant stepsize setting and $\alpha = \eta$ for the backtracking stepsize setting.

There is an algorithm called *Fast Composite Splitting Algorithm* (FCSA) by Huang et. al., which is more or less FISTA with an additional regularizing norm, the *Total Variation* (TV) norm. The TV norm penalizes differences between adjacent pixels and can be used to denoise images, see Beck and Teboulle 2009b [2]. Experiments show that TV-minimization can be a worthwhile addition to Compressed Sensing (image) reconstruction. Although FCSA is beyond the scope of this section, it is featured in the numerical experiments in Section 2.4.

2.3.3.4 Spectral Projected Gradient for ℓ_1 Minimization

Yet another method that employs a gradient projection is *Spectral projected gradient for ℓ_1 minimization* (SPGL1) by Van den Berg and Friedlander [3]. Their idea is based on the notion that if (BP_σ) and (LS_τ) are satisfied, then $\|b - Ax\|_2 = \sigma$ and $\|x\|_1 = \tau$ respectively. Hence, for some fixed σ there must exist some τ_σ such that the solutions of (BP_σ) and (LS_τ) coincide. E.g., if we define τ_σ as

$$\tau_\sigma \equiv \underset{x}{\operatorname{argmin}} \|x\|_1 \quad \text{subject to} \quad \|b - Ax\|_2 = \sigma \quad (2.34)$$

then

$$\sigma = \underset{x}{\operatorname{argmin}} \|b - Ax\|_2 \quad \text{subject to} \quad \|x\|_1 = \tau_\sigma \quad (2.35)$$

for the same x . Efficient methods to solve the Lasso problem (LS_τ) can be used to solve the Basis Pursuit Denoising problem (BP_σ) if τ_σ can be found.

Let x_τ be the optimal solution of (LS_τ) , define $r_\tau \equiv b - Ax_\tau$ and $\phi(\tau) \equiv \|r_\tau\|_2$. The function ϕ parameterizes the so called *Pareto curve*: the curve that defines the optimal trade-off between $\|b - Ax\|_2$ and $\|x\|_1$. Van den Berg and Friedlander prove the following interesting properties of ϕ .

Theorem 13. (a) The function ϕ is convex and nonincreasing.

(b) Let τ_{BP} be the optimal objective value of the problem (BP) . For all $\tau \in (0, \tau_{BP})$, ϕ is continuously differentiable, $\phi'(\tau) = -\lambda_\tau$ and the optimal dual variable $\lambda_\tau = \|A^T y_\tau\|_\infty$, where $y_\tau = r_\tau / \|r_\tau\|_2$.

(c) For $\tau \in [0, \tau_{BP}]$, $\|x_\tau\|_1 = \tau$ and ϕ is strictly decreasing.

The theorem implies that the solution τ_σ of $\phi(\tau) = \sigma$ can be obtained with a straightforward Newton-Raphson iteration, e.g.,

$$\tau^{(k+1)} = \tau^{(k)} + (\sigma - \phi(\tau^{(k)})) / \phi'(\tau^{(k)}). \quad (2.36)$$

To evaluate $\phi(\tau^{(k)})$ it is necessary to solve the Lasso problem in order to obtain $r_{\tau^{(k)}}$. The theorem above also implies that $\phi'(\tau^{(k)})$ can be obtained cheaply from $r_{\tau^{(k)}}$. Of course, solving the Lasso problem itself is rather expensive and to deal with this problem a rough approximation to $x_{\tau^{(k)}}$ is used and the Newton-Raphson iteration is carried out with approximate values. The accuracy of the approximations is increased as k increases.

Algorithm 9: Spectral Projected Gradient for (LS_τ)

Input: Initial iterate x , maximum ℓ_1 -norm τ and tolerance ϵ_2 .

Output: An approximate solution x_τ of (LS_τ) and corresponding residual r_τ .

Set minimum and maximum step lengths $0 < \alpha_{min} < \alpha_{max}$.

Set initial step length $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$ and sufficient descent parameter $\gamma \in (0, 1)$.

Set an integer line search history length $M \geq 1$.

begin

$x^{(0)} \leftarrow P_\tau[x]$, $r^{(0)} \leftarrow b - Ax^{(0)}$, $g^{(0)} \leftarrow -A^T r^{(0)}$

$\epsilon \leftarrow \|r^{(0)}\|_2 - (b^T r^{(0)} - \tau \|g^{(0)}\|_\infty) / \|r^{(0)}\|_2$

for $k = 1, 2, \dots, k_{max}$ **and** $\epsilon > \epsilon_2$ **do**

$\bar{x} \leftarrow P_\tau[x^{(k)} - \alpha g^{(k)}]$

$\bar{r} \leftarrow b - A\bar{x}$

if $\|\bar{r}\|_2^2 > \max_{j \in [0, \min\{k, M-1\}]} \|r^{(k-j)}\|_2^2 + \gamma(\bar{x} - x^{(k)})^T g^{(k)}$ **then**

 | recompute \bar{x} and \bar{r} with $\alpha \leftarrow \alpha/2$

$x^{(k+1)} \leftarrow \bar{x}$, $r^{(k+1)} \leftarrow \bar{r}$, $g^{(k+1)} \leftarrow -A^T r^{(k+1)}$

$\epsilon \leftarrow \|r^{(k+1)}\|_2 - (b^T r^{(k+1)} - \tau \|g^{(k+1)}\|_\infty) / \|r^{(k+1)}\|_2$

$\epsilon_2 x \leftarrow x^{(k+1)} - x^{(k)}$, $\epsilon_2 g \leftarrow g^{(k+1)} - g^{(k)}$

if $\epsilon_2 x^T \epsilon_2 g \leq 0$ **then**

 | $\alpha \leftarrow \alpha_{max}$

else

 | $\alpha \leftarrow \min\{\alpha_{max}, \max[\alpha_{min}, (\epsilon_2 x^T \epsilon_2 x) / (\epsilon_2 x^T \epsilon_2 g)]\}$

$x_\tau \leftarrow x^{(k)}$, $r_\tau \leftarrow r^{(k)}$

The Lasso problem is solved using a *Spectral Projected Gradient* (SPG) method, see [Algorithm 9](#). The algorithm appears to be quite complex at first sight, however, actually consists of only a few simple steps. First some initial values are set, then, at the beginning of each iteration, a gradient step is projected onto the set $\{x : \|x\|_1 \leq \tau\}$ using P_τ . The projection is followed by a line search to make sure that the objective function decreases sufficiently at least every M iterations. Finally, the iterates are updated and a ‘good’ step size for the next iteration is computed. The details of the latter computation are beyond the details of this discussion.

The one-norm projection P_τ defined as

$$P_\tau[c] \equiv \left\{ \underset{x}{\operatorname{argmin}} \|c - x\|_2 \text{ subject to } \|x\|_1 \leq \tau \right\} \quad (2.37)$$

forces the next iterate to satisfy the $\|x\|_1 \leq \tau$ constraint of Lasso. In light of the previous algorithms we discussed, it is interesting to observe that from some unknown value $\epsilon_{2,\tau}$ the above projection reduces to

$$\underset{x}{\operatorname{argmin}} \left\{ \frac{1}{2} \|x - c\|_2^2 + \epsilon_{2,\tau} \|x\|_1 \right\}, \quad (2.38)$$

which we recognize as the shrinkage-operator. Details for the computation of $\epsilon_{2,\tau}$ are given in [\[3\]](#) and are omitted here since they do not provide extra insight into SPGL1.

Finally, [Algorithm 10](#) shows pseudo code for SPGL1 that uses SPG to solve (LS_τ) .

Algorithm 10: Spectral Projected Gradient for ℓ_1 Minimization

Input: Measurement matrix $A \in \mathbb{C}^{m \times n}$ with $m < n$, measurements $b \in \mathbb{C}^m$, maximum number of iterations k_{max} , tolerance $\epsilon_1 \geq 0$ and noise bound $\sigma \geq 0$.

Output: x such that $x \approx \underset{\hat{x}}{\operatorname{argmin}} \|\hat{x}\|_1$ subject to $\|b - A\hat{x}\|_2 \leq \sigma$.

begin

```

 $x \leftarrow 0, \tau \leftarrow 0, \epsilon \leftarrow \infty$ 
for  $k = 1, \dots, k_{max}$  and  $\epsilon > \epsilon_1$  do
     $\epsilon_2 \leftarrow$  suitable tolerance value that decreases as  $k$  and  $\tau$  increase
     $x, r \leftarrow SPG[x, \tau, \epsilon_2]$ 
     $y \leftarrow r / \|r\|_2$ 
     $\lambda \leftarrow \|A^T y\|_\infty$ 
     $\tau' = \tau - (\sigma - \|r\|_2) / \lambda$ 
     $\epsilon \leftarrow \tau' - \tau$ 
     $\tau \leftarrow \tau'$ 

```

2.3.3.5 Nonlinear Conjugate Gradients

As shown by Lustig et al. [\[14\]](#) *Nonlinear Conjugate Gradients* (NLCG) can be used to solve (QP_λ) . The general idea is to minimize the cost function

$$f(x) = 1/2 \|b - Ax\|_2^2 + \lambda \|x\|_1, \quad (2.39)$$

using line searches in (conjugate) gradient directions. However, the gradient ∇f is required while the ℓ^1 -norm is not a smooth function. This problem is ‘solved’ by

introducing a smoothing parameter $\mu > 0$ such that $|x_i|$ can be approximated by $|x_i| \approx \sqrt{x_i^2 + \mu}$ and

$$\frac{d|x_i|}{dx_i} \approx \frac{x_i}{\sqrt{x_i^2 + \mu}}. \quad (2.40)$$

Define f_μ as f where $\|\cdot\|_1$ is replaced by its smooth approximation with smoothing parameter μ . Then f_μ defines a family of functions suitable for use with NLCG.

Algorithm 11: Nonlinear Conjugate Gradients

Input: Arbitrary cost function $f \in C^1(\mathbb{C}^n, \mathbb{R})$, maximum number of iterations k_{max} and tolerance $\epsilon \geq 0$.

Output: x such that $x \approx \operatorname{argmin}_{\hat{x}} f(\hat{x})$.

begin

```

 $x \leftarrow 0, r \leftarrow -\nabla f(x), d \leftarrow r$ 
for  $k \leftarrow 1, 2, \dots, k_{max}$  and  $\|r\|_2 > \epsilon \|b\|_2$  do
     $\alpha \leftarrow \operatorname{argmin}_\alpha f(x + \alpha d)$ 
     $x \leftarrow x + \alpha d$ 
     $r' \leftarrow -\nabla f(x)$ 
     $\beta \leftarrow r'^T r' / r^T r$ 
     $r \leftarrow r'$ 
     $d \leftarrow r + \beta d$ 

```

Algorithm 11 shows the pseudo-code for NLCG. There are multiple ways to compute α , such as a Newton-Raphson iteration, the Secant method and backtracking. One can also choose β from multiple values. For an extensive treatment of (Nonlinear) Conjugate Gradients the various ways to compute α and β , see e.g. Shewchuk 1994 [16]. Lustig uses a straightforward backtracking line search (reducing the step size until an improvement is obtained) and the Fletcher-Reeves formula respectively. The implementation used for the numerical tests later on uses the Polak-Ribière formula and a two step method to find α . First, the Secant method is used to approximate α , which is then followed by Lustig’s line search. This two step method provides faster, but still smooth, convergence.

2.4 Numerical experiments

This section aims to compare the performance of the algorithms in three different tests. The first two tests concern the probability of signal recovery and are fairly synthetic in nature. The third test is a possible real world application and compares accuracy and runtime performance. With the exception of SPGL1 all algorithms use a custom MATLAB implementation.

2.4.1 Recovery from noise-free and noised measurements

In the first test an s -sparse signal x of length $n = 800$ is “measured” using an $m \times n$ matrix A . Elements of x and A are randomly sampled from the standard normal distribution. The experiment is repeated for values $\delta \equiv m/n$ and $\rho \equiv s/m$ from 0.05 to 0.95 divided in 50 equally spaced points. Furthermore, the results are averaged

over ten⁴ runs and every run uses a new A and x . The second experiment is the same but the measurements are contaminated with white noise. The energy of the noise is $1/100$ times the energy $\|Ax\|_2$ of the measurements.

The results are shown in so called phase diagrams. The color in each point of the diagram represents the number of coordinates from the reconstructed x that differ from more than 10^{-4} from their original value. As can be seen from [Figure 2.1](#), some diagrams show a sharp transition from perfect agreement to perfect disagreement, e.g., a “phase transition”.

Relevant details for each algorithm are described below.

- *OMP*. The least squares subproblems were solved exactly because MATLAB’s direct solver was as fast, or even faster, than LSQR for $n = 800$. The maximum number of (outer) iterations was set to $\lceil 3/2s \rceil$.
- *StOMP*. No results were computed for StOMP; proper threshold selection proved to be critical, yet (too) marginally documented. Although SparseLab⁵ includes an implementation of StOMP, there were some difficulties to “get it going”.
- *CoSaMP*. Just like OMP the least squares subproblem in CoSaMP is solved exactly. The sparsity s is set to the exact sparsity of x . Needell And Tropp reported that 10 iterations for CoSaMP were sufficient for all their experiments [15], therefore $k_{max} = 10$ was used.
- *In-Crowd*. The quadratic programming subproblems were solved with QPC⁶. The tweaking parameter and maximum number of iterations were set to $L = 5$ and $\lceil 3/2s/L \rceil$ respectively.
- *NIHT*. The sparsity s was set to the exact sparsity of x . The number of line search iterations was not bounded, the maximum number of outer iterations was (poorly) guesstimated to match FPC’s and FISTA’s running time and set to 200.
- *FPC*. For FPC $f(x) = 1/2 \|Ax - b\|_2^2$ and $\bar{\mu} = 1/\lambda$. μ takes $L = 20$ values and for each μ at most 250 iterations are used. These choices make sure the maximum number of inner iterations are equal for FPC and FISTA.
- *FISTA*. FISTA uses the same f as FPC and in addition $g(x) = \lambda \|x\|_1$. The number of iterations was limited to 5000, well above 5 iterations for each nonzero in every test.
- *SPGL1*. The noise sensitivity parameter is set to $\sigma = 0$ in the noise-free case and set to $0.01 \|b\|_2$ in the noised case. The original authors’ implementation was used.⁷
- *NLCG*. For NLCG $f(x)$ is the smooth approximation to $1/2 \|Ax - b\|_2^2 + \lambda \|x\|_1$, with smoothing parameter $\mu = 10^{-8}$. The number of outer iterations is set to 20, the number of secant steps to 10 (often only 2 or 3 are sufficient) and the number of backtracking steps is unlimited. The running time should match FPC and FISTA.

⁴All tests together require significant computation time; ten runs fit within time constraints.

⁵See <http://sparselab.stanford.edu/>.

⁶Quadratic Programmin in C by Adrian Wills, see <http://sigpromu.org/quadprog/>.

⁷See <http://www.cs.ubc.ca/labs/scl/spg11/>.

In each case the tolerance was set to 10^{-5} and $\lambda = 0.01\|A^*b\|_\infty$ if applicable. Choosing a good regularization parameter λ is difficult and can significantly influence the solution. The current choice is the smallest value for which the phase diagram of In-Crowd is still nicely smooth.

The results for the noise-free tests are shown in [Figure 2.1](#). Phase transitions are clearly visible in the diagrams of OMP, CoSaMP, NIHT and SPGL1. Especially CoSaMP has a sharp phase transition, for SPGL1 the transition is not as sharp. OMP clearly offers the best chances at recovery and NIHT performs marginally better than CoSaMP. For large δ SPGL1 is competitive with CoSaMP and NIHT, but not for small δ . In-Crowd's phase diagram is very smooth, but there is no phase transition. This suggests that either (QP_λ) is unsuitable for truly sparse signals or that λ was poorly chosen. Of course, for $L = 1$ and $\lambda = 0$ In-Crowd should reduce to OMP. The results for FPC and FISTA seem similar to In-Crowd's but not quite as smooth. The performance of NLCG is disappointing.

Better results for FPC can be achieved with a final *denoising* step [13], the same step could be used for In-Crowd and FISTA. The denoising step is straightforward: once the algorithm returns an approximate solution select all coordinates i such that $|x_i| > \epsilon$ for some small ϵ . If the number of selected coordinates is less than or equal to m solve a reduced least squares problem, just like OMP, StOMP and CoSaMP.

Results for the noised case can be seen in [Figure 2.2](#). Solvers for the ℓ_1 -regularized minimization problem (QP_λ) provide virtually unchanged results. NLCG still disappoints. SPGL1 suffers a lot from the noise, which is surprising considering [Theorem 3](#). Phase transitions are still visible for OMP, CoSaMP and NIHT, however OMP does lose its edge for large δ and ρ . From the latter three algorithms CoSaMP appears to be the most robust to noise and still has the sharpest phase transition.

2.4.2 A practical example

The third experiment simulates an MRI scan (although the image is not a typical MR image but instead the famous picture of Lena). The sampling "matrix" A is given by $A \equiv \mathcal{R} \mathcal{F} \mathcal{W}^{-1}$, where \mathcal{W} is an (orthonormal) wavelet decomposition, \mathcal{F} a two dimensional Fourier transform and \mathcal{R} is a restriction to some k -space trajectory. If x is the image, then $\mathcal{W}x$ is the (hopefully) sparse signal that has to be recovered. In this example \mathcal{W} has five levels and the wavelet is Daubechies 4. The k -space trajectory that defines \mathcal{R} is a radial like trajectory (to avoid gridding problems the trajectory consists of points on a Cartesian grid close to a radial line) and includes about 15.1% of the k -space within the field of view, e.g., $b \equiv A\mathcal{W}x$ is more than six and a half times undersampled.

The quality of the reconstructed images is quantified as the *Peak Signal-to-Noise Ratio* (PSNR), the *Mean Squared Error* (MSE), the maximum error between pixels and the ratio between the ℓ_2 -norm of the approximation and the true image. The PSNR is expressed in decibel and, for monochrome images with pixel values in the range $[0, 1]$, defined as

$$PSNR = -10 \log_{10}(MSE). \quad (2.41)$$

The other values follow obvious definitions.

Since we wish to compute the products Ax and A^*x with (inverse) Fast Fourier Transforms and (inverse) Fast Wavelet Transforms, it is undesirable to compute the actual A . Therefore, some extra work needs to be done whenever a selection of columns of A is necessary. This is not a problem for NIHT, nor OMP and CoSaMP if

the least squares subproblem is solved with, e.g., LSQR. The same is not true for QPC, the solver of In-Crowd’s quadratic programming subproblem, it needs a dense matrix. For an image of 512×512 pixels and 10000 non-zero coordinates in x that could mean a selection of A that is about 40 GB in size. Needless to say, In-Crowd was not tested. OMP was not tested either because it was simply too slow.

For CoSaMP and NIHT s was set to 50000, about 19% of the size of the image (actually, for CoSaMP we would want $2s \leq m$, but taking $s = \lceil 512^2 \times 0.15/2 \rceil = 19661$ resulted in a PSNR of 17.35, a massive decrease in quality). When applicable $\lambda = 10^{-3}$ seemed to provide good results. The number of iterations for each tested algorithm can be found in with the results in [Table 2.1](#). Since CoSaMP and NIHT both use fixed sparsity s it seemed natural to match NIHT’s running time to CoSaMP’s instead of the others’.

Table 2.1: Objective reconstruction performance for the “Magnetic Resonance Imaging” test.

Algorithm	PSNR	MSE	Max Err	ℓ_1 -Rat	Iterations	Time (s)
Zero filling	25.5427	0.0028	0.4080	0.9874	–	–
CoSaMP	27.8403	0.0016	0.3332	0.9964	10	1473.61
NIHT	27.4543	0.0018	0.3490	0.9944	700	1511.29
NLCG	28.0191	0.0016	0.3102	0.9924	25	108.64
FPC	29.2593	0.0012	0.3028	0.9958	10×100	107.01
FISTA	29.2923	0.0012	0.3005	0.9962	1000	97.78
SPGL1	29.3162	0.0012	0.3011	0.9964	425	104.26
FCSA	32.4181	0.0006	0.2316	0.9971	100	37.77

The results in the table show that CoSaMP, NIHT and NLCG objectively improve upon zero filling. However, the pictures in [Figure 2.3](#) and [Figure 2.4](#) show little to no visual improvement. FPC, FISTA and SPGL1 do produce visual improvement and objectively achieve nearly identical quality. The convergence histories in [Figure 2.5](#) show that CoSaMP and FISTA converge very fast. FPC and SPGL1 converge in a stepwise manner, unsurprising considering their implementation. NIHT sports slow convergence and the gradient in NLCG fails to converge to zero, indicating that no optimum is reached.

2.4.3 A note about runtime performance

Comparing runtime performance of the algorithms is incredibly difficult. The algorithms differ in their implementation and their actual ability to recover various kinds of signals. Let us start with what we know. [Table 2.2](#) shows the average total running time for all experiments of the phase diagrams. [Table 2.1](#) shows the running time for the MRI experiment and [Figure 2.5](#) the convergence history.

From the tables it is clear that CoSaMP is much faster than OMP. Still, solving the least squares problem is expensive and NIHT proves that there is another way, at least for noise-free signals. FPC and FISTA lack the line search of NIHT and are faster still. NIHT does achieve better recovery for noise-free sparse signals, but FPC and FISTA are more accurate and converge faster in the MRI experiment. The speed of In-Crowd is quite a surprise, one would expect In-Crowd to be about L times as fast as OMP, but In-Crowd is much faster. Perhaps QPC is simply incredibly fast.

Table 2.2: Average total running time (in seconds) for a full set of ‘phase experiments’.

Algorithm	Noise-free	Noised
OMP	9020	10705
CoSaMP	608	667
In-Crowd	61	57
NIHT	308	354
NLCG	189	185
FPC	158	166
FISTA	147	139
SPGL1	2099	485

The case of SPGL1 is interesting, in the phase experiments it is rather slow and there is a large difference between the noise-free and the noised case. Yet, in the MRI experiment it manages to achieve virtually identical accuracy as FISTA in a comparable amount of time.

2.5 Conclusion

There are many different algorithms for compressed sensing reconstruction. All provide very different performance characteristics, both in terms of accuracy and speed. The greedy algorithms perform excellent when the signal is sparse and the number of nonzeros and measurements are both small. While OMP provides slightly better recovery, CoSaMP is much faster and (provably) more robust. Depending on noise and the regularization parameter λ , In-Crowd could be competitive with OMP and CoSaMP.

When there are many measurements and when the signal has many nonzero coordinates, the subproblems of the greedy algorithms may become too expensive to solve. In addition, the need to “select” columns may be an undesirable trait, e.g., if the measurement matrix is a partial Fourier matrix represented by the FFT. In both cases it makes sense to use gradient algorithms such as FPC, FISTA, SPGL1 and NLCG.

Of the gradient algorithms FPC and FISTA provide nearly identical accuracy and speed. FISTA does have an edge over FPC, it has a cleaner implementation, requires fewer tweaking parameters and tunes itself. There does not seem to be a compelling reason to choose NIHT over FPC or FISTA, unless the exact sparsity is known and the measurements are noise-free. NLCG performs poorly compared to the other algorithms, in fact, each test includes algorithms with better recovery at equal or superior runtime performance.

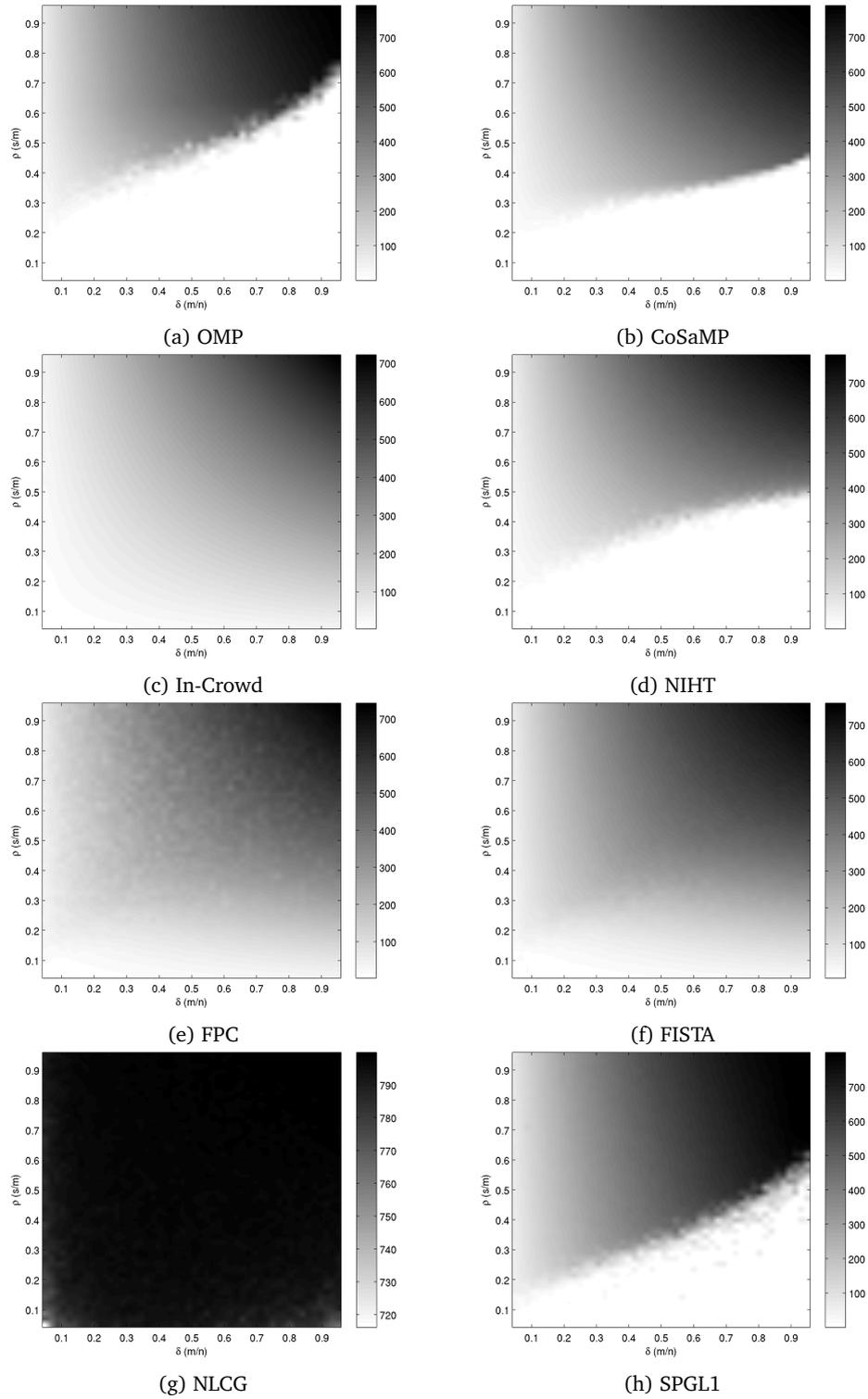


Figure 2.1: Phase diagrams showing the recovery performance of various algorithms for noise-free measurements. See the text the settings used.

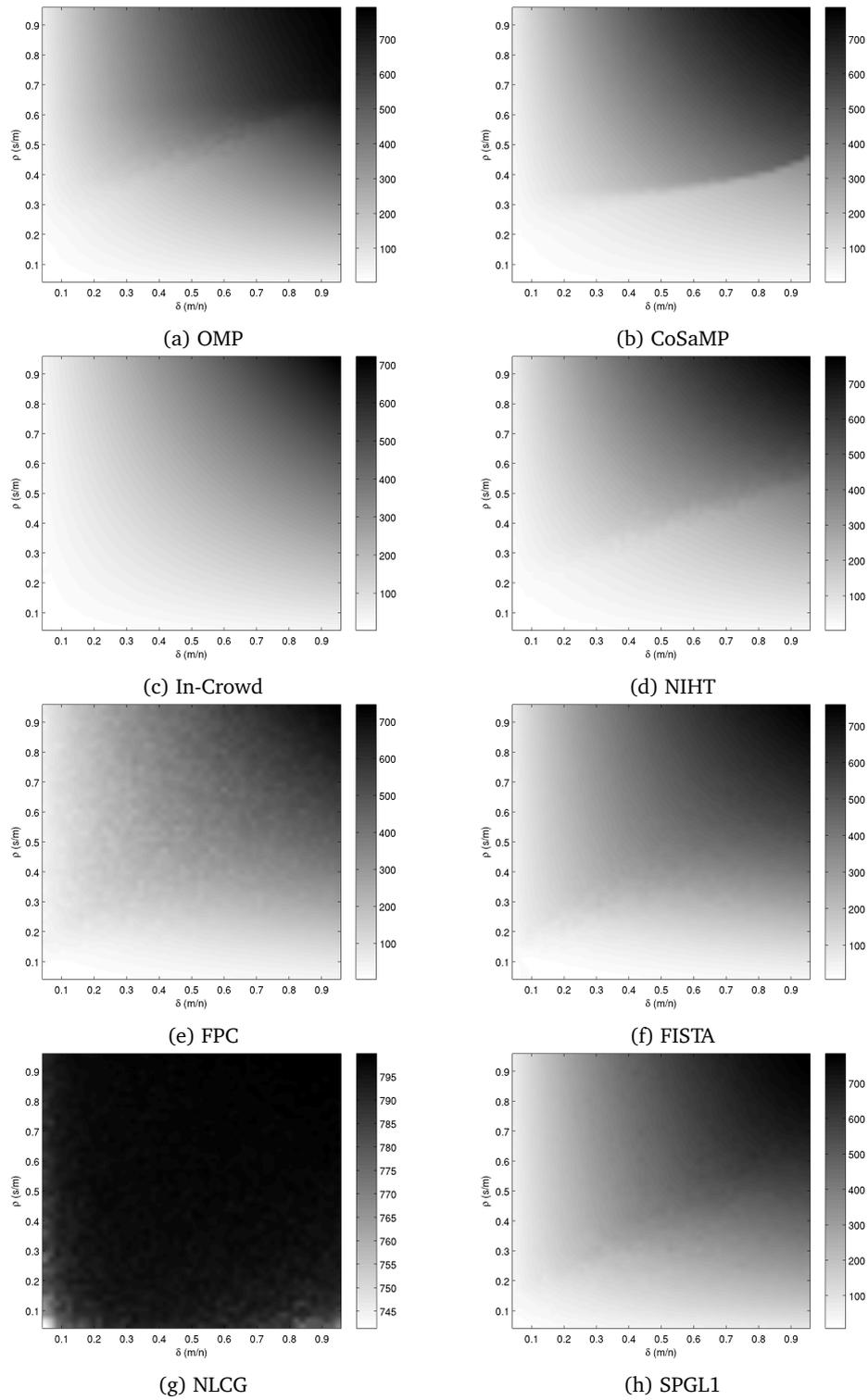
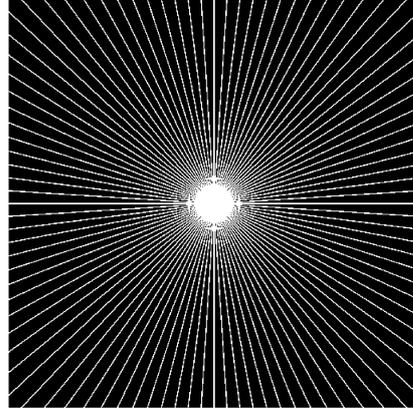


Figure 2.2: Phase diagrams showing the recovery performance of various algorithms for noised measurements. See the text the settings used.



(a) Original



(b) K-Space trajectory



(c) Zero filling



(d) CoSaMP



(e) NIHT

Figure 2.3: Simulated magnetic resonance imaging reconstruction from 15.1 percent approximate radial data.



(a) NLCG



(b) FPC



(c) FISTA



(d) SPGL1



(e) FCSA

Figure 2.4: Simulated magnetic resonance imaging reconstruction from 15.1 percent approximate radial data.

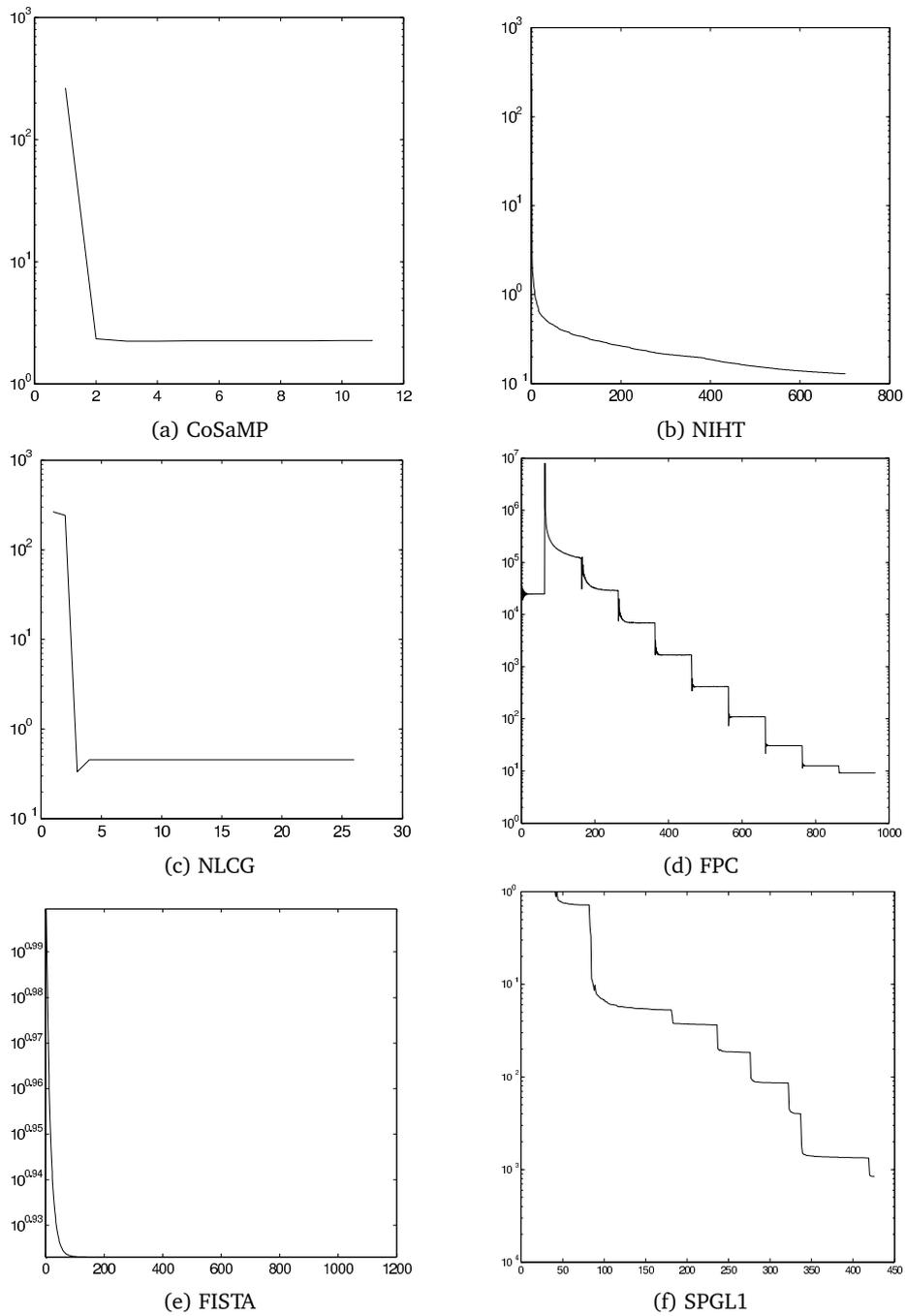


Figure 2.5: Convergence for various algorithms in the ‘Magnetic Resonance Imaging’ test. CoSaMP and NIHT report the residual norm for outer iterations. NLCG reports the gradient norm in outer iterations. FPC and FISTA report (approximate) objective values for inner iterations. The convergence for SPGL1 is the relative error in each iteration as reported by the package.

References

- [1] Amir Beck and Marc Teboulle. “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems”. In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202. DOI: [10.1137/080716542](https://doi.org/10.1137/080716542). URL: <http://link.aip.org/link/?SII/2/183/1> (cit. on pp. 27, 28).
- [2] Amir Beck and Marc Teboulle. “Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems”. In: *Image Processing, IEEE Transactions on* 18.11 (2009), pp. 2419–2434. ISSN: 1057-7149. DOI: [10.1109/TIP.2009.2028250](https://doi.org/10.1109/TIP.2009.2028250) (cit. on p. 29).
- [3] E. van den Berg and M. P. Friedlander. “Probing the Pareto frontier for basis pursuit solutions”. In: *SIAM Journal on Scientific Computing* 31.2 (2008), pp. 890–912. DOI: [10.1137/080714488](https://doi.org/10.1137/080714488). URL: <http://link.aip.org/link/?SCE/31/890> (cit. on pp. 29, 31).
- [4] T. Blumensath and M.E. Davies. “Normalized Iterative Hard Thresholding: Guaranteed Stability and Performance”. In: *Selected Topics in Signal Processing, IEEE Journal of* 4.2 (Apr. 2010), pp. 298–309. ISSN: 1932-4553. DOI: [10.1109/JSTSP.2010.2042411](https://doi.org/10.1109/JSTSP.2010.2042411) (cit. on p. 24).
- [5] Thomas Blumensath and Mike E. Davies. “Iterative hard thresholding for compressed sensing”. In: *Applied and Computational Harmonic Analysis* 27.3 (2009), pp. 265–274 (cit. on p. 23).
- [6] Thomas Blumensath and Mike E. Davies. “Iterative Thresholding for Sparse Approximations”. In: *Journal of Fourier Analysis and Applications* 14.5 (2008), pp. 629–654 (cit. on p. 23).
- [7] E.J. Candes and Y. Plan. “A Probabilistic and RIPless Theory of Compressed Sensing”. In: *Information Theory, IEEE Transactions on* 57.11 (Nov. 2011), pp. 7235–7254. ISSN: 0018-9448. DOI: [10.1109/TIT.2011.2161794](https://doi.org/10.1109/TIT.2011.2161794) (cit. on p. 17).
- [8] Emmanuel J. Candés. “Compressive sampling”. In: *Proceedings of the International Congress of Mathematicians*. 2006 (cit. on p. 17).
- [9] Emmanuel J. Candés. *The Restricted Isometry Property and Its Implications for Compressed Sensing*. Tech. rep. Applied & Computational Mathematics, California Institute of Technology, 2008 (cit. on pp. 16, 17).
- [10] David L. Donoho et al. *Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit*. Tech. rep. 2006 (cit. on p. 20).
- [11] Massimo Fornasier and Holger Rauhut. *Compressive Sensing*. Tech. rep. 2010 (cit. on pp. 17, 18).
- [12] P.R. Gill, A. Wang, and A. Molnar. “The In-Crowd Algorithm for Fast Basis Pursuit Denoising”. In: *Signal Processing, IEEE Transactions on* 59.10 (Oct. 2011), pp. 4595–4605. ISSN: 1053-587X. DOI: [10.1109/TSP.2011.2161292](https://doi.org/10.1109/TSP.2011.2161292) (cit. on p. 23).
- [13] Elaine T. Hale, Wotao Yin, and Yin Zhang. *A Fixed-Point Continuation Method for ℓ_1 -Regularized Minimization with Applications to Compressed Sensing*. Tech. rep. Department of Computational and Applied Mathematics, Rice University, July 2007 (cit. on pp. 25, 26, 34).

- [14] Michael Lustig, David Donoho, and John M. Pauly. “Sparse MRI: The application of compressed sensing for rapid MR imaging”. In: *Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195. ISSN: 1522-2594. DOI: [10.1002/mrm.21391](https://doi.org/10.1002/mrm.21391). URL: <http://dx.doi.org/10.1002/mrm.21391> (cit. on p. 31).
- [15] D. Needell and J. A. Tropp. *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*. Tech. rep. California Institute of Technology, Pasadena, 2008 (cit. on pp. 18, 21, 33).
- [16] Jonathan R Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. rep. Pittsburgh, PA, USA, 1994 (cit. on p. 32).
- [17] J.A. Tropp and A.C. Gilbert. “Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit”. In: *Information Theory, IEEE Transactions on* 53.12 (Dec. 2007), pp. 4655–4666. ISSN: 0018-9448. DOI: [10.1109/TIT.2007.909108](https://doi.org/10.1109/TIT.2007.909108) (cit. on p. 20).

Chapter 3

Accelerated Radial Compressed Sensing

3.1 Introduction

Dynamic MR-guided intervention is one of the next frontiers in medicine and demands robust high speed imaging. The amount of data that can be scanned in real time is limited by physical (e.g. spin-lattice and spin-spin relaxation times), hardware (e.g. magnets and coils) and regulatory (e.g. maximum specific absorption rate) constraints. Thus, it makes sense to attempt to do *more with less*, e.g., reconstruct image space data from undersampled k -space data.

Undersampling k -space is commonly used in static imaging and clinically proven methods include methods such as SENSE [20] and GRAPPA [7]. Recent research [14, 15] shows that Compressed Sensing (CS) is a viable alternative approach. CS allows, under certain conditions, to exactly recover a signal sampled below its Nyquist rate.

An important requirement for CS accelerated acquisitions is incoherent sampling in order to prevent coherent aliasing artifacts. Popular choices include radial acquisitions. Another benefit is that a few radial profiles suffice to densely sample the low frequencies. In addition, radial acquisitions allow short repetition times and fit well in dynamic applications since each radial profile updates information from all spatial frequencies, allowing for exchanging spatial for temporal resolution using a sliding window approach.

A major downside of non-Cartesian acquisitions in general and radial acquisitions in particular is that the Fast Fourier Transform can no longer be used directly to compute image space data from k -space data, see also Figure 3.1. Instead, *gridding* [10, 25] is required or the *Nonequispaced Fast Fourier Transform (NFFT)* [12] can be used. Both are approximate methods and require additional computational effort. The latter is a problem in traditional CS approaches where the Fourier transform is part of the sensing matrix and must be computed many times by the CS reconstruction algorithm. In fact, the computational effort is such that even highly accelerated GPU solutions require hours to reconstruct the signal [16]. This is unacceptable in dynamic MRI guided interventions which require real-time feedback during the procedure.

We present a method dubbed ARCS, short for *Accelerated Radial Compressed*

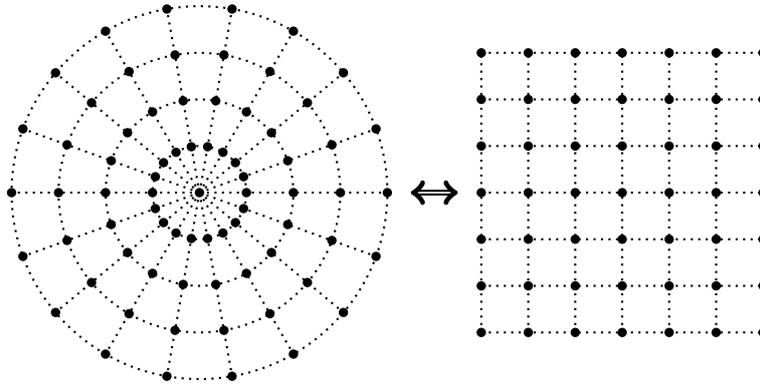


Figure 3.1: Radially acquired k -space points (left) must be transformed to Cartesian image space points (right) and vice versa.

Sensing, which attempts to reconstruct radial k -space data without the need to of gridding or the NFFT, aiming at a reduction of reconstruction time. ARCS can be used for both two and three dimensional radial acquisitions.

3.2 Theory

Current CS accelerated acquisition methods attempt to solve a problem of the form

$$\operatorname{argmin}_x \frac{1}{2} \|\Phi x - b\|_2^2 + \alpha \|x\|_{TV} + \beta \|\Psi^* x\|_1, \quad (3.1)$$

where b is the undersampled k -space signal, x the image space signal, Φ a, possibly nonlinear, partial Fourier Transform and Ψ a wavelet operator. The regularisation parameters α and β are both non-negative and at least one of the two is non-zero. Solving the problem above is the most “obvious” and perhaps the most comfortable way to go. E.g., the acquired k -space data is a sample of the NMR signal of protons in image space and partial Fourier Transforms are known to do well in CS experiments. And in addition, we are comfortable to assert that (clinical) images are (generally) sparse or compressible in the wavelet domain and have low Total Variation (TV) in image space. However, CS theory does not restrict us to the model described above. All that is needed is a sparse or or compressible signal, a sufficient number of measurements and a “good” sampling matrix.

ARCS takes a different approach and exploits the inherent relationship between radial k -space profiles and the Radon Transform in an attempt to reconstruct the sinogram of the image space data. After reconstruction only a single non-linear transform is required to compute the image space data from the recovered sinogram.

3.2.1 Background

Suppose the k -space is fully sampled with $n_\theta \times n_\phi$ radial profiles, where n_θ is the number of azimuthal angles θ and n_ϕ is the number of polar angles ϕ . Furthermore,

let $\alpha \in \{\theta, \phi\}$ and suppose that the radial profiles have angles α_j such that

$$0 \leq \alpha_0 < \alpha_1 < \dots < \alpha_{n_\alpha-1} < \pi \quad \text{with} \quad \alpha_{j+1} - \alpha_j \leq \Delta\alpha,$$

where $\Delta\alpha$ satisfies the Nyquist criterion, see [Section 1.5](#). E.g., $\alpha_j \equiv j\Delta\alpha$. Assume that each radial profile consists of n_k equispaced radial distances k such that

$$-k_{max} \leq k_0 < k_1 < \dots < k_{n_k-1} \leq k_{max} \quad \text{with} \quad k_{j+1} - k_j \leq \Delta k,$$

where Δk satisfies the Nyquist criterion. E.g., $k_j \equiv (j - n_k/2)\Delta k$. Equispaced points are strictly speaking not necessary, but allow the use of the inverse FFT to compute the Radon Transform from the radial profiles.

ARCS exploits the inherent relationship between radially acquired k -space data and the Radon Transform. Recall from [Section 1.5](#) that for fixed angles θ and ϕ a radial profile is measured in k -space along the unit vector $\mathbf{u} = (c_\theta s_\phi, s_\theta s_\phi, c_\phi)^T$ at points $k_j \mathbf{u}$ and that

$$\sigma(k_j \mathbf{u}) = \widehat{P}_{\theta, \phi}(k_j),$$

holds. Define

$$\sigma \equiv \begin{bmatrix} \sigma(k_0 \mathbf{u}) \\ \sigma(k_1 \mathbf{u}) \\ \vdots \\ \sigma(k_{n_k-1} \mathbf{u}) \end{bmatrix} = \begin{bmatrix} \widehat{P}_{\theta, \phi}(k_1) \\ \widehat{P}_{\theta, \phi}(k_2) \\ \vdots \\ \widehat{P}_{\theta, \phi}(k_{n_k-1}) \end{bmatrix},$$

denote the Discrete Fourier transform by \mathcal{F} and let $\pi \equiv \mathcal{F}^{-1}\{\sigma\}$, then

$$\pi \approx \begin{bmatrix} P_{\theta, \phi}(k_1) \\ P_{\theta, \phi}(k_2) \\ \vdots \\ P_{\theta, \phi}(k_{n_k-1}) \end{bmatrix}.$$

The Radon Transform $P_{\theta, \phi}$ is the orthogonal projection of the image space to a single rotated line and the vector π is $P_{\theta, \phi}$ sampled along that line. It is possible to arrange each vector σ in a three dimensional grid Σ , such that k changes along the first dimension (column), θ along the second (row) and ϕ along the third (stack). The structure would somewhat resemble Rubik's cube. The equivalent arrangement of the vectors π in Π is often called a *sinogram*.¹ Π is easily obtained from Σ by a series of one-dimensional FFTs if the k_j are equispaced.

Suppose that each radial k -space profile is indeed fully sampled and that the k_j are equispaced. Then each radial k -space profile is isomorphic to the sinogram profile associated with the same angles, since the discrete (equispaced) Fourier Transform is an orthonormal linear transform. Ergo, we can assume without loss of generality that we sample the sinogram instead of the k -space and undersampling k -space in θ and ϕ is identical to undersampling the sinogram in θ and ϕ respectively.

Suppose there exists an orthonormal transform Ψ such that $x_* \equiv \Psi^* \Pi$ is sparse or compressible, e.g., a Daubechies wavelet operator. Define

$$N \equiv n_k \times n_\theta \times n_\phi \quad \text{and} \quad M \equiv n_k \times \left\lceil \frac{n_\theta}{L_\theta} \right\rceil \times \left\lceil \frac{n_\phi}{L_\phi} \right\rceil, \quad (3.2)$$

¹Although the term 'sinogram' is usually associated with the two-dimensional case, e.g., $n_\theta = 1$ or $n_\phi = 1$.

where $L_\theta \geq 1$ and $L_\phi \geq 1$ are acceleration factors. Let $R : \mathbb{C}^N \mapsto \mathbb{C}^M$ be the restriction of \mathbb{C}^N to a selection of M of its coordinates. Then $y_* \equiv R\Psi x_*$ samples a sparse or compressible signal with a partial orthonormal matrix. If R contains sufficient ‘randomness’ then $R\Psi$ is an example of an incoherent matrix and a candidate for exact signal recovery in CS theory, see also [Section 2.2](#).

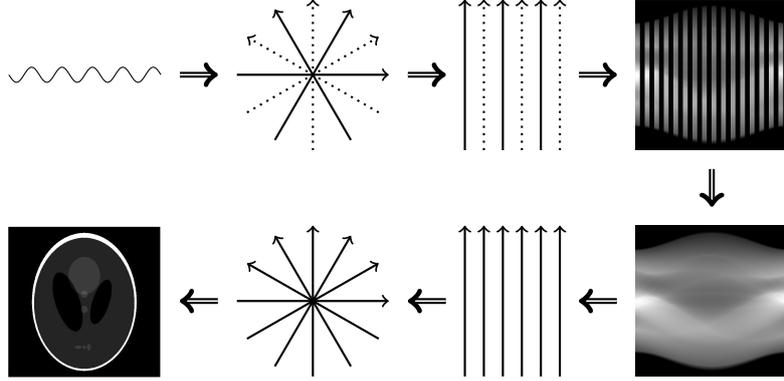


Figure 3.2: k -Space points are acquired with a radial trajectory. Only a part of the k -space is sampled (solid lines). Each radial profile can be arranged as the column of a matrix. The (acquired) profiles are inverse Fourier transformed to obtain part of the sinogram. The sinogram is recovered using compressed sensing. Every vertical profile in the sinogram is Fourier transformed to obtain all k -space profiles. The k -space profiles together form a fully sampled radial k -space. Gridding or the adjoint Nonequispaced Fast Fourier Transform is used to compute the image space signal.

Thus, we undersample the sinogram and obtain a noisy version $y \equiv y_* + \delta$ of y_* and use a CS algorithm to reconstruct (an approximation to) x_* from y . After reconstruction the image space signal is computed using (filtered) backprojection, k -space gridding or the adjoint NFFT. [Figure 3.2](#) shows the entire reconstruction chain for a two dimensional example. To summarize: *ARCS attempts to reconstruct an undersampled sinogram.*

3.2.2 Exploiting periodicity

In practice, radial profiles are always fully sampled, thus it makes sense to restrict the CS reconstruction to the undersampled dimension(s). So perhaps a series of $(d - 1)$ -dimensional reconstruction problems can solve the original d -dimensional reconstruction problem. The lower dimensional problems can easily be solved embarrassingly parallel. It still makes sense to minimize total variation in all dimensions however.

Observe that only the angular direction(s) are undersampled and that the sinogram is inherently periodic in this (these) direction(s). It is possible to exploit this periodicity in our wavelet transform and avoid all boundary effects that would otherwise be introduced by the wavelet. Consider the two dimensional case for simplicity. The goal is to extend Π to Π' with profiles $\Pi'_{n_\theta+1}, \Pi'_{n_\theta+2}, \dots, \Pi'_{2n_\theta}$ such that Π' is periodic in the angular direction θ . Each profile Σ_j contains the k -space signal along vector $\mathbf{u}_j = (c_{\theta_j}, s_{\theta_j})^T$ for some $0 \leq \theta_j < \pi$. Observe that it is natural to

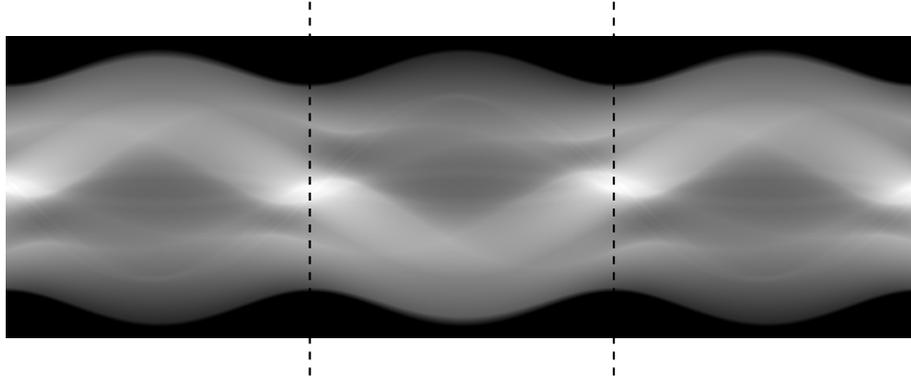


Figure 3.3: Sinograms have a natural periodic extension.

identify the $(n_\theta + j)$ -th profile with the angle $\pi + \theta_j$, such that $\mathbf{u}_{n_\theta+j} = (c_{\theta+\pi}, s_{\theta+\pi})^T$. Notice that $\mathbf{u}_{n_\theta+j} = -\mathbf{u}_j$, thus $\Sigma'_{n_\theta+j}$ should sample along the same line as Σ_j but in the other direction! By making a proper choice for k_0 through k_{n_k-1} the profiles $\Pi'_{n_\theta+j}$ can be obtained from Π_j . For example if $k_j \equiv (j - n_k/2)\Delta k$ then we could use the following MATLAB code, see also [Figure 3.3](#).

```
% 2D periodic extension.
sinogram = reshape(sinogram, nk, ntheta);
reversed = cat(1, sinogram(1,:), sinogram(end:-1:2,:));
periodic = cat(2, sinogram, reversed);

% 2D extraction.
upper = periodic(1:nk/2,1:ntheta);
lower = periodic(nk/2+1:-1:2,ntheta+1:2*ntheta);
sinogram = [upper; lower];
```

Perhaps it is more straightforward to make Π periodic by mirroring the profiles, e.g., identify $\Pi'_{n_\theta+1}$ with Π_{n_θ} , $\Pi'_{n_\theta+2}$ with $\Pi_{n_\theta-1}$ and so forth. However, in this case all n_k rows of Π' are needed to compute Π from Π' . The first $\lfloor n_k/2 \rfloor + 1$ rows suffice for the previous method. This is approximately a factor 2 difference in ‘work’.

The three dimensional “cones” case is slightly more complicated and the details depend on the convention used to represent spherical coordinates. In the case of this paper, the columns can be extended along ϕ in the same way as the two dimensional case. We do not know of a similar extension along θ , the mirror trick can be used however. To see the difference between the angles, let $u(\theta, \phi) = (c_\theta s_\phi, s_\theta s_\phi, c_\phi)^T$ and fix one of the angles while varying the other. For fixed θ_0 , $\phi \mapsto u(\theta_0, \phi)$ rotates through a plane that itself is rotated about the z -axis by θ_0 , in particular the behaviour of u is similar to the two-dimensional case. For fixed ϕ_0 , $\theta \mapsto u(\theta, \phi)$ precesses about the z -axis. Below follows example MATLAB code for the 3D case.

```
% 3D periodic extension.
sinogram = reshape(sinogram, nk, ntheta, nphi);
reversed = cat(1, sinogram(1,:,:), sinogram(end:-1:2,:,:));
extended = cat(3, sinogram, reversed);
```

```

mirrored = flipdim(extended, 2);
periodic = cat(2, extended, mirrored);

% 3D extraction.
upper = periodic(1:nk/2,1:ntheta,1:nphi);
lower = periodic(nk/2+1:-1:2,1:ntheta,nphi+1:2*nphi);
sinogram = [upper; lower];

```

3.2.3 Implementation

In this paper we attempt to reconstruct the undersampled sinogram with by solving a regularized minimization problem analogous to (3.1). The previous sections suggest the form

$$\operatorname{argmin}_x \frac{1}{2} \|\Phi\Psi x - b\|_2^2 + \alpha \|\Psi x\|_{TV} + \beta \|x\|_1, \quad (3.3)$$

where b is the undersampled sinogram, x the sparse signal and $\Phi\Psi$ our partial orthogonal sensing matrix defined by a restriction Φ and a wavelet operator Ψ . Let $z = \Psi x$ such that $x = \Psi^* z$, then we can write the equation above as

$$\operatorname{argmin}_z \frac{1}{2} \|\Phi z - b\|_2^2 + \alpha \|z\|_{TV} + \beta \|\Psi^* z\|_1, \quad (3.4)$$

which has the same form as (3.1). Equation 3.3 is sometimes indicated as the *synthesis* formulation and Equation 3.4 as the *analysis* formulation. Simply put, in the synthesis formulation the signal is sparse and “built up”, whereas the signal is dense in the analysis formulation and “broken down”. Remark that if Φ includes a one dimensional Fourier Transform for each profile, then either formulation allows ARCS to be used with *Single Point Imaging*.

We solve (3.1) and (3.4) using a modified version of the *Fast Composite Splitting Algorithm* proposed by Huang et al. [9], which we will describe below. Huang et al. derive FCSA from FISTA Section 2.3.3.3 and replace the minimization

$$\operatorname{argmin}_x \left\{ \frac{L}{2} \|x - x_g\|_2 + \lambda \|x\|_1 \right\}$$

with

$$\operatorname{argmin}_x \left\{ \frac{L}{2} \|x - x_g\|_2 + \alpha \|x\|_{TV} + \beta \|\Psi^* x\|_1 \right\},$$

which they split in the two minimizations

$$\operatorname{argmin}_x \left\{ \frac{L}{2} \|x - x_g\|_2 + \alpha \|x\|_{TV} \right\} \quad \text{and} \quad \operatorname{argmin}_x \left\{ \frac{L}{2} \|x - x_g\|_2 + \beta \|\Psi^* x\|_1 \right\}.$$

We recognize the minimizations above as *proximal maps* and recall the definition

$$\operatorname{prox}_L(g)(y) \equiv \operatorname{argmin}_x \left\{ \frac{L}{2} \|x - y\|_2 + g(x) \right\}$$

from Section 2.3.1. We give an algorithmic description of FCSA in Algorithm 12. The operator P_C is an orthogonal projection onto a convex set C that acts as the solution space and $F(x)$ is the objective function.

Algorithm 12: Fast Composite Splitting Algorithm

Input: Operators Φ and Ψ , measurement b , $L \geq \lambda_{\max}(\Phi^T \Phi)$, initial guess $x \in \mathbb{R}^{m \times n}$ and regularization parameters α and β .

Output: An approximate solution $x \in C$ to (3.1).

begin

```
for  $k \leftarrow 1, 2, \dots, k_{\max}$  do
   $x_g \leftarrow r - (1/L)\Phi^T(\Phi r - b)$ 
   $x_1 \leftarrow \text{prox}_L(2\alpha\|x\|_{TV})(x_g)$ 
   $x_2 \leftarrow \text{prox}_L(2\beta\|\Psi x\|_1)(x_g)$ 
   $x' \leftarrow (x_1 + x_2)/2$ 
   $x' \leftarrow P_C(x')$ 
   $t' \leftarrow (1 + \sqrt{1 + 4t^2})/2$ 
   $r \leftarrow x + ((t - 1)/t')(x' - x)$ 
   $x \leftarrow x', t \leftarrow t'$ 
```

The work done by Huang et al. is heavily based on work by Beck and Teboulle, specifically FISTA applied to ℓ_1 minimization [2] and FISTA applied to TV minimization [3]. The latter includes an improved version of FISTA, called *Monotone Fast Iterative Soft-Thresholding Algorithm* (MFISTA), that guarantees monotone convergence (in the sense that the objective value can only decrease). The change from FISTA to MFISTA is minimal implementation wise, but interestingly, not included in FCSA. We therefore introduce the *Monotone Fast Composite Splitting Algorithm* in Algorithm 13.

Algorithm 13: Monotone Fast Composite Splitting Algorithm

Input: Operators Φ and Ψ , measurement b , $L \geq \lambda_{\max}(\Phi^T \Phi)$, initial guess $x \in \mathbb{R}^{m \times n}$ and regularization parameters α and β .

Output: An approximate solution $x \in C$ to (3.1).

begin

```
for  $k \leftarrow 1, 2, \dots, k_{\max}$  do
   $x_g \leftarrow r - (1/L)\Phi^T(\Phi r - b)$ 
   $x_1 \leftarrow \text{prox}_L(2\alpha\|x\|_{TV})(x_g)$ 
   $x_2 \leftarrow \text{prox}_L(2\beta\|\Psi x\|_1)(x_g)$ 
   $z \leftarrow (x_1 + x_2)/2$ 
   $z \leftarrow P_C(z)$ 
   $x' \leftarrow \text{argmin}\{F(x') : x' = z, x\}$ 
   $t' \leftarrow (1 + \sqrt{1 + 4t^2})/2$ 
   $t' \leftarrow (1 + \sqrt{1 + 4t^2})/2$ 
   $r \leftarrow x' + (t/t')(z - x') + ((t - 1)/t')(x' - x)$ 
   $x \leftarrow x', t \leftarrow t'$ 
```

The minimization $\text{prox}_L(\alpha\|x_g\|_{TV})$ is non trivial, but can be solved with *Fast Gradient Projection* (FGP) [3]. While FGP works great for MFCSA applied to (3.1), it does not take advantage of the periodic boundary conditions when solving (3.4). Below we present the changes we made to FGP to exploit the periodicity of a

sinogram.

- With periodicity along j the isotropic TV becomes

$$\begin{aligned} \mathbf{x} \in \mathbb{R}^{m \times n}, \quad TV_I(\mathbf{x}) = & \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sqrt{(x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2} \\ & + \sum_{i=1}^{m-1} \sqrt{(x_{i,n} - x_{i+1,n})^2 + (x_{i,n} - x_{i,1})^2} \\ & + \sum_{j=1}^{n-1} |x_{m,j} - x_{m,j+1}| \\ & + |x_{m,n} - x_{m,1}|. \end{aligned}$$

- \mathcal{P} is the set of matrix-pairs $\mathbf{p} \in \mathbb{R}^{(m-1) \times n}$ and $\mathbf{q} \in \mathbb{R}^{m \times n}$ that satisfy

$$\begin{aligned} p_{i,j}^2 + q_{i,j}^2 &\leq 1, \quad i = 1, \dots, m-1, j = 1, \dots, n \\ |q_{m,j}| &\leq 1, \quad j = 1, \dots, n. \end{aligned}$$

- The linear operation $\mathcal{L} : \mathbb{R}^{(m-1) \times n} \times \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{m \times n}$ is defined by the formula

$$\mathcal{L}(\mathbf{p}, \mathbf{q})_{i,j} = p_{i,j} + q_{i,j} - p_{i-1,j} - q_{i,j-1}, \quad i = 1, \dots, m, j = 1, \dots, n,$$

where we assume that $p_{0,j} = p_{m,j} \equiv 0$ and $q_{i,0} \equiv q_{i,n}$ for every $i = 1, \dots, m$ and $j = 1, \dots, n$. Specifically we have

$$\begin{aligned} \mathcal{L}(\mathbf{p}, \mathbf{q})_{1,1} &= p_{1,1} + q_{1,1} && - q_{1,n}, \\ \mathcal{L}(\mathbf{p}, \mathbf{q})_{1,j} &= p_{1,j} + q_{1,j} && - q_{1,j-1}, \quad j = 2, \dots, n \\ \mathcal{L}(\mathbf{p}, \mathbf{q})_{i,1} &= p_{i,1} + q_{i,1} - p_{i-1,1} - q_{i,n}, && i = 2, \dots, m-1 \\ \mathcal{L}(\mathbf{p}, \mathbf{q})_{i,j} &= p_{i,j} + q_{i,j} - p_{i-1,j} - q_{i,j-1}, && i = 2, \dots, m-1, j = 2, \dots, n \\ \mathcal{L}(\mathbf{p}, \mathbf{q})_{m,1} &= && + q_{m,1} - p_{m-1,1} - q_{m,n}, \\ \mathcal{L}(\mathbf{p}, \mathbf{q})_{m,j} &= && + q_{m,j} - p_{m-1,j} - q_{m,j-1}, \quad j = 2, \dots, n. \end{aligned}$$

- The operator $\mathcal{L}^T : \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{(m-1) \times n} \times \mathbb{R}^{m \times n}$ is given by

$$\mathcal{L}^T(\mathbf{x}) = (\mathbf{p}, \mathbf{q}),$$

where $\mathbf{p} \in \mathbb{R}^{(m-1) \times n}$ and $\mathbf{q} \in \mathbb{R}^{m \times n}$ are the matrices defined by

$$\begin{aligned} p_{i,j} &= x_{i,j} - x_{i+1,j}, \quad i = 1, \dots, m-1, j = 1, \dots, n \\ q_{i,j} &= x_{i,j} - x_{i,j+1}, \quad i = 1, \dots, m, j = 1, \dots, n-1 \\ q_{i,n} &= x_{i,n} - x_{i,1}, \quad i = 1, \dots, m. \end{aligned}$$

- In the proof of proposition 4.1 in Beck and Teboulle 2009b [3] need the

definition

$$\begin{aligned}
T(x, \mathbf{p}, \mathbf{q}) &= \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} [p_{i,j}(x_{i,j} - x_{i+1,j}) + q_{i,j}(x_{i,j} - x_{i,j+1})] \\
&\quad + \sum_{i=1}^{m-1} [p_{i,n}(x_{i,n} - x_{i+1,n}) + q_{i,n}(x_{i,n} - x_{i,1})] \\
&\quad + \sum_{j=1}^{n-1} [q_{m,j}(x_{m,j} - x_{m,j+1})] \\
&\quad + q_{m,n}(x_{m,n} - x_{m,1}).
\end{aligned}$$

- And finally, for a pair (\mathbf{p}, \mathbf{q}) , the projection $P_{\mathcal{P}}(\mathbf{p}, \mathbf{q})$ is given by $P_{\mathcal{P}}(\mathbf{p}, \mathbf{q}) = (\mathbf{r}, \mathbf{s})$ where $\mathbf{r} \in \mathbb{R}^{(m-1) \times n}$ and $\mathbf{s} \in \mathbb{R}^{m \times n}$ are given by

$$r_{i,j} = \frac{p_{i,j}}{\max\{1, \sqrt{p_{i,j}^2 + q_{i,j}^2}\}} \quad i = 1, \dots, m-1, j = 1, \dots, n$$

and

$$s_{i,j} = \begin{cases} \frac{q_{i,j}}{\max\{1, \sqrt{p_{i,j}^2 + q_{i,j}^2}\}} & i = 1, \dots, m-1, j = 1, \dots, n \\ \frac{q_{m,j}}{\max\{1, |q_{m,j}|\}} & j = 1, \dots, n. \end{cases}$$

An algorithmic description of FGP is given in [Algorithm 14](#).

Algorithm 14: Fast Gradient Projection

Input: Observed image \mathbf{b} , regularization parameter λ and number of iterations k_{max} .

Output: An optimal solution $x_* \in C$ to $\text{prox}_1(\lambda \|x\|_{TV})(\mathbf{b})$ up to a tolerance.

begin

```

     $t \leftarrow 1, (\mathbf{r}, \mathbf{s}) \leftarrow (\mathbf{p}, \mathbf{q}) \leftarrow (\mathbf{0}, \mathbf{0}) \in \mathbb{R}^{(m-1) \times n} \times \mathbb{R}^{m \times n}$ 
    for  $k \leftarrow 1, 2, \dots, k_{max}$  do
         $(\mathbf{p}', \mathbf{q}') \leftarrow P_{\mathcal{P}} \left[ (\mathbf{r}, \mathbf{s}) + (8\lambda)^{-1} \mathcal{L}^T (P_C[\mathbf{b} - \lambda \mathcal{L}(\mathbf{r}, \mathbf{s})]) \right]$ 
         $t' \leftarrow (1 + \sqrt{1 + 4t^2})/2$ 
         $(\mathbf{r}, \mathbf{s}) \leftarrow (\mathbf{p}', \mathbf{q}') + ((t-1)/t')(\mathbf{p}' - \mathbf{p}, \mathbf{q}' - \mathbf{q})$ 
         $t \leftarrow t', (\mathbf{p}, \mathbf{q}) \leftarrow (\mathbf{p}', \mathbf{q}')$ 
     $x_* \leftarrow P_C[\mathbf{b} - \lambda \mathcal{L}(\mathbf{p}, \mathbf{q})]$ 

```

3.3 Motivation

Before we continue, let us take a look at the “compressibility” of sinograms. The compressibility of sinograms is an empirical observation just like the compressibility of images, which many CS accelerated acquisition methods heavily rely on. Sinograms tend to be “smooth”, which is intuitively unsurprising since each profile in the

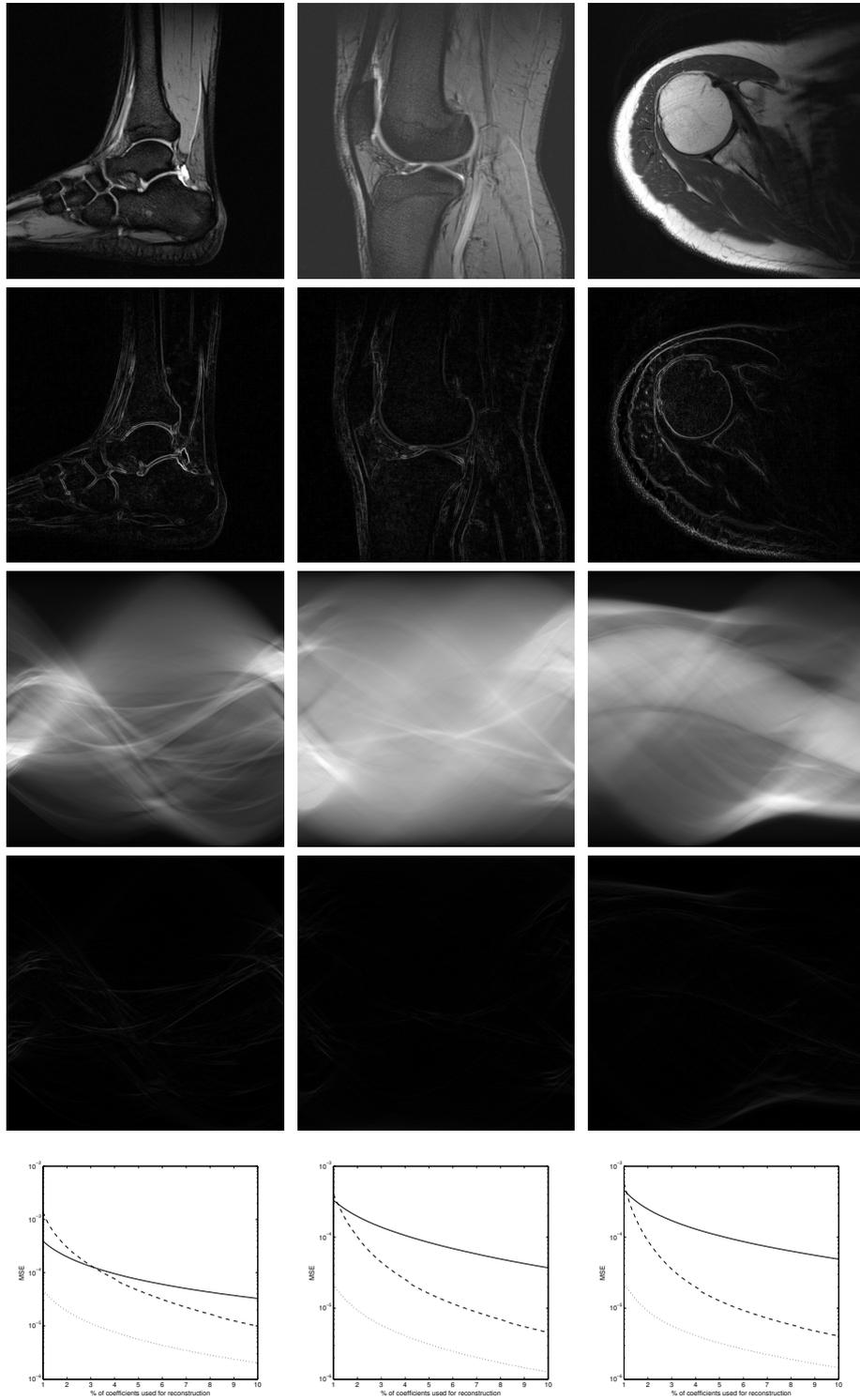


Figure 3.4: Three motivation examples for ARCS, see [Section 3.3](#) for a detailed explanation.

sinogram is the orthogonal projection of the image space onto a rotated line. The lines and profiles rotate slowly and the projections smooth the image space signal.²

See Figure 3.4 for motivating examples. There are three MR images, one of an ankle, one of a knee and one of a shoulder. The original images are shown in the top row. Below are the variation images. The row in the middle shows the sinograms of the MR images and their variation images follow on the row below. The intensities of all variation images are doubled for visibility. The final row is especially interesting, the graphs represent the reconstruction quality when all but a small percentage of the largest wavelet coefficients are replaced by zero. The solid line is the reconstruction quality for the original image, the dashed line for the sinogram with wavelet compression in the angular direction only and the dotted line for the sinogram with wavelet compression in both directions. The reconstruction “quality” is measured as the Mean Squared Error (MSE) and the values of the original are scaled between 0 and 1. The maximum error and the Correlation Coefficient (CC) give similar results. The wavelet was in both cases Daubechies 4 with the number of levels tuned for “optimal” results.

Table 3.1: For the three examples in Figure 3.4 this table shows the ℓ_2 -norm ratio between the image and its variation image, the maximum value in the variation image and the same values for the sinogram and its variation. The image values are marked by (I) and the sinogram values by (S).

	ratio (I)	max (I)	ratio (S)	max (S)
Ankle	0.1845	0.4943	0.0273	0.1418
Knee	0.1325	0.2624	0.0118	0.0707
Shoulder	0.1167	0.3876	0.0187	0.0768

The three examples clearly show that the sinograms contain less “variation”, e.g., they are smoother, see also Table 3.1. Furthermore, the sinograms are better approximated with large wavelet coefficients than their image space counterparts, even with compression in only one direction. The only exception is the ankle if less than about 3.5% of the large wavelet coefficients are used. Although these three images do not provide definite proof, they do motivate the approach taken by ARCS.

3.4 Numerical experiments

3.4.1 Test setup

We use four different datasets to test ARCS, the Shepp-Logan phantom sampled in k -space [11] and the Ankle, Knee and Shoulder from Figure 3.4 transformed to a radial k -space trajectory using NFFT [12]. We sample the Shepp-Logan phantom with 256 radial profiles and oversample the readout direction twice with 512 points for each profile. The images are 512×512 pixels, so we use 512 radial profiles of 1024 points each.

We undersample the k -space radially with acceleration factors $L = 2, 4$ and 8 and, for each L , add complex Gaussian white noise such that the data obtains *Signal-to-noise ratios* (SNRs) $\infty, 50$ and 25 . For simplicity we use uniform and

²Recall that the projections are integrals.

non-random sampling. In theory we should use random incoherent sampling to prevent coherent aliasing artifacts. However, initial testing seemed to indicate that uniformity was more important than randomness. The reconstructed sinogram is transformed back to k -space and subsequently transformed to image space using the adjoint NFFT.

We solve the “traditional” minimization problem (3.1) with MFCSA and NFFT and solve (3.4) for ARCS with our periodic MFCSA. We shall call the former approach NFFT-MFCSA and the latter ARCS-MFCSA to distinguish the two. In either case the projection P_C is the identity since the range of the data can be hard to know in practice. Nor do we use any post processing. We run both algorithms until they ‘converge’, which in this case means no decrease in the objective value for five consecutive iterations. We use $\alpha = 2 \cdot 10^{-4}$ and $\beta = 10^{-5}$ if $\text{SNR} = \infty$ and $\alpha = 10^{-3}$ and $\beta = 10^{-4}$ otherwise; these parameters seemed to produce ‘very good’ results in most cases (but are unlikely to be the ‘best’ in any case). Furthermore, we use a series of one dimensional Daubechies-4 wavelets with periodic borders for ARCS-MFCSA and use seven levels for the Shepp-Logan phantom and eight levels for the images. For NFFT-MFCSA we use a two dimensional Daubechies-4 wavelet with half-point symmetric padding at the borders and use six levels in all cases. We compare ARCS-MFCSA and NFFT-MFCSA to each other and also to the case (indicated by “zero” or “no reconstruction”) where we use NFFT directly on the undersampled data to compute the image space signal.

Let X be the vector of length N containing all the original pixel values and $X_i \in [0, 1]$ for all $i = 1, \dots, N$. Let Y be a reconstruction of X and scaled such that $\|X - Y\|_2 \leq \|X - tY\|_2$ for all $t \in \mathbb{R}$ (recall that $\|X - tY\|_2$ is minimal for $t = \langle X, Y' \rangle / \langle Y', Y' \rangle$). We quantify the reconstruction quality by the following measures. The *Mean Squared Error* (MSE) defined by

$$\text{MSE} \equiv \frac{1}{N} \sum_{i=1}^N (X_i - Y_i)^2$$

is the mean error per pixel. A similar value is the *Peak Signal to Noise Ratio* (PSNR)

$$\text{PSNR} \equiv -10 \log_{10}(\text{MSE})$$

expressed in logarithmic decibel scale. The *Maximum Error* (MaxErr)

$$\text{MaxErr} \equiv \|X - Y\|_\infty$$

is another important measure. Sometimes its useful to know the energy loss or gain

$$\ell_2\text{-Ratio} \equiv \frac{\|Y\|_2^2}{\|X\|_2^2}.$$

The *Correlation Coefficient* (CC)

$$\text{CC} = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^N (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^N (Y_i - \bar{Y})^2}}$$

is another widely used measure and is scale invariant. The timings are the best of five consecutive runs of an MATLAB implementation running on an Intel Core i7-3770 processor.

3.4.2 Results and discussion

Initial tests showed that results were rather poor with piecewise one dimensional compression and one dimensional TV minimization. The results were a lot better with one dimensional compression and two dimensional TV minimization. Two dimensional compression and two dimensional TV minimization seemed to be a little bit better still, but more thorough testing is needed to confirm this observation and to test the strengths and weaknesses of one dimensional versus two dimensional compression.

The results can be found in [Table 3.2](#) through [Table 3.5](#) and [Figure 3.5](#) through [Figure 3.16](#).

[Table 3.2](#) shows that ARCS-MFCSA improves upon NFFT-MFCSA by every measure. We notice that ARCS-MFCSA is between five to twenty times as fast as NFFT-MFCSA and at the same time the MSE is generally only half as large.

[Table 3.3](#) shows similar results for $L = 2$, though for $L = 4$ and $SNR = \infty$ and $L = 8$ NFFT-MFCSA is slightly better. The maximum error is generally lower for NFFT-MFCSA when the SNR is high. ARCS-MFCSA is about two to twenty times faster.

In [Table 3.4](#) we see that the reconstruction quality is better with NFFT-MFCSA for $L = 2$ and $SNR \neq \infty$ and $L = 4$ and $SNR = 50$. ARCS-MFCSA has lower MSE and better ℓ_2 -ratios and CC in the remaining cases, although NFFT-MFCSA has often a lower maximum error. For this example ARCS-MFCSA is between ten and thirty times as fast.

The final dataset in [Table 3.5](#) shows again that NFFT-MFCSA often has a lower maximum error. However, the other measures show better results for ARCS-MFCSA, except when $L = 8$ and $SNR = \infty$. ARCS-MFCSA is again five to twenty times as fast.

The runtime is often longer when $SNR = \infty$. This is a result of the lower α and β . Another remarkable results is the often lower maximum error for NFFT-MFCSA, even when ARCS-MFCSA has a lower MSE and higher CC. A possible source for this peculiarity is the TV regularizer. Recall that the Radon Transform smooths the image, thus a large maximum error in the image is likely to manifest as a smaller in the sinogram. In addition, NFFT-MFCSA performs TV-minimization (basically denoising) to the final result, e.g. the image, whereas ARCS-MFCSA does not. Note that denoising could be added as a post processing step to ARCS-MFCSA.

The figures show that reconstruction for $L = 2$ is quite good for, even for low SNR . For $L = 4$ there is some loss of high-frequencies, though not all small details are lost, compare for example the muscle in [Figure 3.8](#) and [Figure 3.9](#). For $L = 8$ we lose a lot more high-frequencies and see a lot more streaking artifacts, e.g. [Figure 3.7](#), [Figure 3.10](#) and [Figure 3.16](#). The latter figures show less streaking for ARCS-MFCSA than for NFFT-MFCSA. The reconstruction quality by NFFT-MFCSA for $L = 8$ and $SNR = 25$ is visually rather poor compared to ARCS-MFCSA. In fact, the visual reconstruction quality of ARCS-MFCSA seems to decay graceful even for $L = 8$, which does not seem the case for NFFT-MFCSA.

We conclude this section by remarking that regularization acts as a low pass filter, so some loss of high frequencies was to be expected.

3.5 Conclusion

Our tests show that ARCS is competitive in quality with traditional CS and is five to twenty times as fast at the same time. In addition, ARCS is a flexible approach and can be applied in two and three dimensions. Therefore, we believe that ARCS is a novel approach that warrants additional research.

Future research topics include, or could include, the use of co-RASOR [21] to localize small paramagnetic objects with high positive contrast while suppressing long T_2^* components, a graphics processing unit implementation for increased reconstruction speed [16, 13], use of partial Fourier techniques such as homodyne reconstruction [17] (each k -space profile has conjugate symmetry), use of shearlets instead of wavelets for improved sparsity and reconstruction, use of non-Cartesian SENSE [20, 19], combined reconstruction with radial GRAPPA (static and real-time reconstruction of undersampled non-Cartesian data with GRAPPA is extensively researched [7, 1, 8, 23, 24, 22, 6, 4]) and last but not least, (incoherent) sampling patterns tuned for ARCS such as Golden-Angle sampling [5, 18].

Table 3.2: Shepp-Logan

Method	L	SNR	PSNR	MSE	MaxErr	Ratio	CC	Time (s)
Zero	2	∞	46.6	0.000 02	0.0467	1.000	1.000	
		50	25.4	0.002 87	0.2355	0.976	0.981	
		25	19.7	0.010 70	0.4581	0.912	0.922	
	4	∞	36.7	0.000 21	0.1090	0.998	0.999	
		50	22.4	0.005 75	0.3288	0.952	0.960	
		25	17.3	0.018 72	0.5506	0.845	0.853	
	8	∞	27.6	0.001 72	0.2693	0.986	0.989	
		50	19.2	0.012 03	0.4787	0.901	0.912	
		25	15.2	0.030 47	0.7036	0.748	0.741	
NFFT	2	∞	39.1	0.000 12	0.1004	0.999	0.999	12.1
		50	33.5	0.000 44	0.1622	0.996	0.997	9.1
		25	29.0	0.001 26	0.4712	0.990	0.992	8.8
	4	∞	35.5	0.000 28	0.1489	0.998	0.998	15.7
		50	31.3	0.000 75	0.2136	0.994	0.995	9.7
		25	28.1	0.001 55	0.3470	0.987	0.989	9.6
	8	∞	31.7	0.000 68	0.2043	0.994	0.995	20.7
		50	27.0	0.001 99	0.2747	0.984	0.986	13.6
		25	27.2	0.001 92	0.2917	0.984	0.987	11.3
ARCS	2	∞	46.2	0.000 02	0.0945	1.000	1.000	0.6
		50	36.8	0.000 21	0.1239	0.998	0.998	0.4
		25	31.4	0.000 73	0.1908	0.994	0.995	0.4
	4	∞	40.4	0.000 09	0.1423	0.999	0.999	2.1
		50	35.1	0.000 31	0.1495	0.997	0.998	0.5
		25	31.7	0.000 68	0.2491	0.994	0.995	0.6
	8	∞	34.8	0.000 33	0.1735	0.997	0.998	3.6
		50	31.9	0.000 64	0.2017	0.995	0.995	0.7
		25	30.3	0.000 94	0.2718	0.992	0.993	0.7

Table 3.3: Ankle

Method	L	SNR	PSNR	MSE	MaxErr	Ratio	CC	Time (s)
Zero	2	∞	44.4	0.000 04	0.0376	0.999	0.999	
		50	30.6	0.000 87	0.1570	0.968	0.977	
		25	25.0	0.003 19	0.3161	0.882	0.913	
	4	∞	33.8	0.000 41	0.1459	0.985	0.988	
		50	27.0	0.001 99	0.2482	0.927	0.947	
		25	22.5	0.005 65	0.4982	0.791	0.835	
	8	∞	27.4	0.001 80	0.2601	0.933	0.949	
		50	23.6	0.004 33	0.4018	0.840	0.876	
		25	20.5	0.008 95	0.6380	0.669	0.712	
NFFT	2	∞	39.3	0.000 12	0.0920	0.996	0.997	56.7
		50	35.2	0.000 30	0.1592	0.989	0.993	42.2
		25	31.0	0.000 80	0.4388	0.970	0.980	39.1
	4	∞	35.8	0.000 26	0.1785	0.990	0.993	68.9
		50	32.2	0.000 61	0.2886	0.978	0.985	46.2
		25	31.5	0.000 71	0.4364	0.974	0.982	46.1
	8	∞	32.1	0.000 61	0.3075	0.977	0.983	87.4
		50	29.9	0.001 01	0.3725	0.963	0.973	72.0
		25	29.1	0.001 23	0.3960	0.955	0.967	53.3
ARCS	2	∞	40.8	0.000 08	0.1166	0.997	0.998	9.0
		50	36.6	0.000 22	0.1725	0.992	0.994	2.1
		25	33.8	0.000 41	0.1943	0.985	0.988	1.8
	4	∞	34.3	0.000 37	0.2915	0.986	0.990	29.7
		50	32.6	0.000 55	0.3127	0.980	0.985	12.3
		25	31.9	0.000 65	0.3096	0.976	0.982	3.0
	8	∞	29.5	0.001 11	0.4031	0.959	0.969	58.5
		50	29.1	0.001 24	0.4121	0.954	0.965	9.7
		25	28.9	0.001 30	0.4184	0.952	0.963	10.9

Table 3.4: Knee

Method	L	SNR	PSNR	MSE	MaxErr	Ratio	CC	Time (s)
Zero	2	∞	43.9	0.000 04	0.0357	1.000	0.999	
		50	23.7	0.004 30	0.3494	0.952	0.925	
		25	19.3	0.011 67	0.5486	0.870	0.803	
	4	∞	34.9	0.000 32	0.1148	0.996	0.994	
		50	21.2	0.007 53	0.4536	0.916	0.871	
		25	17.9	0.016 30	0.6756	0.818	0.725	
	8	∞	29.0	0.001 26	0.2074	0.986	0.978	
		50	19.2	0.012 04	0.5780	0.866	0.797	
		25	16.9	0.020 63	0.7423	0.770	0.651	
NFFT	2	∞	36.9	0.000 20	0.1447	0.998	0.997	55.2
		50	33.9	0.000 40	0.2079	0.995	0.993	49.3
		25	29.5	0.001 12	0.2639	0.987	0.980	39.9
	4	∞	34.5	0.000 35	0.2252	0.996	0.994	79.0
		50	32.9	0.000 52	0.2631	0.994	0.991	57.6
		25	29.0	0.001 27	0.3004	0.986	0.977	47.1
	8	∞	31.9	0.000 65	0.3270	0.993	0.989	111.6
		50	31.0	0.000 79	0.3470	0.991	0.986	76.5
		25	29.7	0.001 07	0.3586	0.988	0.981	94.8
ARCS	2	∞	40.0	0.000 10	0.1722	0.999	0.998	1.9
		50	33.3	0.000 47	0.2467	0.995	0.992	1.9
		25	28.3	0.001 47	0.3103	0.984	0.974	1.9
	4	∞	35.3	0.000 30	0.2403	0.997	0.995	8.0
		50	32.0	0.000 63	0.2769	0.993	0.989	3.3
		25	29.3	0.001 17	0.3712	0.987	0.979	3.0
	8	∞	31.5	0.000 71	0.3503	0.992	0.987	10.2
		50	30.0	0.001 00	0.3542	0.989	0.982	4.6
		25	28.7	0.001 34	0.3712	0.985	0.976	5.0

Table 3.5: Shoulder

Method	L	SNR	PSNR	MSE	MaxErr	Ratio	CC	Time (s)
Zero	2	∞	40.3	0.000 09	0.0611	0.999	0.999	
		50	25.2	0.003 04	0.2864	0.968	0.974	
		25	19.7	0.010 64	0.5565	0.887	0.905	
	4	∞	32.9	0.000 51	0.1298	0.995	0.996	
		50	22.2	0.006 00	0.4567	0.936	0.948	
		25	17.4	0.018 23	0.7340	0.806	0.829	
	8	∞	27.7	0.001 70	0.2882	0.982	0.985	
		50	19.4	0.011 54	0.6211	0.877	0.896	
		25	15.5	0.028 34	0.8886	0.698	0.712	
NFFT	2	∞	36.6	0.000 22	0.1699	0.998	0.998	55.5
		50	32.9	0.000 52	0.2676	0.994	0.996	55.0
		25	27.9	0.001 63	0.5914	0.983	0.987	38.6
	4	∞	34.2	0.000 38	0.2682	0.996	0.997	69.1
		50	31.5	0.000 71	0.3404	0.992	0.994	63.9
		25	28.9	0.001 29	0.3352	0.986	0.990	45.1
	8	∞	31.5	0.000 70	0.4043	0.993	0.994	88.2
		50	29.6	0.001 10	0.4191	0.988	0.991	68.0
		25	27.3	0.001 85	0.3877	0.980	0.985	53.7
ARCS	2	∞	39.4	0.000 12	0.1972	0.999	0.999	2.1
		50	33.5	0.000 45	0.2909	0.995	0.996	2.1
		25	29.4	0.001 16	0.3169	0.988	0.990	1.8
	4	∞	34.4	0.000 36	0.3018	0.996	0.997	14.1
		50	31.8	0.000 66	0.3787	0.993	0.994	3.2
		25	29.6	0.001 09	0.4095	0.988	0.991	3.0
	8	∞	30.8	0.000 83	0.3870	0.991	0.993	10.7
		50	29.8	0.001 05	0.4585	0.989	0.991	4.4
		25	28.8	0.001 33	0.4668	0.986	0.988	4.2

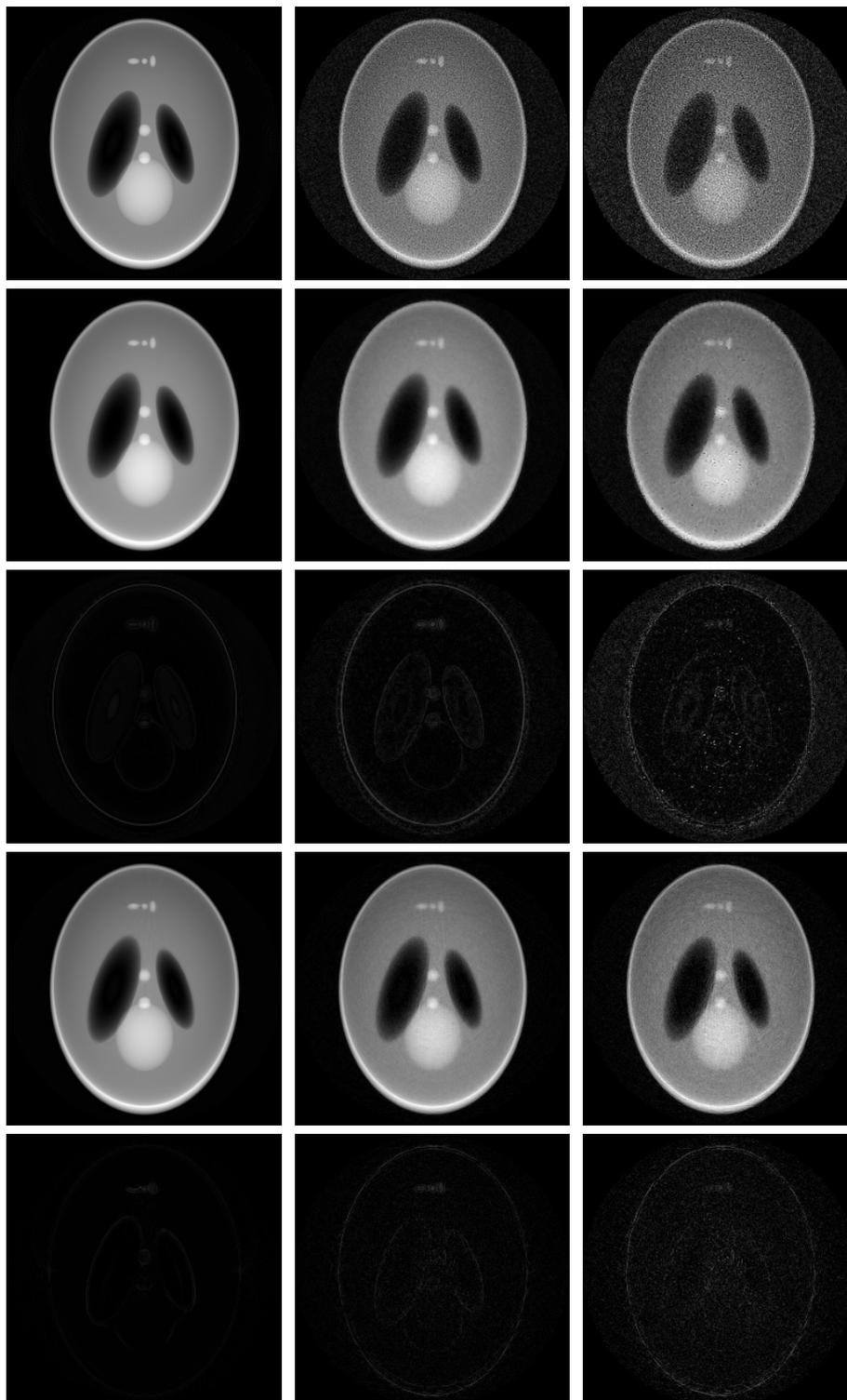


Figure 3.5: Shepp-Logan phantom for $L = 2$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

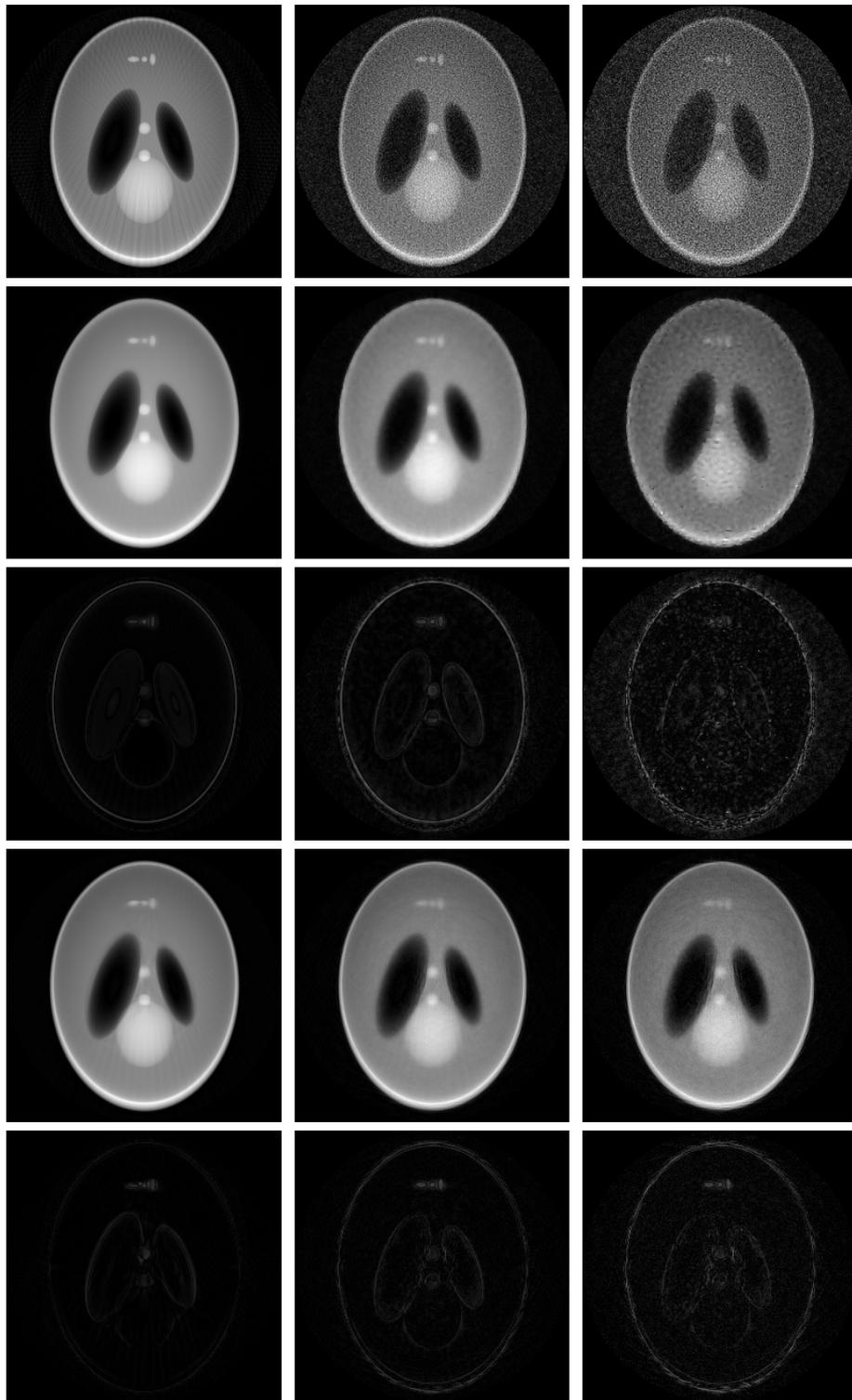


Figure 3.6: Shepp-Logan phantom for $L = 4$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

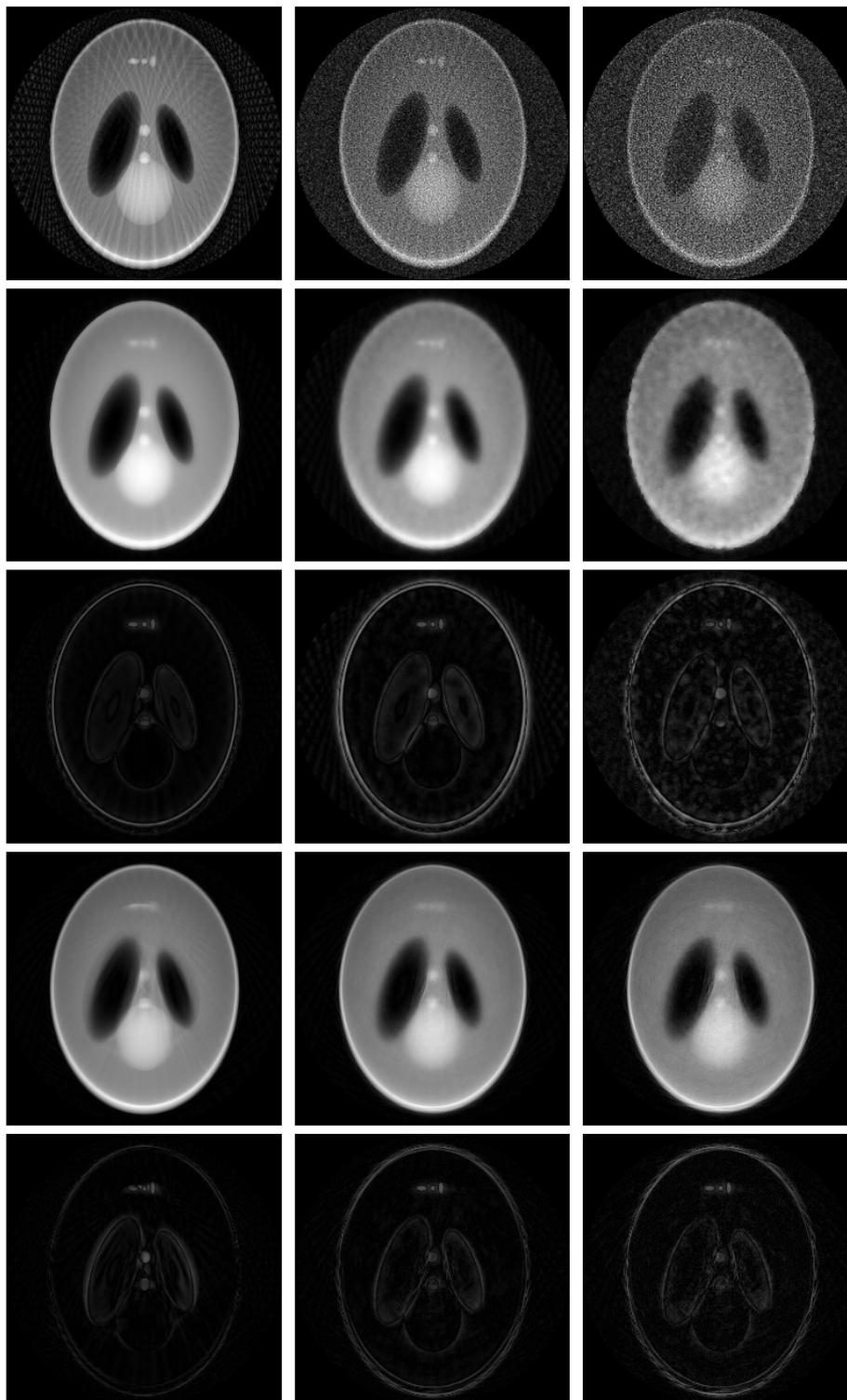


Figure 3.7: Shepp-Logan phantom for $L = 8$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

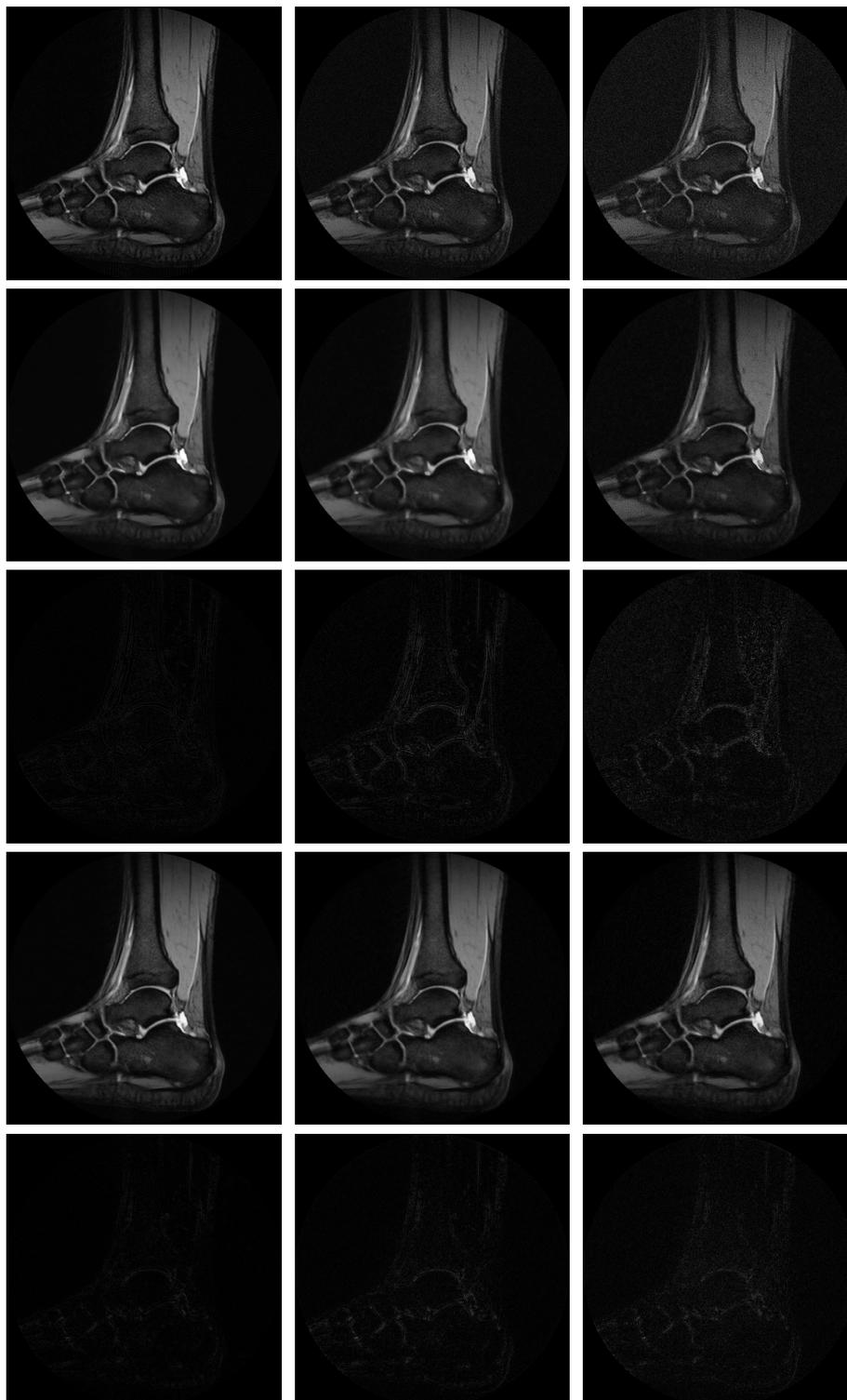


Figure 3.8: Ankle for $L = 2$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

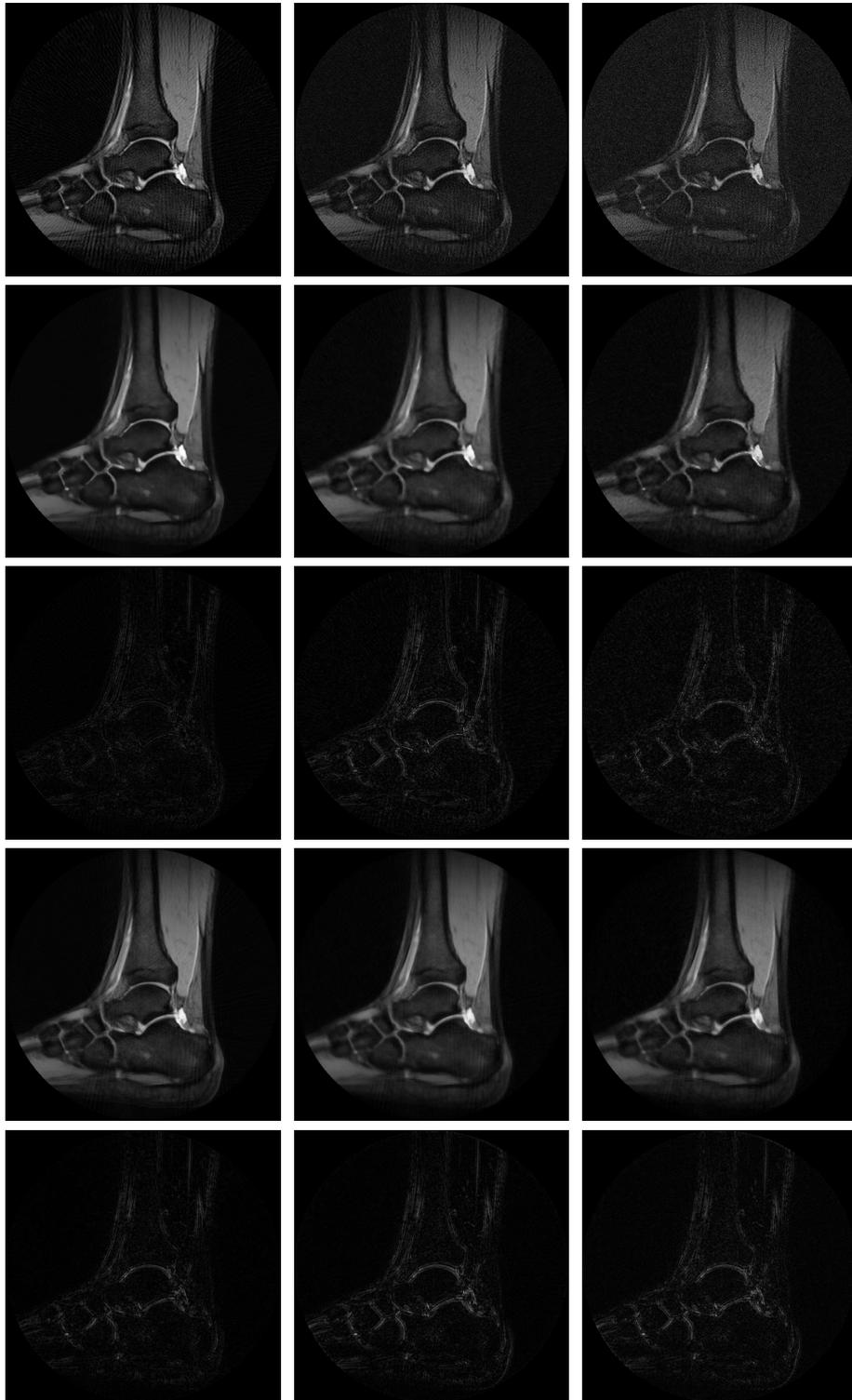


Figure 3.9: Ankle for $L = 4$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

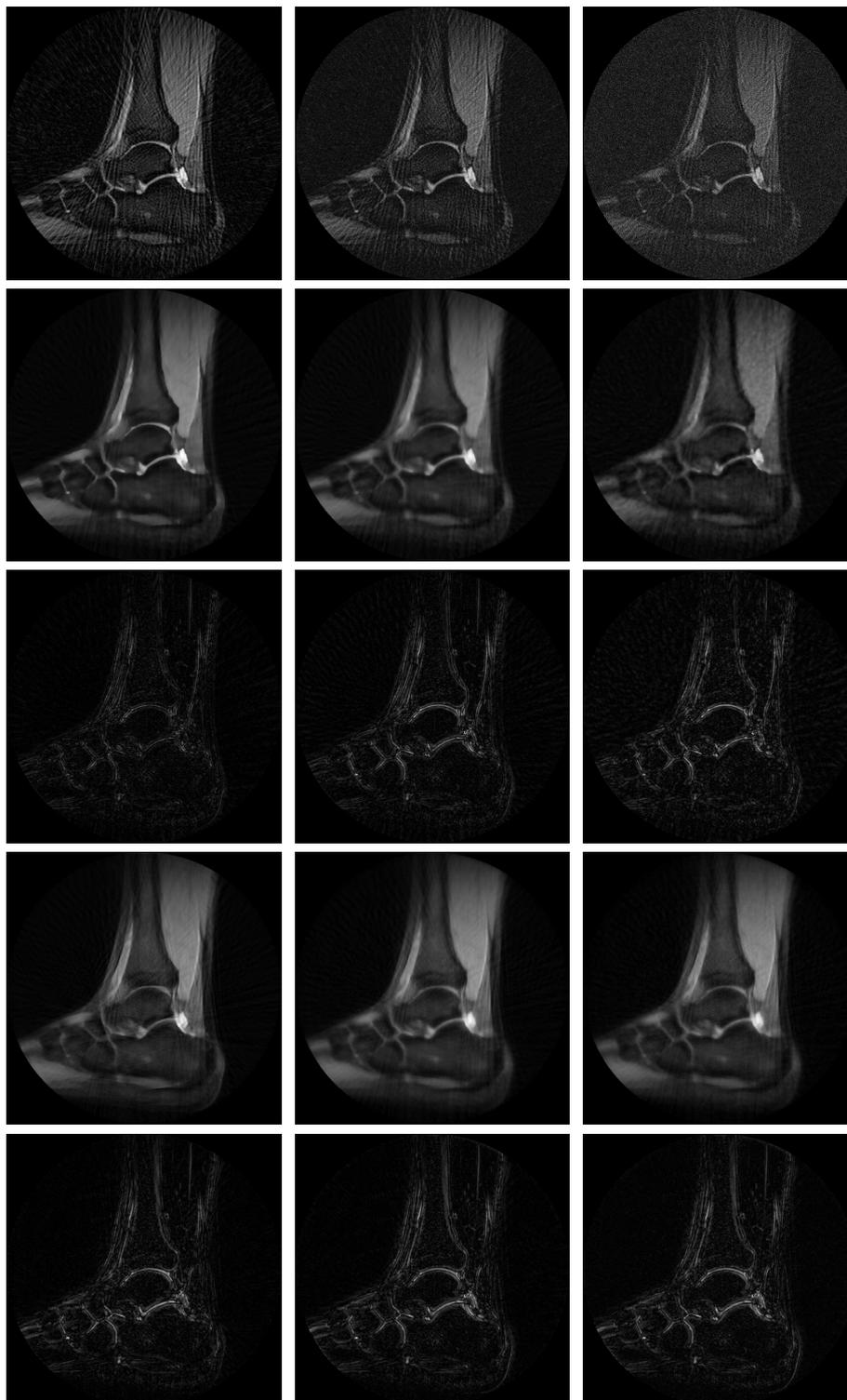


Figure 3.10: Ankle for $L = 8$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

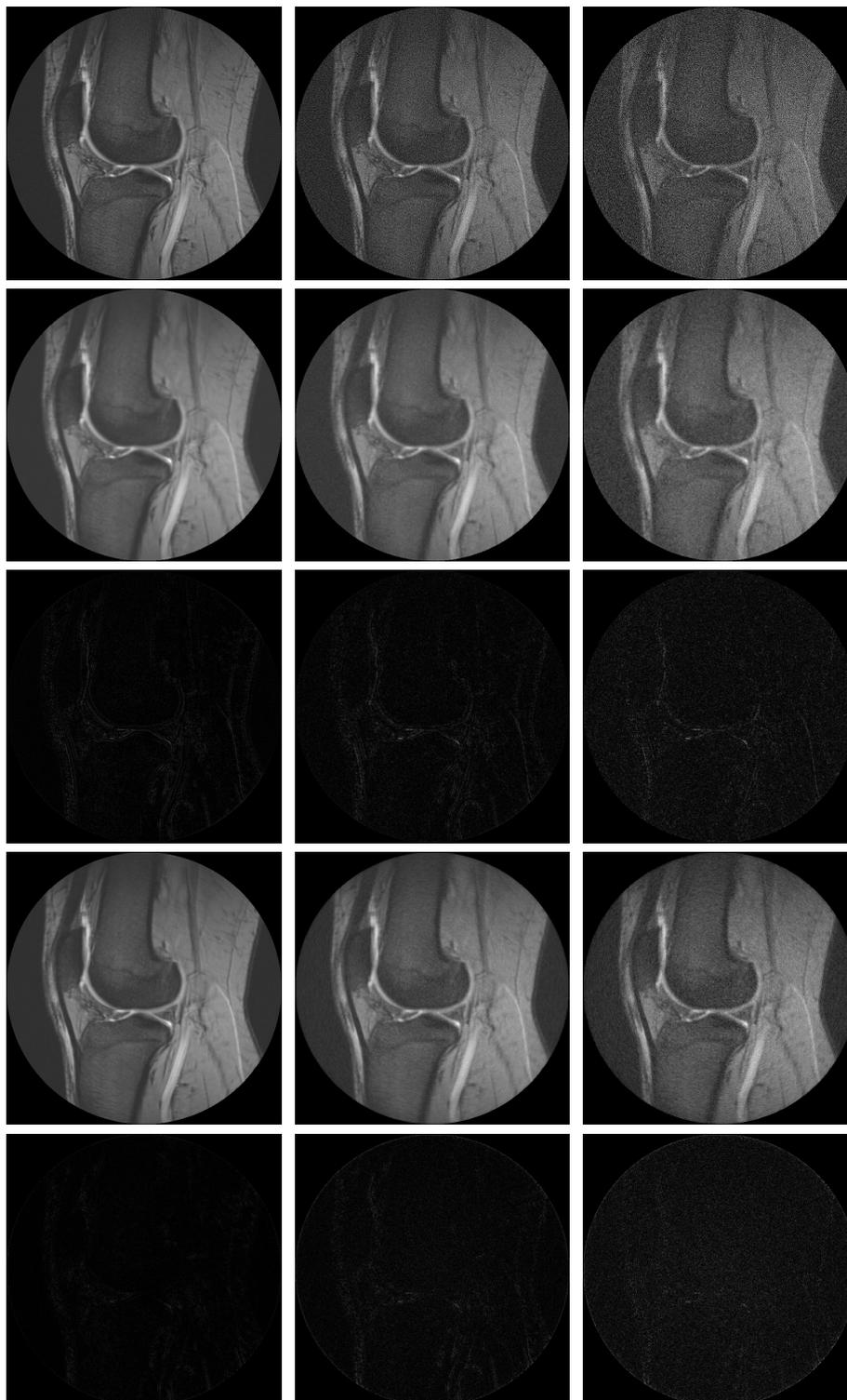


Figure 3.11: Knee for $L = 2$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

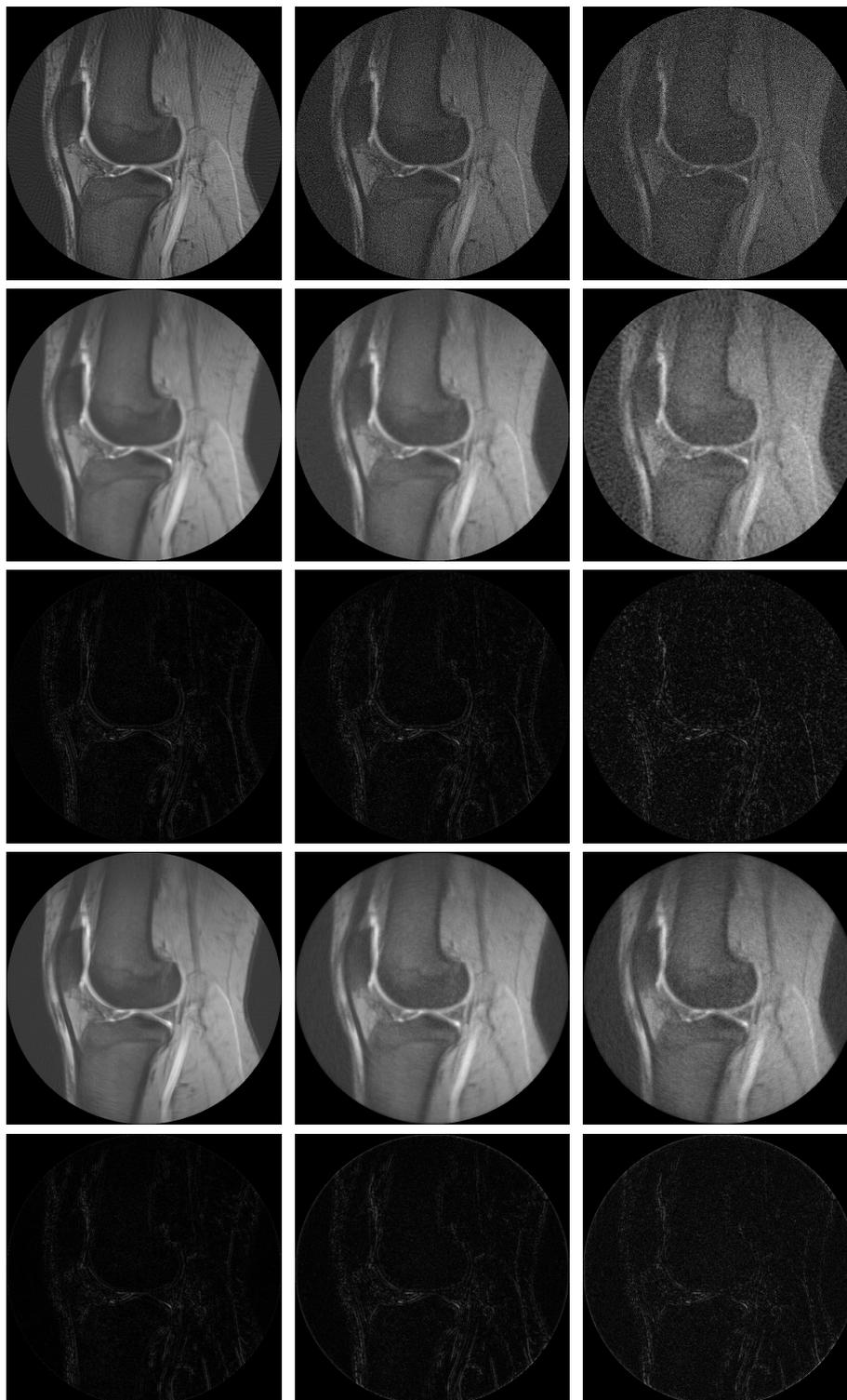


Figure 3.12: Knee for $L = 4$. From left-to-right: $SNR = +\infty, 50, 25$. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

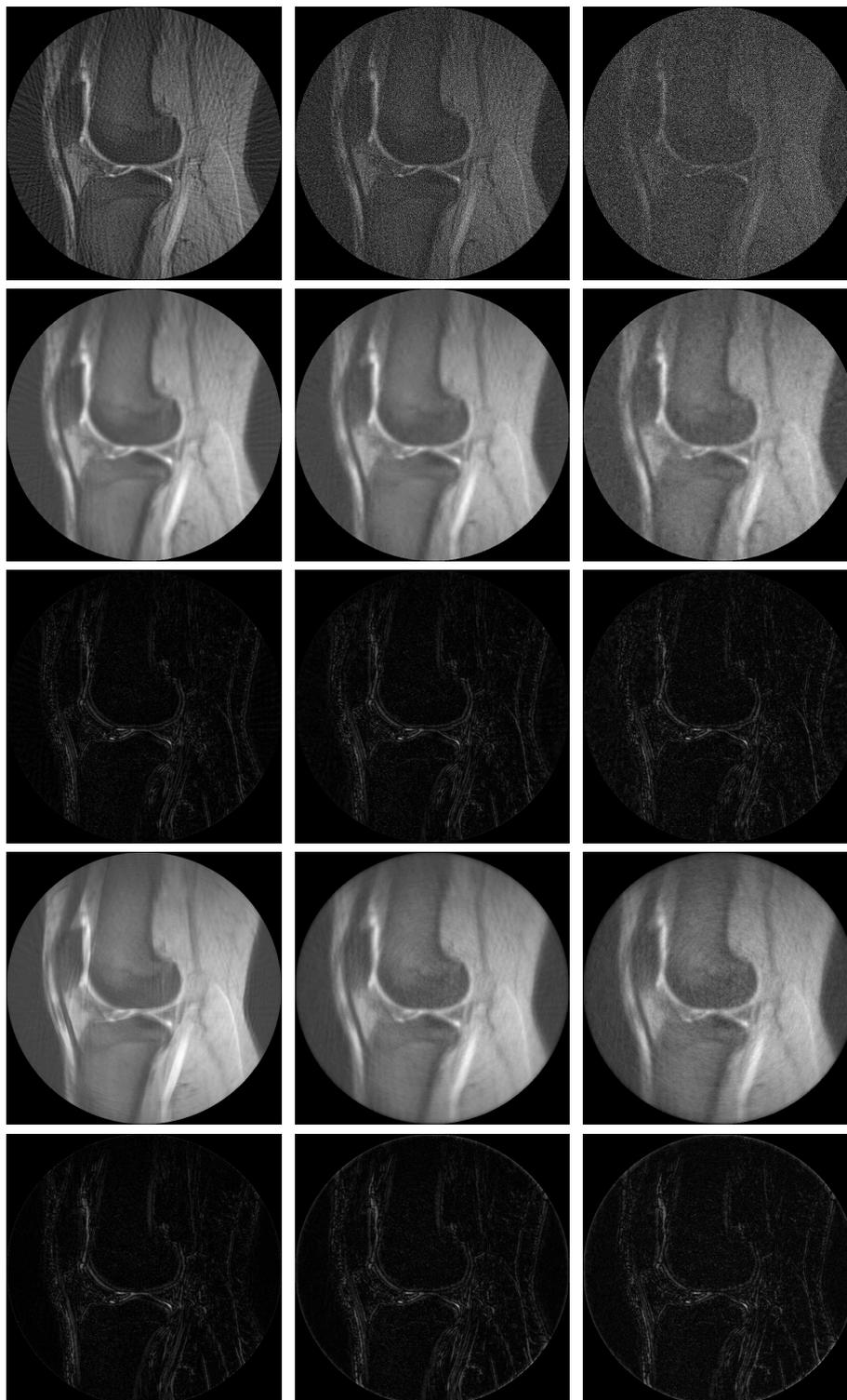


Figure 3.13: Knee for $L = 8$. From left-to-right: $SNR = +\infty, 50, 25$. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

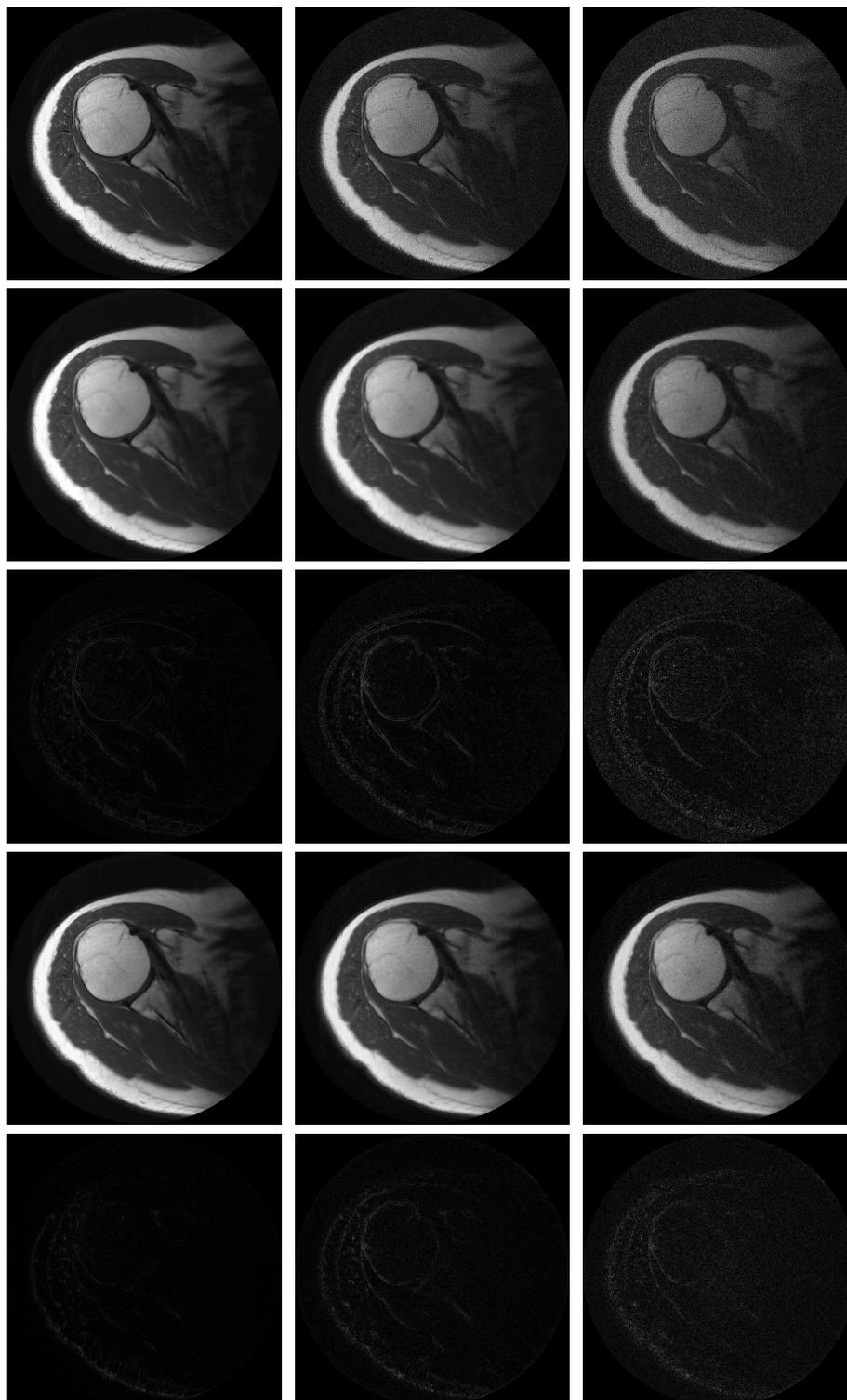


Figure 3.14: Shoulder for $L = 2$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

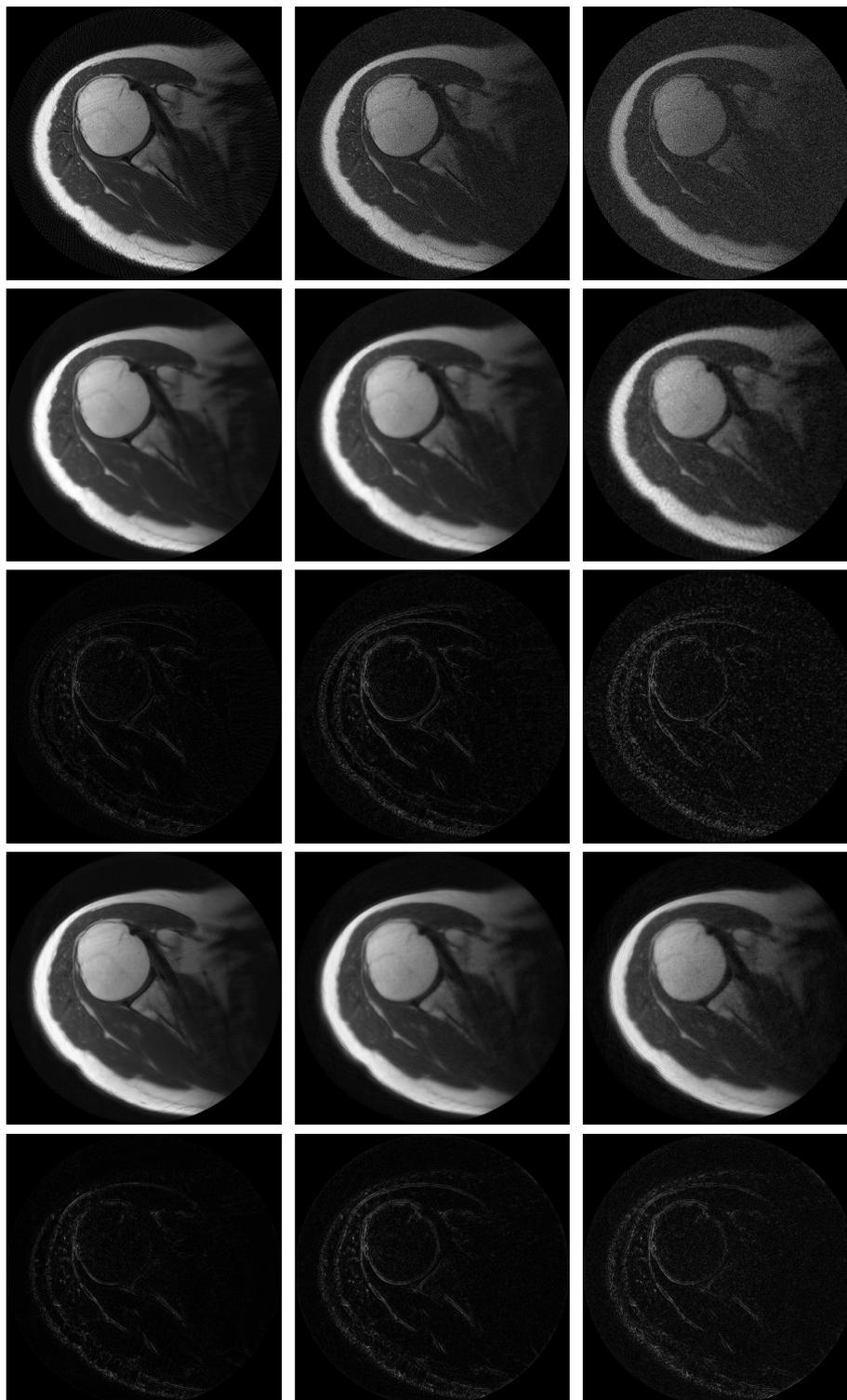


Figure 3.15: Shoulder for $L = 4$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

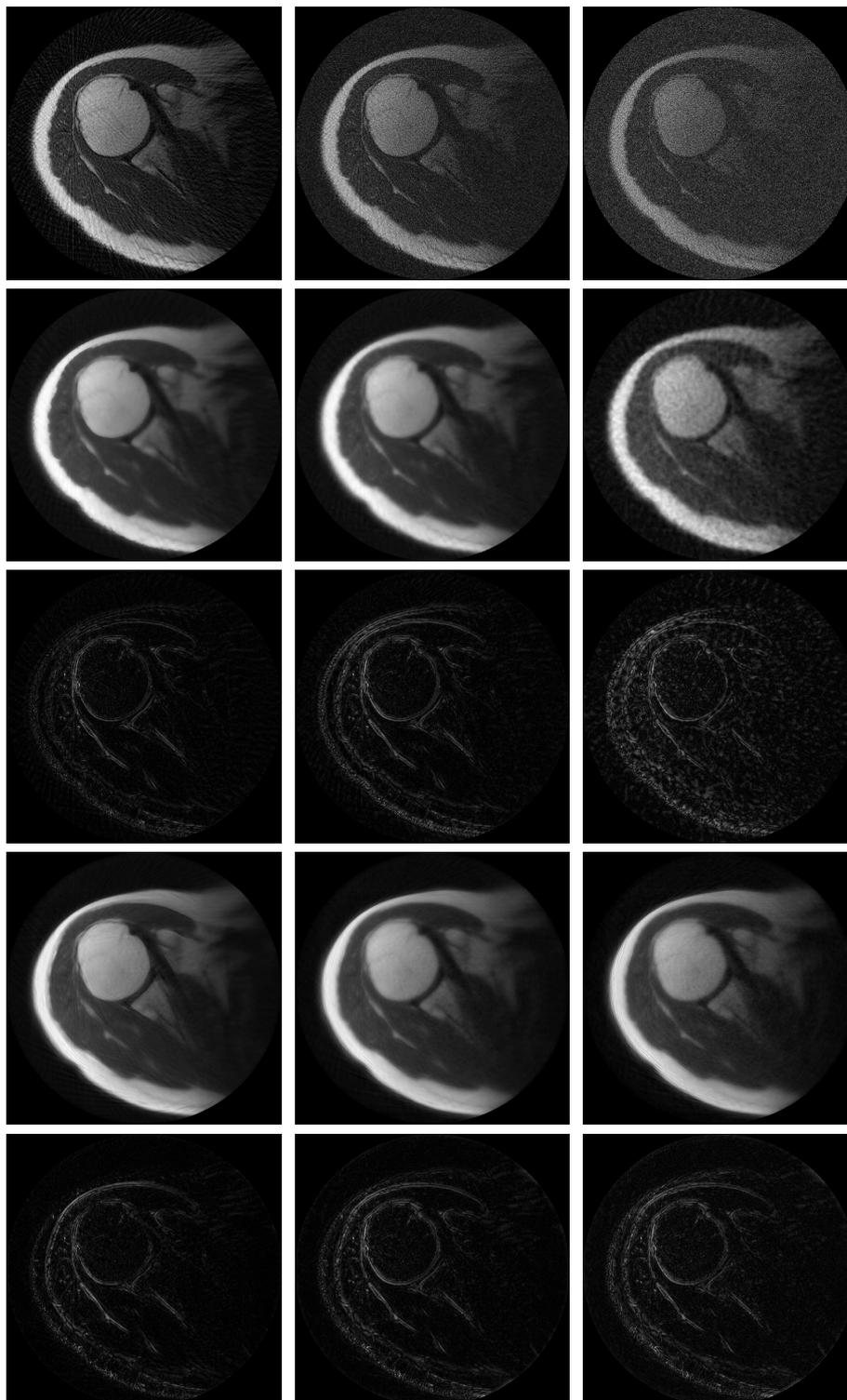


Figure 3.16: Shoulder for $L = 8$. From left-to-right: $SNR = +\infty$, 50, 25. From top-to-bottom: no reconstruction, NFFT, NFFT difference, ARCS, ARCS difference.

References

- [1] P.J. Beatty et al. “A Method for non-Cartesian Parallel Imaging Reconstruction with Improved Calibration”. In: *Proceedings of the 15th Annual Meeting of the ISMRM*. Berlin, Germany, May 2007, p. 335 (cit. on p. 57).
- [2] Amir Beck and Marc Teboulle. “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems”. In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202. DOI: [10.1137/080716542](https://doi.org/10.1137/080716542). URL: <http://link.aip.org/link/?SII/2/183/1> (cit. on p. 50).
- [3] Amir Beck and Marc Teboulle. “Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems”. In: *Image Processing, IEEE Transactions on* 18.11 (2009), pp. 2419–2434. ISSN: 1057-7149. DOI: [10.1109/TIP.2009.2028250](https://doi.org/10.1109/TIP.2009.2028250) (cit. on pp. 50, 51).
- [4] Felix Breuer et al. “High quality Real-Time Cardiac MRI using Self-Calibrating Radial GRAPPA with Sparsification”. In: *Proceedings of the 20th Annual Meeting of the ISMRM*. Melbourne, Victoria, Australia, May 2012, p. 4246 (cit. on p. 57).
- [5] Rachel W. Chan et al. “The influence of radial undersampling schemes on compressed sensing reconstruction in breast MRI”. In: *Magnetic Resonance in Medicine* 67.2 (2012), pp. 363–377. ISSN: 1522-2594. DOI: [10.1002/mrm.23008](https://doi.org/10.1002/mrm.23008). URL: <http://dx.doi.org/10.1002/mrm.23008> (cit. on p. 57).
- [6] Noel C.F. Codella et al. “A radial self-calibrated (RASCAL) generalized autocalibrating partially parallel acquisition (GRAPPA) method using weight interpolation”. In: *NMR in Biomedicine* 24.7 (2011), pp. 844–854. ISSN: 1099-1492. DOI: [10.1002/nbm.1630](https://doi.org/10.1002/nbm.1630). URL: <http://dx.doi.org/10.1002/nbm.1630> (cit. on p. 57).
- [7] M. A. Griswold, R. M. Heidemann, and P. M. Jakob. “Direct Parallel Imaging Reconstruction of Radially Sampled Data Using GRAPPA with Relative Shifts”. In: *Proceedings of the 11th Annual Meeting of the ISMRM*. Toronto, Ontario, Canada, July 2003, p. 2349 (cit. on pp. 44, 57).
- [8] Feng Huang et al. “Self-calibration method for radial GRAPPA/k-t GRAPPA”. In: *Magnetic Resonance in Medicine* 57.6 (2007), pp. 1075–1085. ISSN: 1522-2594. DOI: [10.1002/mrm.21233](https://doi.org/10.1002/mrm.21233). URL: <http://dx.doi.org/10.1002/mrm.21233> (cit. on p. 57).
- [9] Junzhou Huang, Shaoting Zhang, and Dimitris Metaxas. “Efficient MR Image Reconstruction for Compressed MR Imaging”. In: *Medical Image Analysis* (Apr. 2011). Submitted preprint (cit. on p. 49).
- [10] J.I. Jackson et al. “Selection of a convolution function for Fourier inversion using gridding [computerised tomography application]”. In: *Medical Imaging, IEEE Transactions on* 10.3 (Sept. 1991), pp. 473–478. ISSN: 0278-0062. DOI: [10.1109/42.97598](https://doi.org/10.1109/42.97598) (cit. on p. 44).
- [11] Cheng Guan Koay, Joelle E. Sarlls, and Evren Özarlan. “Three-dimensional analytical magnetic resonance imaging phantom in the Fourier domain”. In: *Magnetic Resonance in Medicine* 58.2 (2007), pp. 430–436. ISSN: 1522-2594. DOI: [10.1002/mrm.21292](https://doi.org/10.1002/mrm.21292). URL: <http://dx.doi.org/10.1002/mrm.21292> (cit. on p. 54).

- [12] Stefan Kunis. “Nonequispaced FFT Generalisation and Inversion”. PhD thesis. Institut für Mathematik, Universität zu Lübeck, June 2006 (cit. on pp. 44, 54).
- [13] Susanne Kunis and Stefan Kunis. “The nonequispaced FFT on graphics processing units”. In: *Proceedings in Applied Mathematics and Mechanics* (2012) (cit. on p. 57).
- [14] Michael Lustig, David Donoho, and John M. Pauly. “Sparse MRI: The application of compressed sensing for rapid MR imaging”. In: *Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195. ISSN: 1522-2594. DOI: [10.1002/mrm.21391](https://doi.org/10.1002/mrm.21391). URL: <http://dx.doi.org/10.1002/mrm.21391> (cit. on p. 44).
- [15] M. Lustig et al. “Compressed Sensing MRI”. In: *Signal Processing Magazine, IEEE* 25.2 (Mar. 2008), pp. 72–82. ISSN: 1053-5888. DOI: [10.1109/MSP.2007.914728](https://doi.org/10.1109/MSP.2007.914728) (cit. on p. 44).
- [16] Seunghoon Nam et al. “Compressed sensing reconstruction for whole-heart imaging with 3D radial trajectories: A graphics processing unit implementation”. In: *Magnetic Resonance in Medicine* (2012), n/a–n/a. ISSN: 1522–2594. DOI: [10.1002/mrm.24234](https://doi.org/10.1002/mrm.24234). URL: <http://dx.doi.org/10.1002/mrm.24234> (cit. on pp. 44, 57).
- [17] D.C. Noll, D.G. Nishimura, and A. Macovski. “Homodyne detection in magnetic resonance imaging”. In: *Medical Imaging, IEEE Transactions on* 10.2 (June 1991), pp. 154–163. ISSN: 0278-0062. DOI: [10.1109/42.79473](https://doi.org/10.1109/42.79473) (cit. on p. 57).
- [18] Claudia Prieto et al. “3D undersampled golden-radial phase encoding for DCE-MRA using inherently regularized iterative SENSE”. In: *Magnetic Resonance in Medicine* 64.2 (2010), pp. 514–526. ISSN: 1522-2594. DOI: [10.1002/mrm.22446](https://doi.org/10.1002/mrm.22446). URL: <http://dx.doi.org/10.1002/mrm.22446> (cit. on p. 57).
- [19] Klaas P. Pruessmann et al. “Advances in sensitivity encoding with arbitrary k-space trajectories”. In: *Magnetic Resonance in Medicine* 46.4 (2001), pp. 638–651. ISSN: 1522-2594. DOI: [10.1002/mrm.1241](https://doi.org/10.1002/mrm.1241). URL: <http://dx.doi.org/10.1002/mrm.1241> (cit. on p. 57).
- [20] Klaas P. Pruessmann et al. “SENSE: Sensitivity encoding for fast MRI”. In: *Magnetic Resonance in Medicine* 42.5 (1999), pp. 952–962. ISSN: 1522-2594. DOI: [10.1002/\(SICI\)1522-2594\(199911\)42:5<952::AID-MRM16>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1522-2594(199911)42:5<952::AID-MRM16>3.0.CO;2-S). URL: [http://dx.doi.org/10.1002/\(SICI\)1522-2594\(199911\)42:5%3C952::AID-MRM16%3E3.0.CO;2-S](http://dx.doi.org/10.1002/(SICI)1522-2594(199911)42:5%3C952::AID-MRM16%3E3.0.CO;2-S) (cit. on pp. 44, 57).
- [21] Peter R. Seevinck et al. “Highly localized positive contrast of small paramagnetic objects using 3D center-out radial sampling with off-resonance reception”. In: *Magnetic Resonance in Medicine* 65.1 (Aug. 2011), pp. 146–156. ISSN: 1522-2594. DOI: [10.1002/mrm.22594](https://doi.org/10.1002/mrm.22594). URL: <http://dx.doi.org/10.1002/mrm.22594> (cit. on p. 57).
- [22] Nicole Seiberlich et al. “Improved radial GRAPPA calibration for real-time free-breathing cardiac imaging”. In: *Magnetic Resonance in Medicine* 65.2 (2011), pp. 492–505. ISSN: 1522-2594. DOI: [10.1002/mrm.22618](https://doi.org/10.1002/mrm.22618). URL: <http://dx.doi.org/10.1002/mrm.22618> (cit. on p. 57).

- [23] Nicole Seiberlich et al. “Non-Cartesian data reconstruction using GRAPPA operator gridding (GROG)”. In: *Magnetic Resonance in Medicine* 58.6 (2007), pp. 1257–1265. ISSN: 1522-2594. DOI: [10.1002/mrm.21435](https://doi.org/10.1002/mrm.21435). URL: <http://dx.doi.org/10.1002/mrm.21435> (cit. on p. 57).
- [24] Nicole Seiberlich et al. “Reconstruction of undersampled non-Cartesian data sets using pseudo-Cartesian GRAPPA in conjunction with GROG”. In: *Magnetic Resonance in Medicine* 59.5 (2008), pp. 1127–1137. ISSN: 1522-2594. DOI: [10.1002/mrm.21602](https://doi.org/10.1002/mrm.21602). URL: <http://dx.doi.org/10.1002/mrm.21602> (cit. on p. 57).
- [25] Nicholas R. Zwart, Kenneth O. Johnson, and James G. Pipe. “Efficient sample density estimation by combining gridding and an optimized kernel”. In: *Magnetic Resonance in Medicine* 67.3 (2012), pp. 701–710. ISSN: 1522-2594. DOI: [10.1002/mrm.23041](https://doi.org/10.1002/mrm.23041). URL: <http://dx.doi.org/10.1002/mrm.23041> (cit. on p. 44).