Technical University of Denmark

OLIVER ACKERMANN LYLLOFF (S082312)

Efficient Nonnegative Least Squares Solvers for Beamforming Deconvolution

Master's Thesis, March 2014

Supervisors:
Finn T. Agerkvist, Associate Professor at Department of Electrical Engineering, DTU
Efren Fernandez-Grande, Assistant Professor at Department of Electrical Engineering, DTU
Elisabet Tiana Roig, Ph.D. student at Department of Electrical Engineering, DTU
Martin S. Andersen, Postdoc at Section for Scientific Computing, DTU Compute
Jørgen Hald, Research Engineer at Brüel & Kjær

DTU - Technical University of Denmark, Kgs. Lyngby - 2014

DTU Electrical Engineering
Department of Electrical Engineering

Brüel & Kjær

Efficient Nonnegative Least Squares Solvers for Beamforming Deconvolution

This report was prepared by:
Oliver Ackermann Lylloff (s082312)

Supervisors:
Finn T. Agerkvist, Associate Professor at Department of Electrical Engineering, DTU
Efren Fernandez-Grande, Assistant Professor at Department of Electrical Engineering, DTU
Elisabet Tiana Roig, Ph.D. student at Department of Electrical Engineering, DTU
Martin S. Andersen, Postdoc at Section for Scientific Computing, DTU Compute
Jørgen Hald, Research Engineer at Brüel & Kjær

DTU Electrical Engineering
Acoustic Technology
Technical University of Denmark
Elektrovej, Building 352
2800 Kgs. Lyngby
Denmark

studieadministration@elektro.dtu.dk

Project period: September 2013- March 2014

ECTS: 35

Education: MSc

Field: Electrical Engineering

Remarks: This report is submitted as partial fulfillment of the requirements for graduation in the above education at the Technical University of Denmark.

Abstract

Beamforming is a widely used signal processing technique that, when applied to measurements obtained with a microphone array, can map and thereby localize and rank sources of exterior aerodynamic noise. The main limitations are poor spatial resolution at low frequencies and a high level of sidelobes and ghost sources at high frequencies. To remedy these limitations and increase spatial resolution, deconvolution methods have been applied. These methods assume that the beamforming map can be approximated by a convolution of the true source distribution and the beamformer's point-spread function, that is, the beamformer's response to a point-source. This assumption is valid when the sources are mutually incoherent. Such sources can be described by power based models, and since the individual powers cannot be negative, the deconvolution problem can be formulated as a Nonnegative Least Squares (NNLS) problem. Typical problem sizes amount to several thousand unknowns and the same number of equations. If the coverage angle of the beamformer is not too wide, the point-spread function is to a good approximation shift-invariant and the convolution product can be calculated efficiently using the FFT algorithm.

This thesis presents an analysis of the deconvolution problem and propose efficient algorithms for the NNLS problem. Specifically it is found that the first-order gradient projection methods are the only practical choice for efficient convergence, and in particular, three such methods are chosen from litterature and tested on simulated and experimental data. One of these algorithms, commonly known as FFT-NNLS, have been shown to produce good results in the beamforming community, while the two others have not previously been applied in this field.

The results show that the deconvolution algorithms attain better convergence rates and higher spatial resolution when the point-spread function is predominantly shift-invariant. In most tests, one proposed algorithm, denoted fast gradient projection (FGP), is seen to be superior in terms of number of iterations and computational run time.

Finally, a new sequence method is proposed to calculate a range of frequency bands in series efficiently. The method is seen to improve the overall computational time and spatial resolution and is based on supplying the deconvolution algorithms with a qualified starting guess, also known as a warm-start.

Preface

This Master's Thesis is the concluding work of the MSc program in Engineering Acoustics at the Technical University of Denmark. The project has been supervised by Elisabet Tiana-Roig, Efren Fernandez Grande, and Finn Agerkvist at the Acoustic Technology department between September 2013 and March 2014.

The project was originally proposed by Jørgen Hald at Brüel & Kjær and Martin S. Andersen at Section for Scientific Computing, DTU Compute.

Oliver Ackermann Lylloff, February 28, 2014.

Acknowledgements

Many great people have been involved in this project and contributed with guidance and support. Initially, I would like to thank Efren Fernandez Grande for his dedication in the project, open door policy, and many hours of great discussions. I would also like to thank Finn Agerkvist and Elisabet Tiana-Roig for revising this document and providing valuable suggestions.

Also a thanks to Martin S. Andersen for his patience and many useful suggestions. To Jørgen Hald for great discussions and support, and Brüel & Kjær for supplying measurement equipment and data.

The technical support by Tom Petersen and Jørgen Rasmussen has been outstanding, for that, thank you.

Finally I must take the opportunity to thank my wife, Stella, for her love, friendship and unlimited support.

Abbreviations

Abbreviations	**Description**
1D, 2D, and 3D	one, two, and three dimensions
CG	Conjugate Gradient
DAS	Delay-and-sum
DR	Diagonal Removal
FFT	Fast Fourier Transform
FGP	Fast gradient projection
GPL	Gradient projection with exact linesearch
GPBB	Gradient projection with BB-steps
MSL	Maximum sidelobe level
PSF	Point-spread function
QP	Quadratic program
SNR	Signal-to-noise ratio
SVD	Singular value decomposition

List of Symbols

Symbol	Description
$\boldsymbol{\kappa}$	Unit vector
$(\cdot)^H$	Hermitian transpose
$(\cdot)^*$	Complex conjugate
$*$	Convolution
A_0	Amplitude
d_0	Area covered by beamformer
D	Microphone array aperture
$\mathcal{F}$	Fourier Transform
$\mathcal{F}^{-1}$	Inverse Fourier Transform
k	Wave number or iteration counter. Specified in text.
$\mathbf{k}_0$	Wave number vector
L	Lipschitz constant
M	Number of microphones
N	Number of grid points
$p_m(t)$	Sound pressure measured at m'th microphone
$P_m(\omega)$	Fourier transformed sound pressure measured at m'th microphone
$\mathbf{P}$	Vector of (short-time) Fourier transformed sound pressure for M microphones
$\mathbf{r}_\circ$	Coordinates to the center of microphone array
$\mathbf{r}_m$	Position (x, y, z) of m'th microphone
R	Beamformer resolution
$R_m(\mathbf{r})$	$\equiv \lvert\mathbf{r} - \mathbf{r}_m\rvert$ distance from a point in space to m'th microphone position
$R_\circ(\mathbf{r}_s)$	$\equiv \lvert\mathbf{r}_\circ - \mathbf{r}_s\rvert$ distance from the center of array to point source position
$f(\mathbf{x})_+$	Projected gradient
$\text{TOL}\nabla f(\mathbf{x})_+$	Stopping criterion of deconvolution algorithm $\lVert\nabla f(\mathbf{x}^k)_+\rVert / \lVert\nabla f(\mathbf{x}^0)_+\rVert \leq \epsilon$
Δ_m	Time or phase delay of m'th microphone in beamformer expression
Δx	Source separation
ω	Angular frequency
w_m	Microphone weights
ϕ	Beamformer opening angle
θ_{sep}	Angular separation of sources

Table of Contents

1 Introduction

Sound visualization is the field of 'seeing sound' where measurements of sound fields are mapped to give an acoustic image. Sound sources can then be located in space and their relative contributions to the overall sound field can be assessed. Methods such as near field acoustic holography (NAH) [1], beamforming [2], and acousto-optic tomography [3] provide images or maps of sound and are specialized to perform under different measurement scenarios. What is mutual for the methods is that they capture information about the sound field over a spatial extent to gain information about the origin of sound sources – much like the human auditory system that utilize two ears to pinpoint the location of a sound.

NAH is an established measurement technique from which calibrated acoustic source maps can be calculated. Measurements are performed with a collection of microphones arranged in a rectangular grid close to the source. The grid spacing must be less than half a wavelength at the highest frequency of interest and furthermore, the measurement grid must capture the major part of the sound radiation plus a 45° solid angle [2]. For sources of large extent and/or at high frequencies, the sampling density and coverage angle make NAH impractical. In such cases beamforming is an attractive alternative.

Beamforming is mainly used for mapping of exterior aerodynamic noise from large industrial applications, such as cars, airplanes, and wind turbines by combining measurements from multiple microphones, arranged in a so-called microphone array. The upper frequency limit, for certain optimized array geometries, is given when the average microphone spacing is 3-4 wavelengths [4]. In comparison to NAH, where the limit is half a wavelength, beamforming requires much fewer measurement positions to map a given area. The resolution of NAH is approximately half a wavelength at high frequencies and equal to the measurement distance at low frequencies [4]. The resolution of beamforming on the other hand is proportional to one wavelength at all frequencies. These resolution properties make NAH suitable for measurements at low frequencies of smaller objects and beamforming suitable for larger objects at high frequencies.

In order to remedy the resolution limitations of beamforming, deconvolution methods have been applied. The deconvolution problem seek to approximate the true source distribution based on the poorly resolved beamforming map, which constitute an inverse problem. In general the problem is ill-posed and iterative methods are often applied in order to gain meaningful reconstructions.

The aim of this thesis is to

- ▷ outline the properties of the deconvolution problem and identify the numerical difficulties of its solution,
- ▷ investigate a set of possible ways to enhance convergence and stability,
- ▷ propose efficient algorithms for the solution of the deconvolution problem,
- ▷ test and validate algorithms on simulated and experimental data.

1.1 Outline of thesis

This thesis is structured as follows: Chapter 2 introduces the delay-and-sum beamformer, describes the main limitations, and sets up a model for calculating the point-spread function. Chapter 3 describes the deconvolution problem and provides an introduction to optimization. The basic properties of efficient deconvolution algorithms are identified and three solution methods are proposed. The simulation study is presented in chapter 4 and experimental study in chapter 5. A summary and comparison of the test results are given in chapter 6 along with suggestions for future work.

1.2 Notation

Throughout this thesis, vectors are stated in bold face letters: $\mathbf{x}$ and matrices in capital letters: A. Lower case letters: x, are scalars unless otherwise stated. Commonly used symbols and abbreviations are listed above or otherwise stated in the text.

2 Beamforming

Beamforming is a signal processing technique applied to data captured simultaneously from multiple sensors. When the measurement objective is sound, microphones act as sensors and are typically arranged in a specific pattern called a microphone array. An ordinary microphone provides temporal information of the pressure fluctuations generated by the sound field in the immediate vicinity of the microphone. A single microphone however, cannot provide any spatial information about the location of a sound source. Simultaneous measurements with multiple microphones can provide spatio-temporal information about the location of sound sources due to the known distribution of microphones in the array. Beamforming provides the ability to focus towards a direction or a point, from where one signal among others is arriving, and thereby average out unwanted signals and noise. In other words, the array directivity is changed algorithmically rather than physically [5].

In the following section is delay-and-sum beamforming introduced for infinite and finite focusing distance and the convenient cross-spectrum formulation is derived.

2.1 Conventional beamforming

Delay-and-sum (DAS) beamforming is one of the most simple signal processing algorithms known and it provides an estimation of the relative amplitude and location of acoustical sources in a given area of interest. The algorithm makes use of the fact that wavefronts are received with different delays at the microphone positions according the relative distance between the individual microphones and the sound sources [2].

Consider a planar array of M microphones located in the xy-plane and a plane wave incident from an unknown direction, described by the unit-vector $\boldsymbol{\kappa}$. The sound pressure p_m at the m'th microphone caused by the incident plane wave with angular frequency ω_0 and wave number vector $\mathbf{k_0}$, is given by

$$p(\mathbf{r_m}, t) \equiv p_m(t) = A_0 e^{j(\omega_0 t - \mathbf{k_0} \cdot \mathbf{r_m})}, \tag{2.1}$$

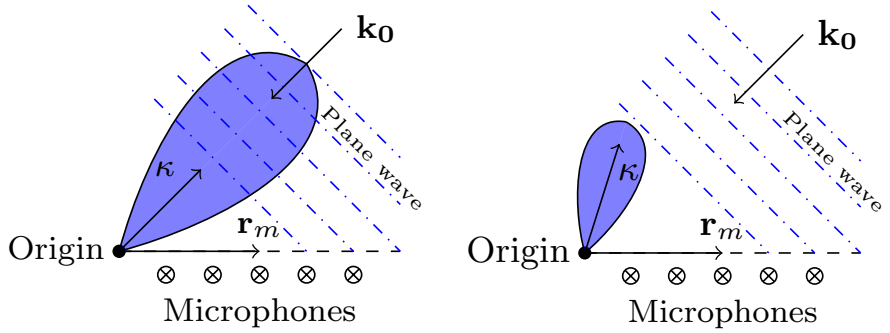
where $\mathbf{r_m}$ denotes the position of the m'th microphone [5]. The DAS beamformer output is given by,

$$b(\boldsymbol{\kappa}, t) = \frac{1}{M} \sum_{m=1}^{M} w_m p_m(t - \Delta_m(\boldsymbol{\kappa})), \tag{2.2}$$

where w_m are weighting coefficients and individual time-delays,

$$\Delta_m(\boldsymbol{\kappa}) = \frac{\boldsymbol{\kappa} \cdot \mathbf{r_m}}{c}, \tag{2.3}$$

are applied in order to steer the DAS beamformer in a chosen direction [5]. It is hence the choice of time delays $\Delta_m(\boldsymbol{\kappa})$ that steers the beamformer in the direction $\boldsymbol{\kappa}$ and thereby changes the array directivity. The amplitude weights w_m can be applied to enhance the beamformer's resolution or reduce sidelobe levels [5]. Uniform shading $w_m = 1$ will exclusively be considered in the following and w_m will therefore be omitted.



(a) Maximum DAS beamformer output when $\mathbf{k}_0 = -k_0\boldsymbol{\kappa}$.

(b) Weaker DAS beamformer output when $\mathbf{k}_0 \neq -k_0\boldsymbol{\kappa}$.

Figure 2.1: Far-field beamforming (infinite focus distance).

Figure 2.1 illustrate the output of the DAS beamformer at two different focusing directions. The directional response of the DAS beamformer is represented by a broad lobe. When the focusing direction $\mathbf{k}_0 = -k_0\boldsymbol{\kappa}$, the main lobe attains a maximum response, represented by a large lobe. If the focusing direction is different from the incident wave direction, $\mathbf{k}_0 \neq -k_0\boldsymbol{\kappa}$, the output is weaker, represented by a smaller lobe.

Signals with broadband frequency content can be decomposed into a superposition of monochromatic (single-frequency) plane waves and thereby treated independently by frequency. This is essentially done by applying the Fourier Transform to the output of each microphone – the result is the frequency domain version of the DAS beamformer (2.2),

$$B(\boldsymbol{\kappa}, \omega) = \frac{1}{M} \sum_{m=1}^{M} P_m(\omega) e^{-j\omega\Delta_m(\boldsymbol{\kappa})}, \tag{2.4}$$

where an implicit time factor $e^{j\omega t}$ is suppressed and $P_m(\omega)$ is the Fourier Transform of the m'th microphone output [5].

The beamformer output is steered in direction $\boldsymbol{\kappa}$ by applying a phase shift instead of a time-delay. The frequency domain version will exclusively be used in the following.

Example 2.1 Consider a uniform linear array of $M = 9$ microphones located on the positive x-axis with uniform spacing $d = 0.08\,\text{m}$, applied for beamforming in the frequency domain using Equation (2.4). A unit-amplitude plane wave, incident from angle $\theta_0 = 60°$, has wave number vector $\mathbf{k_0}$ and frequency $f_0 = 2000\,\text{Hz}$. The individual time delays in Equation (2.3) are now given by a steering angle $\theta = [0; 180]°$ due to the simple geometry,

$$\begin{aligned} \boldsymbol{\kappa} &= (\cos(\theta), \sin(\theta), 0)^T, \\ \mathbf{r}_m &= m \cdot d \cdot (1, 0, 0)^T, \\ \Delta_m(\theta) &= \frac{md\cos(\theta)}{c}. \end{aligned} \tag{2.5}$$

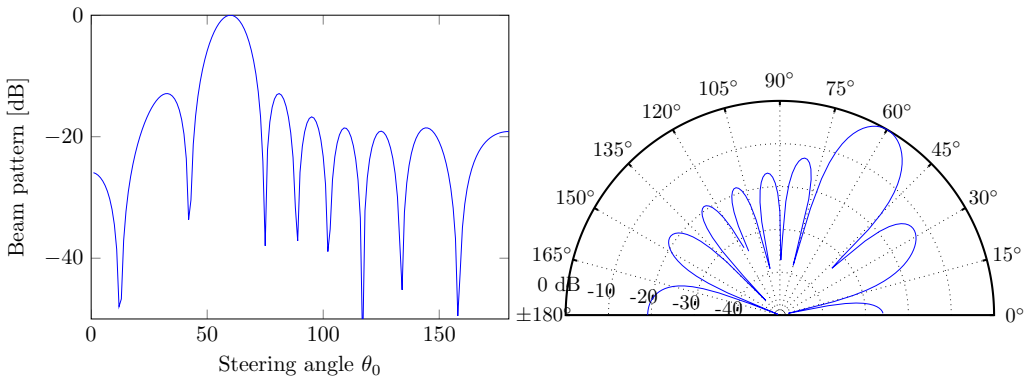
The measured sound pressure is given by $P_m(\omega_0) = e^{-j\mathbf{k}_0 \cdot \mathbf{r}_m}$ such that the beamformer output (2.4) can be written,

$$B(\boldsymbol{\kappa}, \omega_0) = \frac{1}{M} \sum_{m=1}^{M} e^{-j\mathbf{k}_0 \cdot \mathbf{r}_m} e^{-j\omega_0\Delta_m(\boldsymbol{\kappa})} = \frac{1}{M} \sum_{m=1}^{M} e^{j(\mathbf{k}-\mathbf{k}_0)\cdot \mathbf{r}_m}, \tag{2.6}$$

where $\mathbf{k} = -\frac{\omega_0}{c}\cos(\theta)\boldsymbol{\kappa}$. The beamformer output for $\omega_0 = 2\pi f_0$ is

$$B(\theta, \omega_0) = \frac{1}{M}\sum_{m=1}^{M} e^{j\omega_0 md(\cos(\theta_0)-\cos(\theta))/c}. \tag{2.7}$$

The magnitude of Equation (2.7) as function of angle is shown in Figure 2.2. A broad main lobe is pointing in the direction of the incident plane wave. A number of sidelobes are also present, all with magnitudes less than 10 dB below the magnitude of the main lobe.



(a) (b)

Figure 2.2: A conventional plot (a) and a polar plot (b) show a main lobe in the direction of the incident wave and several sidelobes of lower level. From Example 2.1.

The expression (2.6) can be rewritten as

$$\begin{aligned} B(\boldsymbol{\kappa}, \omega) &= \frac{1}{M}\sum_{m=1}^{M} e^{j(\mathbf{k}-\mathbf{k}_0)\cdot\mathbf{r}_m} \equiv \frac{1}{M}\sum_{m=1}^{M} W(\mathbf{k}-\mathbf{k}_0) \\ W(\mathbf{K}) &= \sum_{m=1}^{M} e^{j\mathbf{K}\cdot\mathbf{r}_m}, \end{aligned} \tag{2.8}$$

where $W(\mathbf{K})$ is denoted the Array Pattern. $W(\mathbf{K})$ contains information about the directivity pattern of the array for all directions and frequencies, and depends only on the array geometry (for uniform shading $w_m = 1, m = 1, \ldots, M$) [2].

It is evident that the frequency-domain beamformer output in Equation (2.6) asserts its maximum when $\mathbf{k} = \mathbf{k}_0$, i.e. when the focusing direction coincides with the plane wave direction of propagation which forms the main lobe. However, additional lobes are also present in the directivity pattern – see e.g. Figure 2.2. These sidelobes are a result of the measured level of incident plane waves arriving from directions not equal to the actual focus direction. The maximum sidelobe level (MSL) defines the beamformer's ability to suppress sidelobe levels and is measured as the sidelobe level relative to the main lobe level. A low MSL can be achieved by specific array designs, such as arrays with circular and irregular geometries and by applying amplitude weights w_m [2].

Grating lobes is another well-known phenomenon which exhibits repetitions of the main lobe – they typically occur when uniform arrays, such as the linear array described in Example 2.1, are applied for beamforming above a certain frequency. The upper frequency limit is bounded by spatial aliasing due to the Nyquist sampling theorem: Plane waves with frequency above $f_{max} = \frac{c}{2d}$ cannot be reconstructed unambiguously from the spatial sampling with microphones. Here c denotes the speed of sound and d the microphone

spacing [5]. The effect is that plane waves incident from an angle θ will also contribute at other aliased angles. This aliasing effect is seen in Example 2.2 where broadband noise excitation is considered.

Example 2.2 Consider a linear array as described in Example 2.1, now with incident plane waves of many different frequencies, i.e. broadband noise. The frequency dependent directivity pattern of the beamformer is shown in Figure 2.3. The main lobe is still present in the direction of the incident plane waves, however at higher frequencies (above $f_{max} = 2144\,\text{Hz}$) grating lobes occur. The width of the main lobe increases at low frequencies, resulting in poor resolution. The resolution increases at higher frequencies along with the addition of grating lobes, making it impossible to determine the correct direction of the incident plane waves.

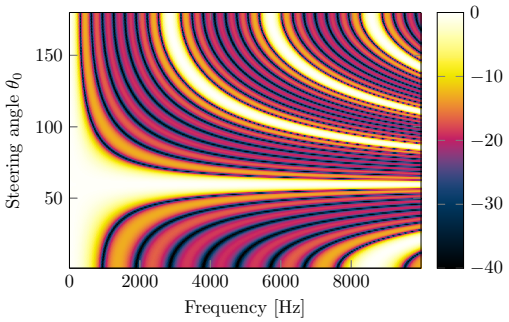


Figure 2.3: Linear array applied for beamforming with broadband noise excitation incident from $\theta = 60°$, $f_{max} = 2144\,\text{Hz}$. From Example 2.2.

Example 2.2 captures the main limitations of the frequency-domain beamformer: Poor resolution at low frequencies and high level of sidelobes and grating lobes at high frequencies. The resolution of the beamformer's output refers to its ability to distinguish waves incident from slightly different directions. By considering two incident plane waves with wave number vectors, $\mathbf{k}_1$ and $\mathbf{k}_2$, and the same wavenumber $|\mathbf{k}_1| = |\mathbf{k}_2| = k$, the beamformer's output can be written as a superposition of the two in terms of the Array Pattern function $W(\mathbf{K})$,

$$B(\boldsymbol{\kappa}, \omega) = W(\mathbf{k} - \mathbf{k}_1) + W(\mathbf{k} - \mathbf{k}_2). \tag{2.9}$$

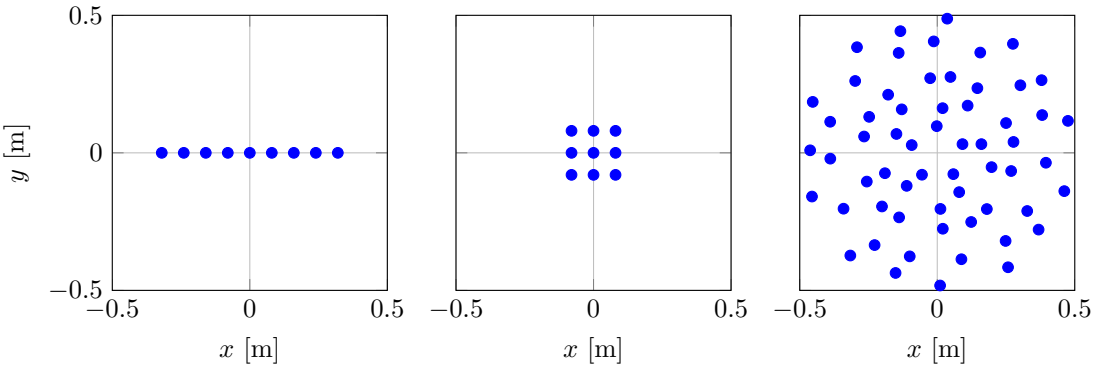
Rayleigh's criterion then states that the two waves can just be resolved when the peak of $W(\mathbf{k} - \mathbf{k}_2)$ falls on the first zero of $W(\mathbf{k} - \mathbf{k}_1)$ [5]. Following this, it can be shown [2] that the resolution of the frequency-domain beamformer is given by

$$R(\theta) = \frac{az\lambda}{\cos^3(\theta)D} = \frac{az_0c}{\cos^3(\theta)fD}, \tag{2.10}$$

where $a = 1$ for linear apertures and $a \approx 1.22$ for circular, planar apertures, z_0 is the measurement distance, D is the array aperture, λ is the wavelength, and θ is the angular separation of the incoming waves. The resolution is inversely proportional to the frequency, as it was seen in Example 2.2, the size of the aperture D, and the angular separation θ. It is directly proportional to the measurement distance z_0, which will become relevant when finite focus distance beamforming is introduced in the next section.

By now it is clear that the array design has a significant impact on the output of the frequency-domain beamformer - specifically in terms of resolution and MSL. Three different array designs are shown in Figure 2.4. The uniform linear array design (Figure 2.4a) was used in Example 2.1 and 2.2.

Example 2.3 Consider a grid array of 3×3 microphones with uniform spacing $d = 0.08\,\text{m}$ applied for frequency-domain beamforming with incident plane waves as described in Example 2.1. The



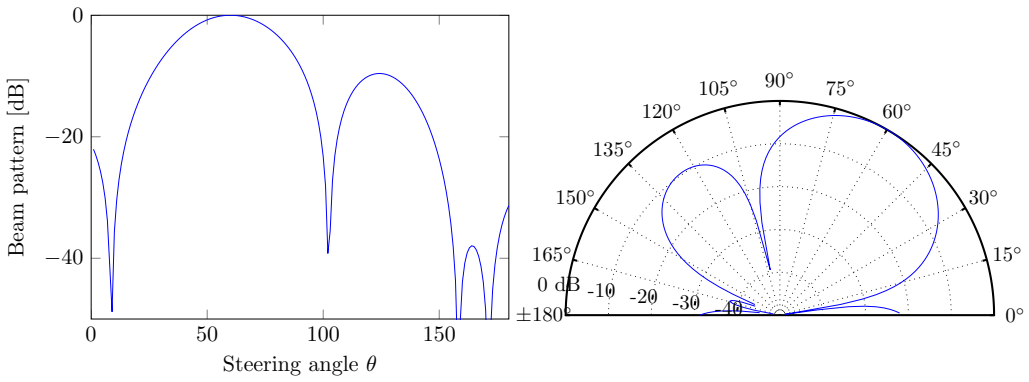
(a) 9 ch. uniform linear array (b) 3 × 3 ch. grid array (c) 60 ch. irregular array (B&K P60 D100)

Figure 2.4: Examples of array designs.

directional response (Fig. 2.5) is clearly of poorer resolution than the response obtained with the linear array (Fig. 2.2). The resolution ratio, given by (2.10), is approximately

$$\frac{R_{\text{grid}}}{R_{\text{lin}}} = \frac{D_{\text{lin}}}{D_{\text{grid}}} \approx \frac{9 \cdot 0.08\,\text{m}}{3 \cdot 0.08\,\text{m}} = 3. \tag{2.11}$$

That is, the spatial resolution of the linear array is three times better than that of the grid array. The result is a wider mainlobe for the grid array than for the line array which would make it more difficult to distinguish two incident plane waves propagating in two slightly different directions towards the array.



(a) (b)

Figure 2.5: A conventional plot (a) and a polar plot (b) showing a 3×3 grid array applied for frequency-domain beamforming with incident plane waves from $\theta = 60°$. The frequency is $f = 2000\,\text{Hz}$. From Example 2.3.

2.1.1 Finite focus distance

The examples considered so far have all dealt with 1D directional responses of the frequency-domain beamformer and specifically incoming plane waves corresponding to point sources located towards infinity. In real-world applications one is often concerned with the 2D or 3D response which focus on points at finite distance, taking spherical wave radiation into account. The 2D case is introduced in the following.

The sound pressure at $\mathbf{r}$ generated by a point source at $\mathbf{r}_s$ is the solution to the inhomogeneous Helmholtz equation

$$\nabla^2 p(\mathbf{r}) + k^2 p(\mathbf{r}) = -j\omega\rho Q\delta(\mathbf{r}_s - \mathbf{r})e^{j\omega t}, \tag{2.12}$$

where Q is the volume velocity, ρ is the density, and $\delta(\mathbf{r}_s - \mathbf{r})$ is the Dirac-delta function [6]. By setting $j\omega\rho Q = 1$, the equation simplifies to

$$\left(\nabla^2 + k^2\right) G(\mathbf{r}|\mathbf{r}_s) = -\delta(\mathbf{r}_s - \mathbf{r}), \tag{2.13}$$

where $G(\mathbf{r}|\mathbf{r}_s)$ is the Green's function. In free-space $G(\mathbf{r}|\mathbf{r}_s)$ is given by

$$G(\mathbf{r}|\mathbf{r}_s) = \frac{1}{R}e^{-jkR}, \tag{2.14}$$

where $R \equiv |\mathbf{r}_s - \mathbf{r}|$ [6].

The sound pressure obtained from the m'th microphone due to a monopole point source at $\mathbf{r}_s$ is thus given by

$$P_m(\omega) = G(\mathbf{r}_m|\mathbf{r}_s)q(\mathbf{r}_s) = \frac{q(\mathbf{r}_s)}{R}e^{-jkR}, \tag{2.15}$$

where $q(\mathbf{r}_s) \equiv j\omega\rho Q$ is the spherical wave amplitude of the point source at $\mathbf{r}_s$. This expression is only of use when considering simulated sound fields where the exact location of a point source is known.

The beamforming algorithm (2.4) is altered to account for spherical wave radiation by changing the delays in equation (2.3) to

$$\Delta_m(\mathbf{r}) = \frac{|\mathbf{r}| - R_m(\mathbf{r})}{c} \tag{2.16}$$

and the frequency-domain beamformer output for finite focus distance is given by [2]

$$B(\mathbf{r}, \omega) = \frac{1}{M}\sum_{m=1}^{M} P_m(\omega)e^{-j\omega\Delta_m(\mathbf{r})}, \tag{2.17}$$

where $\mathbf{r}$ is the focus point of the beamformer and $R_m(\mathbf{r}) \equiv |\mathbf{r} - \mathbf{r}_m|$ denotes the distance from the m'th microphone to the focus point $\mathbf{r}$ – see Figure 2.6.

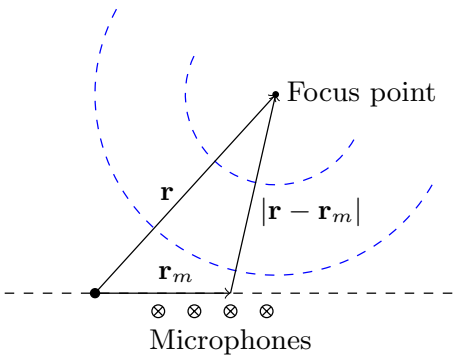
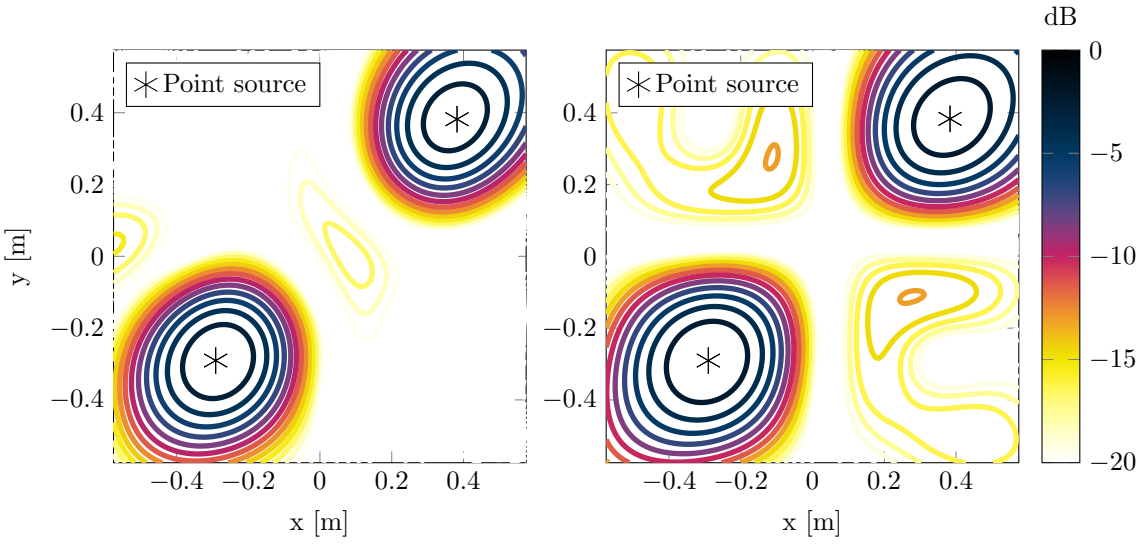


Figure 2.6: Near-field beamforming (finite focus distance). Spherical waves emitted by a monopole source at the focus point.

Example 2.4 Consider an 8×8 ch. grid array and a 60 ch. irregular array (Fig. 2.4b and 2.4c respectively) applied for frequency-domain beamforming. Two simulated point sources are located at $(x, y, z)^T = (0.38, 0.38, 1)^T$ and $(-0.29, -0.29, 1)^T$ with frequency $f = 1500$ Hz. The grid array has microphone spacing $d = 0.1$ m and array aperture $D_{\text{grid}} = 0.99$ m, the irregular grid has pseudo random microphone spacing and array aperture $D_{\text{irr}} = 0.99$ m. The magnitude of the beamformer's output is shown in Figure 2.7.



(a) Irregular array

(b) Grid array

Figure 2.7: 2D directional response of frequency-domain beamformer with grid array and irregular array.

The irregular array facilitate a lower MSL and better resolution than the regular grid array. This is in agreement with the resolution expression (2.10) and analysis of the Array Pattern $W(\mathbf{K})$ discussed above.

2.1.2 Cross-spectrum formulation

For stationary sound fields, the mean-square (or average power) output of the frequency-domain beamformer for finite focus distance, is given by

$$\overline{|B(\boldsymbol{\kappa},\omega)|^2} = \frac{1}{M^2}\sum_{m,n=1}^{M} \overline{P_m(\omega)P_n^*(\omega)}e^{j\omega(\Delta_n(\mathbf{r})-\Delta_m(\mathbf{r}))}, \tag{2.18}$$

where $C_{nm} \equiv \overline{P_m(\omega)P_n^*(\omega)}$ is the cross-spectrum matrix and $(\cdot)^*$ denotes the complex conjugate [2]. The cross-spectrum matrix can be calculated by taking short-time Fourier transforms of the measured sound pressures p_m across the array and forming the vector $\mathbf{P}$ from these [5]

$$\mathbf{P} \equiv \begin{bmatrix} P_1(\omega) \\ P_2(\omega) \\ \vdots \\ P_M(\omega) \end{bmatrix}. \tag{2.19}$$

Then $C \equiv \mathbf{P}\mathbf{P}^H$ is the $M \times M$ cross-spectrum matrix at angular frequency ω. It is the outer product of the short-time Fourier Transform vector $\mathbf{P}$ with itself. $(\cdot)^H$ denotes the Hermitian transpose. In practical applications, the temporal sound pressures $p_m(t)$ are measured simultaneous with M microphones. Since the measurements are of finite duration, the short-time Fourier transform is applied to obtain the corresponding sound pressures $P_m(\omega)$ in the frequency domain. The details of the spectral averaging procedures are described later on.

C has a Toeplitz structure (constant values along its diagonals) and is conjugate symmetric $C_{mn} = C_{nm}^*$

(Hermitian),

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} & \cdots & C_{1M} \\ C_{12}^* & C_{22} & C_{12} & \cdots & C_{1(M-1)} \\ C_{13}^* & C_{12}^* & C_{33} & \cdots & C_{1(M-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{1M}^* & C_{1(M-1)}^* & C_{1(M-2)}^* & \cdots & C_{MM} \end{bmatrix}. \tag{2.20}$$

The diagonals $C_{mm} = \overline{P_m(\omega)P_m^*(\omega)}$ represents autospectra without phase information and thus only contain self-noise, such as wind-noise or electronic noise from the measurement hardware [2]. The off-diagonal elements $C_{m\neq n}$, represent cross-spectra with information of the relative phase between each microphone pair. Any self-noise present in the cross-spectrum will be averaged out since the self-noise in one channel is assumed to be incoherent with other channels. It can be shown that the exclusion of the autospectra – also known as Diagonal Removal (DR) – decreases the maximum sidelobe level [2]. The resulting average power output of the frequency-domain beamformer becomes

$$\begin{aligned} \overline{|B(\mathbf{r},\omega)|^2} &= \frac{1}{M^2}\left[\sum_{m=1}^{M} C_{mm} + \sum_{m\neq n}^{M} C_{nm} e^{j\omega(\Delta_n(\mathbf{r})-\Delta_m(\mathbf{r}))}\right], \\ \overline{|B(\mathbf{r},\omega)|^2} &= \frac{1}{M(M-1)}\sum_{m\neq n}^{M} C_{nm} e^{j\omega(\Delta_n(\mathbf{r})-\Delta_m(\mathbf{r}))}, \end{aligned} \tag{2.21}$$

where diagonal removal is applied to the latter. One side effect of using DR is that low-level noise regions might attain negative values which must be kept in mind when interpreting beamformer maps [7].

The Green's function in (2.15) is a complex-valued matrix $G \in \mathbb{C}^{M\times S}$, where each column $\mathbf{g} \in \mathbb{C}^M$ represents the transfer function between a point source at $\mathbf{r}_s$ and all M microphones. S is the number of point sources. This vector $\mathbf{g}$ is commonly known as a steering vector, since it holds the phase shifts of each microphone necessary to steer the beamformer towards a given point source. This point source steering vector is given by

$$\mathbf{g}(\mathbf{r}_s) \equiv R_\circ(\mathbf{r}_s)\begin{bmatrix} e^{-jkR_1(\mathbf{r}_s)}/R_1(\mathbf{r}_s) \\ e^{-jkR_2(\mathbf{r}_s)}/R_2(\mathbf{r}_s) \\ \vdots \\ e^{-jkR_M(\mathbf{r}_s)}/R_M(\mathbf{r}_s) \end{bmatrix}, \tag{2.22}$$

where $R_\circ(\mathbf{r}_s) \equiv |\mathbf{r}_\circ - \mathbf{r}_s|$ is the distance from the center of the array to $\mathbf{r}_s$, used to normalize the distance related decay of the signals [8]. For simulation purposes, the Fourier transformed sound pressure output $P_m(\omega)$, generated at the m'th microphone, by a single point source, is given by

$$P_m(\omega) = \mathbf{g}(\mathbf{r}_s)q(\mathbf{r}_s). \tag{2.23}$$

The sound pressure generated at all microphones for all possible sources is given by

$$\mathbf{P} = G(\mathbf{r_m}|\mathbf{r}_s)\mathbf{q}(\mathbf{r}_s), \quad \text{for } m = 1\ldots M,\ s = 1,\ldots,S, \tag{2.24}$$

where S is the number of sources.

A model of the cross-spectrum matrix C_{mod} given S incoherent monopole point sources can be written as

$$C_{\text{mod}} = \mathbf{P}\mathbf{P}^H = \sum_{s=1}^{S} |q(\mathbf{r}_s)|^2 \cdot \mathbf{g}(\mathbf{r}_s)\mathbf{g}(\mathbf{r}_s)^H, \tag{2.25}$$

where $|q(\mathbf{r}_s)|^2$ is a power descriptor of the s'th source. That is, the sources contribute additively to power descriptors, which per definition are nonnegative.

The average power output of the frequency-domain beamformer can be written, equivalent to $\mathbf{g}(\mathbf{r}_s)$, in terms of a steering vector $\mathbf{e}$, given by

$$\begin{aligned} \overline{|B(\mathbf{r},\omega)|^2} &= \frac{1}{M^2}\mathbf{e}^H\mathbf{P}\mathbf{P}^H\mathbf{e} = \frac{1}{M^2}\mathbf{e}^H C\mathbf{e}, \\ \overline{|B(\mathbf{r},\omega)|^2} &= \frac{1}{M(M-1)}\mathbf{e}^H C_0\mathbf{e}, \end{aligned} \tag{2.26}$$

where diagonal removal is applied to the latter and $C_0 \equiv C_{mm} = 0$ for $m = 1, \ldots, M$. The steering vector $\mathbf{e}$ is defined as

$$\mathbf{e} \equiv \frac{1}{R_\circ(\mathbf{r})} \begin{bmatrix} e^{-jkR_1(\mathbf{r})} \cdot R_1(\mathbf{r}) \\ e^{-jkR_2(\mathbf{r})} \cdot R_2(\mathbf{r}) \\ \vdots \\ e^{-jkR_M(\mathbf{r})} \cdot R_M(\mathbf{r}) \end{bmatrix}, \tag{2.27}$$

where $R_\circ(\mathbf{r})$ and $R_m(\mathbf{r})$ are used to normalize the distance related decay [8].

The cross-spectral formulation provides a mathematical model that is compact and easy to implement. The strength of this formulation will become apparent in the next chapter where the framework for the deconvolution problem is derived.

3 Deconvolution Methods

Deconvolution methods are widely used in many fields of imaging to improve spatial resolution. Research in fields such as Astronomy [9], Microscopy [10], Signal Processing [11], and Image Deblurring [12] has suggested various methods for improving image quality. Some of these methods have been adapted to the visualization of acoustical sources, such as CLEAN [13] and Richardson-Lucy [8], while other have been specifically developed for acoustical purposes, such as DAMAS [7]. As previously stated, the inherent limitations of beamforming are poor resolution at low frequencies and high level of sidelobes and ghost sources at high frequencies. Deconvolution methods have been shown to improve these effects by reducing the sidelobe levels and increasing resolution [8]. These methods make use of the fact that the beamformer's map can be written as a convolution of the acoustic source distribution and the point-spread function, which is the beamformer's response to a point source.

In the following sections, the point-spread function is defined and the deconvolution problem stated. Furthermore, an introduction to optimization is given and requirements on efficient algorithms for the deconvolution problem are discussed. Finally, three efficient deconvolution algorithms are proposed.

3.1 The point-spread function

Consider the average power output of the frequency-domain beamformer generated by a unit strength point source located at $\mathbf{r}_s$. Insertion of the model cross-spectrum matrix (2.25) into Equation (2.26) of the average power beamformer output gives [8]

$$\text{PSF}(\mathbf{r}|\mathbf{r}_s) \equiv \frac{1}{M^2}\mathbf{e}^H \left[\mathbf{g}(\mathbf{r}_s)\mathbf{g}(\mathbf{r}_s)^H\right] \mathbf{e} = \frac{1}{M^2}|\mathbf{e}^H\mathbf{g}(\mathbf{r}_s)|^2. \tag{3.1}$$

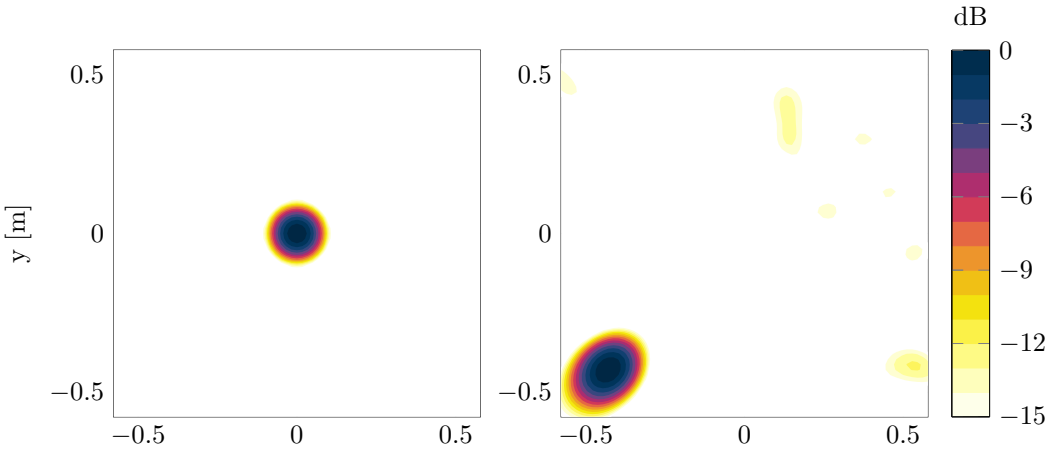
This expression describes the power transfer function between a unit point source at $\mathbf{r}_s$ and the focus point $\mathbf{r}$ and is known as the point-spread function $\text{PSF}(\mathbf{r}|\mathbf{r}_s)$. With the assumption that the sources are mutually incoherent, they contribute additively to the power descriptor at $\mathbf{r}$, such that an arbitrary number of sources S can be mapped by,

$$\overline{|B(\mathbf{r},\omega)|^2} = \sum_{s=1}^{S} |q(\mathbf{r}_s)|^2 \cdot \text{PSF}(\mathbf{r}|\mathbf{r}_s). \tag{3.2}$$

The shape of the point-spread function for each source is determined by the beamformer's directivity pattern. If the shape of the directivity pattern does not change between the sources, then it is said to be shift-invariant and thus depends only on the distance between the source and microphone array $\text{PSF}(\mathbf{r} - \mathbf{r}_s)$. This property has great practical implications since Equation (3.2) can now be written as

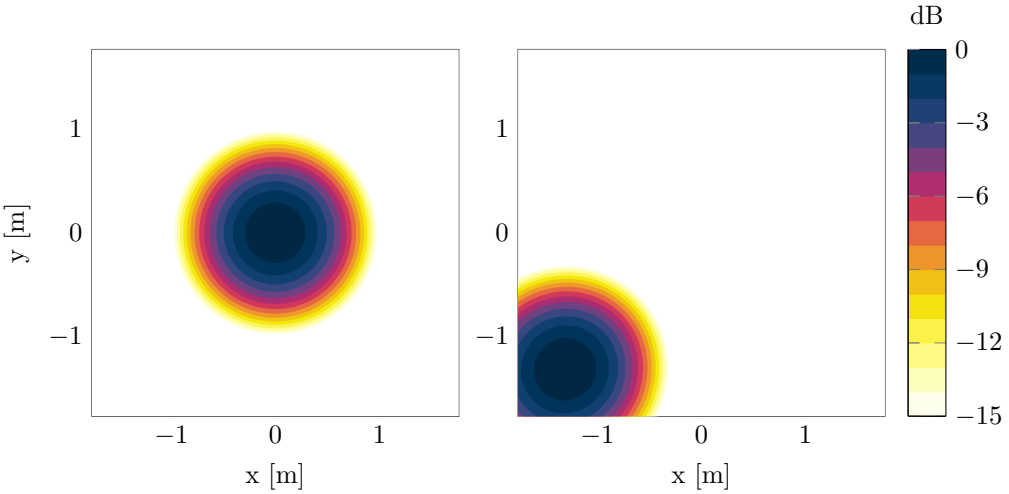
$$\overline{|B(\mathbf{r},\omega)|^2} = \sum_{s=1}^{S} |q(\mathbf{r}_s)|^2 \cdot \text{PSF}(\mathbf{r} - \mathbf{r}_s), \tag{3.3}$$

which corresponds to a linear convolution of the source power distribution and the shift-invariant point-spread function.



(a) Centered point source, large opening angle. $z_0 = 1$ m.

(b) Off-centered point source, large opening angle. $z_0 = 1$ m.



(c) Centered point source, small opening angle. $z_0 = 11$ m.

(d) Off-centered point source, small opening angle. $z_0 = 11$ m.

Figure 3.1: Examples of point-spread functions $\mathrm{PSF}(\mathbf{r}|\mathbf{r}_s)$ from centered and off-centered point sources. Source distance $z_0 = 1$ m and opening angle $\phi = 30°$ in the top panel, and $z_0 = 11$ m, $\phi = 10°$ in the bottom panel. The frequency is 3500 Hz.

Two point-spread functions are shown in Figure 3.1 to illustrate the shift-invariance property. One is the response at distance $z_0 = 1$ m and the other at $z_0 = 11$ m. The shape of the point-spread function in the upper panel changes when the position of the point source is shifted, it is hence shift-variant. The shape of the point-spread function in the bottom panel however is not altered much by the shift, it is approximately shift-invariant.

Example 3.1 One quantitative measure of shift-invariance could be the ratio of elliptical eccentricity of a point-spread function confined to the corner relative to one at the center. The mathematical measure of the elongation of a circle is given by its eccentricity

$$e = \sqrt{1 - (b/a)^2},$$

where a and b are one half of the major and minor axes of the ellipse. If $e = 0$, the ellipse is a circle and $e = 1$ defines a line.

The eccentricities of the ellipse defined by the -3 dB level contour line in Figure 3.1 are 0.7 in the upper right plot and 0.3 in the bottom right. The eccentricities of the centered point-spread functions are 0. Eccentricities of point-spread functions moving from the corner position along the diagonal to the center position are given in Table 3.1.

ϕ	z_0	Eccentricity						
30°	1	0.71	0.64	0.55	0.44	0.31	0.17	0.03
10°	11	0.28	0.21	0.21	0.15	0.11	0.06	0.006

Table 3.1: Eccentricities of point-spread functions gradually positioned from the corner to the center of the image. The eccentricity of a circle is zero and one for a line. $f = 3500$ Hz.

The resolution of the beamformer is given by Equation (2.10). The relative resolution is approximately 15 times greater for the point source at $z_0 = 1$ m, which agrees quite well with the visual difference in their widths. This example shows the trade-off between resolution and shift-invariance of the point-spread function: Small opening angles are required in order to assume a shift-invariant point-spread function, this will however limit the spatial extend of the source plane in sight and the array must be moved further away if the coverage area is too small.

In practice, to gain efficiency, the point-spread function is assumed to be shift-invariant. This means that only one point-spread function of a centered point source is used to describe all possible point-spread functions in the computational grid. If the assumption of shift-invariance is good, the mismatch between the representative and true point-spread functions is small. This is referred to as the model–data mismatch.

3.2 Problem formulation

Equation (3.2) describes how the true source distribution is *blurred* by the point-spread function. The point-spread function thus describes how a delta function is spread by the imaging system (microphone array). Consider a set of discrete points (x, y) in a grid of $N \times N$ elements with a focus plane located z_0 from the observation plane. An unknown source distribution of point sources is measured and the average power output $\overline{|B(\mathbf{r}, \omega)|}^2$ is calculated for each point in the grid. The objective is to retrace the true source distribution from the blurred beamformer power output. The process is often denoted deconvolution when the point-spread function is shift-invariant, since a reversal of the convolution product between the true source distribution and the point-spread function hopefully will deblur the mean-square beamformer output and reveal the true source distribution.

By computing the point-spread functions for each grid point and arranging them column wise in a matrix A of dimensions $N^2 \times N^2$, and arranging the unknown source distribution and beamformer power output as vectors, the relation between the source power descriptors $\mathbf{x} \equiv |q(\mathbf{r}_s)|^2$ and the average power output of the beamformer $\mathbf{b}$ is given by

$$A\mathbf{x} = \mathbf{b}, \tag{3.4}$$

where the underlying assumption is that the blurring operation can be described by a linear mapping. The deconvolution problem is thus to obtain $\mathbf{x}$ when A and $\mathbf{b}$ are known. This constitutes a discrete inverse problem [14].

When the point-spread function is shift-invariant, then A is the two-dimensional convolution operator which has a block Toeplitz structure, and each block is itself a Toeplitz matrix, this is also known as Block Toeplitz

with Toeplitz blocks (BTTB). This structure can be exploited to obtain faster deconvolution methods [15]. However, it will become apparent that explicit formulation of A is omitted due to its often large dimensions. When the point-spread function is shift-variant, no assumptions can be made about the structure of A.

The discrete inverse problem (3.4) is ill-posed due to the fact that the matrix A is ill-conditioned. This severely complicates the deconvolution process since the the naive solution $\mathbf{x} = A^{-1}\mathbf{b}$ will be buried in inverted noise from measurements or numerical rounding errors from simulations.

One way of obtaining an approximate solution to (3.4) involves solving an optimization problem, where the task is to find an $\mathbf{x}$ that minimizes the 2-norm of the residual squared $\|A\mathbf{x} - \mathbf{b}\|\|^2$. This basic idea is known in data-fitting as regression. The optimization problem is given, making use of the fact that the source powers $\mathbf{x} = |q(\mathbf{r}_s)|^2$ cannot attain negative values, by

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2, \\ &\text{subject to} \quad x_i \geq 0. \end{aligned} \tag{3.5}$$

The main objective is henceforth to identify algorithms that can solve this problem efficiently.

3.2.1 The Singular Value Decomposition

The singular value decomposition (SVD) is an essential and powerful tool for the analysis of linear inverse problems. For any square matrix $A \in \mathbb{R}^{N\times N}$, the SVD takes the form

$$A = U\Sigma V^T = \sum_{i=1}^{N} \mathbf{u}_i \sigma_i \mathbf{v}_i^T. \tag{3.6}$$

Here Σ is a square, diagonal matrix with singular values

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0 \tag{3.7}$$

and U and V are unitary matrices that consist of the left and right singular vectors $\mathbf{u}_i$ and $\mathbf{v}_i$. The inverse of A exist, if A is non-singular, and is given by

$$A^{-1} = V\Sigma^{-1}U^T \tag{3.8}$$

or alternatively

$$A^{-1}\mathbf{u}_i = \sigma_i^{-1}\mathbf{v}_i, \tag{3.9}$$

which leads to the so-called naive solution to (3.4),

$$\mathbf{x} = A^{-1}\mathbf{b} = \sum_{i=1}^{N} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i. \tag{3.10}$$

The reason, being dubbed naive, can be seen by basic perturbation theory: The expected accuracy can be calculated by considering the exact and perturbed solutions $\mathbf{x}^{\text{exact}}$ and $\mathbf{x}$, given by

$$A\mathbf{x}^{\text{exact}} = \mathbf{b}^{\text{exact}}, \quad A\mathbf{x} = \mathbf{b} = \mathbf{b}^{\text{exact}} + \mathbf{e}, \tag{3.11}$$

where $\mathbf{e}$ denotes the perturbation. The upper bound on the relative error is given by

$$\frac{\|\mathbf{x}^{\text{exact}} - \mathbf{x}\|_2}{\|\mathbf{x}^{\text{exact}}\|_2} \leq \text{cond}(A)\frac{\|\mathbf{e}\|_2}{\|\mathbf{b}^{\text{exact}}\|_2}, \tag{3.12}$$

where the 2-norm condition number of A is given by

$$\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2 = \sigma_1/\sigma_N. \tag{3.13}$$

This result shows that a large condition number implies that there is a risk of obtaining a solution $\mathbf{x}$ that is very far from the exact. Such a matrix is said to be ill-conditioned contrary to a well-conditioned matrix with a small condition number. An ill-conditioned matrix A will have small singular values σ_i such that any small perturbation $\mathbf{b} = \mathbf{b}^{\text{exact}} + \mathbf{e}$ will be amplified in the naive solution (3.10)

$$\mathbf{x} = A^{-1}\mathbf{b} = A^{-1}\left(\mathbf{b}^{\text{exact}} + \mathbf{e}\right) = \sum_{i=1}^{N} \frac{\mathbf{u}_i^T\left(\mathbf{b}^{\text{exact}} + \mathbf{e}\right)}{\sigma_i}\mathbf{v}_i. \tag{3.14}$$

This will in general lead to a noisy and useless solution since small σ_i will amplify errors in $\mathbf{b} = \mathbf{b}^{\text{exact}} + \mathbf{e}$. Alternative solution methods are necessary to obtain more meaningful solutions.

Unfortunately, no one solution method exists in general for these types of discrete inverse problems. However, a wide range of tools and methods have been developed, e.g. within the field on image deblurring, and are being used in problems arising in many different fields of research. The task is to understand the underlying model of the problem and then pick a solution method that supports that model and furthermore meets certain criteria, such as accuracy, robustness, and performance.

3.3 Optimization

Optimization is the minimization or maximization of a function subject to constraints on its variables. The general form of an optimization problem is given by [16]

$$\begin{aligned} \text{maximize or minimize} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq b_i, \quad i = 1, \ldots, m, \end{aligned} \tag{3.15}$$

where

- ▷ $\mathbf{x}$ is the optimization variable and the unknown of the problem,
- ▷ f_0 is the objective function, a quantitative measure that can be represented by a single number, of the performance of the system that must be minimized or maximized,
- ▷ f_i are the constraint functions, and
- ▷ b_i are constants that hold the bounds for the constraints.

The identification of objective, variable, and constraints for a specific problem is collectively known as modelling. A wide range of algorithms have been developed for various type-specific problems and choosing the optimal one is highly dependent on the modelling of the optimization problem.

The optimization problem considered in the following is the nonnegative least-squares (NNLS) problem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2 \\ \text{subject to} \quad & \mathbf{x} \succeq 0, \end{aligned} \tag{3.16}$$

where the objective function is the 2-norm of the squared residual and $\mathbf{x}$ is the variable subject to nonnegativity constraints[1] $\mathbf{x} \succeq 0$. In this case $\succeq$ simply states component wise inequalities $x_i \geq 0$, or put in other words, $\mathbf{x}$ is confined to the real nonnegative half-space *the nonnegative orthant* $\mathbb{R}_+^n$,

$$\mathbb{R}_+^n = \{\mathbf{x} \in \mathbb{R}^n | x_i \geq 0, i = 1, \ldots, n\} = \{x \in \mathbb{R}^n | x \succeq 0\} \tag{3.17}$$

where $\mathbb{R}_+$ denotes the set of nonnegative numbers: $\mathbb{R}_+ = \{x \in \mathbb{R} | x \geq 0\}$.

An optimization problem is called a quadratic program (QP) if the objective function is quadratic and can be expressed in standard form [17]

$$\begin{aligned} \text{maximize or minimize} \quad & f_0(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T P\mathbf{x} + \mathbf{c}^T\mathbf{x} + d \\ \text{subject to} \quad & G\mathbf{x} \preceq \mathbf{h} \\ & A\mathbf{x} = \mathbf{b}, \end{aligned} \tag{3.18}$$

where the gradient $\nabla f_0(\mathbf{x})$ and Hessian $\nabla^2 f_0(\mathbf{x})$ of $f_0(\mathbf{x})$ is given by

$$\nabla f_0(\mathbf{x}) = P\mathbf{x} + c, \quad \nabla^2 f_0(\mathbf{x}) = P.$$

The optimization problem in (3.16) is a QP with:

$$\begin{aligned} f(\mathbf{x}) &= \tfrac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2 = \tfrac{1}{2}\mathbf{x}^T A^T A\mathbf{x} - \mathbf{b}^T A\mathbf{x} + \tfrac{1}{2}\mathbf{b}^T\mathbf{b} \\ P &= A^T A, \quad \mathbf{c} = -\mathbf{b}^T A, \quad d = \tfrac{1}{2}\mathbf{b}^T\mathbf{b} \\ G &= -I, \quad h = \mathbf{0}. \end{aligned} \tag{3.19}$$

[1] Here the symbol $\succeq$ denotes a generalized inequality. When used in conjunction with matrices it implies positive semidefiniteness, e.g $A \succeq 0$. This follows the nomenclature in optimization litterature.

Furthermore, if A is real and non-singular, then $P = \nabla^2 f(\mathbf{x}) = A^T A$ is symmetric and positive semidefinite, i.e $A^T A \succeq 0$.

The optimization problem (3.16) belongs to the special class of *convex* optimization problems, for which a global solution can be found efficiently in many cases [17].

3.3.1 Convex optimization

Optimization problems are said to be convex if the functions f_i in (3.15) satisfy[2]

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \quad \alpha + \beta = 1, \alpha \geq 0, \beta \geq 0, \tag{3.20}$$

that is, the line connecting any two points on f must lie above the graph of f – see Figure 3.2. This leads to the first-order condition for convexity: Suppose f is differentiable, then f is convex if and only if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x). \tag{3.21}$$

In other words, the first order Taylor approximation of f near x gives a lower bound on the convex function globally [16]. This means that local information about a convex function (its value and derivative at a point) also provides information about the location of the global minimizer of f. Figure 3.2 illustrates this graphically: The first-order Taylor approximation of f near x, shown as the dashed straight line in Figure 3.2, lie below the graph of $f(y)$. The general optimality condition for a convex function f states [16]:

- x^* is a global minimizer of f, if and only if it is feasible (i.e. it satisfies the constraints of the optimization problem) and $\nabla f(x^*) = 0$.

If $\nabla f(x) = 0$ in (3.21) then $f(y) \geq f(x)$ and x is a global minimizer of f. If f is twice differentiable, then f is convex if and only if

$$\nabla^2 f(x) \succeq 0, \tag{3.22}$$

that is, the Hessian of $f(x)$ must be positive semidefinite [16]. In one dimension this is equivalent to the condition that the second derivative $f''(x)$ always be non-negative (have nonnegative curvature). Equation (3.22) constitutes the second-order condition for convexity.

If f is convex, and continuously differentiable, and has a Lipschitz continuous gradient with Lipschitz constant $L > 0$, then f is said to be smooth convex [18],

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

This can also be seen by expanding the first and second order conditions for convexity, Equation (3.21) and (3.22), respectively [18],

$$\begin{aligned} \nabla^2 f(x) &\preceq LI, \\ f(y) &\approx f(x) + \nabla f(x)^T (y - x) + \tfrac{1}{2}\nabla^2 f(x)\|y - x\|_2^2, \\ f(y) &\leq f(x) + \nabla f(x)^T (y - x) + \tfrac{L}{2}\|y - x\|_2^2. \end{aligned} \tag{3.23}$$

In other words: The largest eigenvalue of the Hessian is uniformly upper bounded by L. This translates to a quadratic upper bound on f, which is shown in Figure 3.2 as the quadratic dashed line that lies above $f(y)$.

[2] Note that the vector notation is omitted in this subsection in order to simplify the graphical representation. The theory is however fully valid in multiple dimensions.

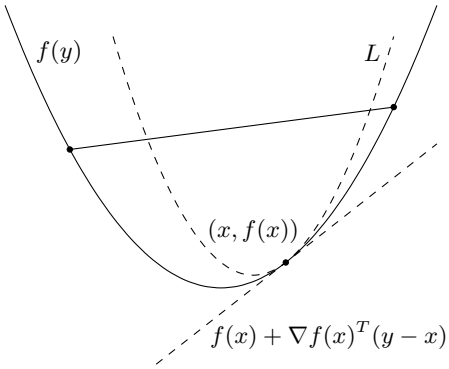


Figure 3.2: f is convex if the line connecting any two points on f lies above the graph of f. If f is convex and differentiable then $f(y) \geq f(x) + \nabla f(x)^T(y - x)$. If $\nabla f(x^*) = 0$ then x^* is a global minimizer of f. If f has a Lipschitz continuous gradient, there exists a quadratic upper bound designated by the Lipschitz constant L .

f is said to be strongly convex if there exists a constant μ such that $\mu I \preceq \nabla^2 f(x)$ [18]. The Hessian is then uniformly lower bounded by μ, the so-called strong convexity parameter. If the smallest eigenvalue of the Hessian is zero, e.g. if A is rank-deficient, then $\mu = 0$ and f is not strongly convex.

The properties of convex optimization outlined here will prove very useful in the following analysis of deconvolution algorithms.

3.4 Algorithms

Optimization algorithms solve a given problem, such as NNLS (3.16), iteratively. An initial guess is supplied by the user and a sequence of improved estimates is generated by the algorithm until a predetermined stopping criterion is met. The strategy used to step from one iterate to another is what distinguishes optimization algorithms. Choosing an optimal algorithm for solving a specific problem depends specifically on three properties:

- ▷ Accuracy – how precise should the estimated solution be.
- ▷ Robustness – should the method work for all reasonable choices of initial variables and problem classes.
- ▷ Performance – are there any limits on computational time and/or storage.

A given optimization algorithm often shows trade-offs between these properties. Based on the wanted outcome of the optimization algorithm, the user must decide and prioritize the properties of the algorithm to find the optimal method [16].

The aim of this project is to propose efficient algorithms for the NNLS problem (3.16). In that scope, the main priority in choosing optimal algorithms is on performance. The accuracy is however still of importance, because deconvolution in the first place is applied to gain spatial resolution. Since practical applications often requires multiple runs at a range of frequencies, preference is towards algorithms that can reduce the computational run time. In the following, three conditions are stated that comply with the main aim of choosing efficient deconvolution algorithms for the NNLS problem.

An optimal method should efficiently produce an approximate solution within some accuracy. The performance of optimization algorithms can be estimated by theoretical convergence rates, which are usually

measured by the number of iterations a method require to obtain an ϵ-optimal solution, i.e. $f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \epsilon$, for a given accuracy $\epsilon > 0$. The first condition on an optimal method for solving the NNLS problem (3.16) is

▷ An optimal method for (3.16) should achieve $f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \epsilon$ as fast as possible.

The problem dimensions considered herein will often be given by a square grid of 70×70 points[3], which results in a fully populated square matrix A and Hessian $A^T A$ of dimensions 4900×4900. The computational work associated with arithmetic operations, such as a matrix-vector multiplication, can be assessed by using 'big $\mathcal{O}$' notation. The $\mathcal{O}$ shows the approximate number of flops (floating-point operations per second) involved in some numerical operation. For instance, if the number of operations used in solving some problem can be expressed by $w(n) = c_1 n^3 + c_2 n^2 + c_3 n$, then as $n \to 0$, the dominating term is $w(n) \approx c_1 n^3$, which is written $\mathcal{O}(n^3)$ in 'big $\mathcal{O}$' notation [19]. The maximum complexities of some arithmetic operations are given in Table 3.3.

Operation	Complexity (flops)
$\mathbf{y} = A^{m\times m} \cdot \mathbf{x}^{m\times 1}$	$\mathcal{O}(m^2)$
$Y = P^{n\times n} * X^{n\times n}$	$\mathcal{O}(n^2 \log(n^2))$
$Y = A^{m\times m} \cdot B^{m\times m}$	$\mathcal{O}(m^3)$

Table 3.2: Computational complexity of some arithmetic operations. $*$ denotes 2D convolution via FFT.

Example 3.2 Consider a random matrix A of size 4900×4900. The storage requirement is about 0.2GB of memory in double precision floating point format in MATLAB. The matrix-matrix multiplication $A^T A$ is timed to approximately 8 seconds on an 2.4 GHz Intel Core 2 Duo processor with 8GB of RAM.

The Hessian should not be formed explicitly due to the size and computational complexity of the matrix-matrix product $A^T A$. The second condition on an optimal method for solving NNLS (3.16) is

▷ An optimal method for (3.16) should avoid explicit formulation of the Hessian $A^T A$.

The matrix A is impractical to work with when comparing to the alternative, which is a 2D FFT-based convolution of the smaller point-spread function with a matrix X, both of dimension $N \times N$.

Example 3.3 In order to compare the complexity result from a matrix-vector multiplication with that of a 2D convolution via FFT, consider again a matrix A of dimensions 4900×4900 and a random vector $\mathbf{x}$ of dimensions 4900×1. The matrix-vector multiplication $\mathbf{y} = A\mathbf{x}$ is timed to 0.7 s when A has no special structure and 0.05 s when A has Toeplitz structure. In comparison, the convolution product between two random matrices P and X, both of dimension 70×70, performed by means of the FFT algorithm is timed to 0.006 s. In this somewhat coarse comparison, the convolution product is about 10 times faster.

An optimization algorithm that rely on direct computations or factorization of A will evidently be much slower. This makes the third condition on an optimal method,

▷ An optimal method for the NNLS problem (3.16) should be matrix-free, i.e. all matrix-vector products $\mathbf{y} = A\mathbf{x}$ should be written as a 2D convolution product $Y = \text{PSF} * X$, where Y, PSF and X are matrices of same dimension as the computational grid.

[3] This grid size is a good compromise between the degree of detail and computational run time.

Furthermore, an optimal method should be robust towards different choices of problem dimensions and degrees of ill-conditioning of A and PSF.

These conditions summarize the most important properties of an optimal method for solving the NNLS problem (3.16). A general class of algorithms that meet these conditions are known as gradient methods. They rely only on function evaluations and first-order derivatives of the objective function, and thus belong to the class of first-order methods. Each iteration is relatively inexpensive and the implementation is straightforward. The main drawback is a slow convergence when high accuracy solutions are sought, although gradient methods often remain the only practical alternative for large-scale problems [20].

Efficient algorithms can be built from methods that assume a shift-invariant point-spread function such as Richardson-Lucy and DAMAS. It is shown in [8], that these methods are outperformed by a simple gradient method denoted FFT-NNLS. This method serves as a reference method for the exploration of efficient NNLS solvers in the following.

3.4.1 Gradient methods

One of the most basic first-order algorithms for unconstrained optimization, is the class of descent methods, which takes the form [17]

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k. \tag{3.24}$$

Given a starting point $\mathbf{x}^0$, take a step α_k in a descent direction $\mathbf{d}^k$. A typical choice of $\mathbf{d}^k$ is in direction of the negative gradient, i.e. $\mathbf{d}^k = -\nabla f(\mathbf{x})$. Methods with this particular choice of $\mathbf{d}^k$ are known as gradient descent methods – or simply gradient methods. Equation (3.24) is repeated until $\nabla f(\mathbf{x}) = \mathbf{0}$ or rather $\|\nabla f(\mathbf{x})\| \leq \epsilon$, where ϵ is some small predefined positive number. While this idea of taking a step in the direction the negative gradient direction is intuitively appealing, it can however be inefficient for specific problems characterized by zig-zagging [20] – see Figure 3.3.

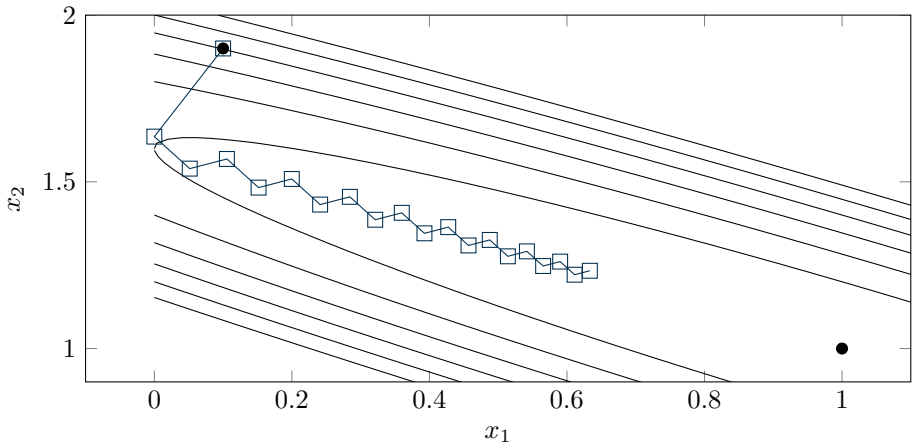


Figure 3.3: Zig-zagging produced by a gradient method on a well-conditioned 2D quadratic problem. The black points represent the start guess and exact minimizer of the function.

The steplength α_k can be calculated by a linesearch along the descent direction. The linesearch is called exact, if α_k is determined analytically. Alternatively, backtracking-linesearch can be employed, where trial points along the descent direction are evaluated against a descent condition, e.g. (3.21) [16]. This requires more function evaluations which add to the total complexity of the algorithm.

The 'big $\mathcal{O}$' notation is also used to assess the performance (unfortunately also denoted complexity) of optimization algorithms. The worst-case complexity of a gradient method is $\mathcal{O}(1/\epsilon^2)$. That is, it requires at

most k iterations to achieve an ϵ-optimal solution $f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \epsilon$ [18]. For instance, at most $k = 10000$ iterations are required to achieve a 1% error in objective function value given a complexity of $\mathcal{O}(1/\epsilon^2)$. A best-case complexity of any first-order method can be shown to have $\mathcal{O}(1/\sqrt{\epsilon})$ [18]. This means that a 1% error in objective function value can be obtained, in the best case, with only 10 iterations. Methods that attain this complexity are known as optimal first-order methods. The complexity bounds are summarized in Table 3.3.

Problem class	Smooth & Convex
Optimal first-order methods	$\mathcal{O}(1/\sqrt{\epsilon})$
Gradient methods	$\mathcal{O}(1/\epsilon)$
Fast gradient methods	$\mathcal{O}(1/\sqrt{\epsilon})$

Table 3.3: Complexity bounds for optimal first-order methods, gradient methods, and fast gradient methods, where $\epsilon > 0$ is a small constant [18].

No ordinary gradient method attains the optimal complexity of a first-order method. The fast gradient methods however, provide an optimal convergence rate for a first-order method. The full complexity estimates for gradient methods with fixed stepsize $\alpha_k = \frac{1}{L}$ on smooth convex functions are given by [18]

$$\begin{aligned} f(\mathbf{x}^k) - f(x^*) &\leq \frac{L}{2k}\|\mathbf{x}^0 - \mathbf{x}^*\|_2^2 \qquad \text{Gradient method} \\ f(\mathbf{x}^k) - f(\mathbf{x}^*) &\leq \frac{2L}{(k+1)^2}\|\mathbf{x}^0 - \mathbf{x}^*\|_2^2 \quad \text{Accelerated gradient method,} \end{aligned} \tag{3.25}$$

where the estimates are given as function of iterations k and $L > 0$ is a Lipschitz constant. The convergence rate depends on a good initial guess, that is if $\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2$ is small then the method converges faster initially.

Solving *constrained* optimization problems, such as NNLS in (3.16), requires that the method only takes steps within the feasible set determined by the constraints. This can be accomplished by projecting the solution and gradient onto the feasible set in each iteration. For NNLS (3.16) this requires a projection onto the nonnegative orthant $\mathbb{R}^n_+$. The gradient projection method with exact linesearch (GPL) is given by

---

Algorithm: *Gradient projection with exact linesearch (GPL)*

Given a starting point $\mathbf{x}_0$ and $\epsilon > 0$, repeat until $\|\nabla f(\mathbf{x})\| \leq \epsilon$:

1. *Search direction*: $\mathbf{d}^k = -\nabla f(\mathbf{x}^k)_+$, where

$$\partial_i f(\mathbf{x})_+ = \begin{cases} 0 & \text{if } x_i = 0 \quad \text{and} \quad \partial_i f(\mathbf{x}) > 0 \\ \partial_i f(\mathbf{x}) & \text{if } x_i > 0. \end{cases} \tag{3.26}$$

2. *Exact line search*: Calculate optimal step

$$\alpha_k = \frac{\mathbf{g}^T\mathbf{r}}{\mathbf{g}^T\mathbf{g}}, \quad \mathbf{g} = A\mathbf{d}^k. \tag{3.27}$$

3. *Update*: $\mathbf{x}^{k+1} = \mathcal{P}_+(\mathbf{x}^k + \alpha_k\mathbf{d}^k)$,

where $\mathcal{P}_+(\cdot)$ is the orthogonal projection of $\mathbf{x}$ onto $\mathbb{R}^n_+$.

The steplength α that minimizes $f(\mathbf{x} + \alpha\mathbf{d})$ is determined by differentiating with respect to α and solving for $\nabla f(\mathbf{x} + \alpha\mathbf{d}) = 0$. In the general case with a quadratic objective function on the form (3.18), α is given by [16]

$$\alpha = \frac{\nabla f(\mathbf{x})^T \nabla f(\mathbf{x})}{\nabla f(\mathbf{x})^T P \nabla f(\mathbf{x})}, \tag{3.28}$$

which reduces to (3.27) for $P = A^T A$ and $\nabla f(\mathbf{x}) = A^T \mathbf{r}$. This algorithm is described in [8] and denoted NNLS. To avoid confusion with the optimization problem (3.16) it has been renamed to GPL.

The gradient projection $\nabla f(\mathbf{x})_+$ ensures that the gradient is projected onto the feasible set and that the update $\mathbf{x}^{k+1}$ is only performed for the set of variables $x_i > 0$. That is, if a variable x_i encounters a constraint it is fixed at the bound $x_i = 0$ and the gradient search direction is bent in order to stay within the feasible set [16].

No complexity bound exists on the form (3.25) due to the exact line search which produces a variable step length for each iteration. A complexity bound for each iteration is given by [16]

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 f(\mathbf{x}^k) - f(\mathbf{x}^*), \tag{3.29}$$

where $\lambda_1 \leq \ldots \lambda_n$ are the eigenvalues of the Hessian for $\mathbf{x}^*$. Each iteration will reduce the error in f by a factor $[(\lambda_n - \lambda_1)/(\lambda_n + \lambda_1)]^2$, which is $\ll 1$ for well-conditioned problems and very close to one for ill-conditioned problems. This bound explains why gradient methods can perform very inefficiently for ill-posed problems.

A recently proposed method [21] shows that the efficiency can be improved by introducing another choice of stepsizes, originally developed for unconstrained optimization. The gradient projection method with Barzilai & Borwein [22] steps (GPBB), is given by

---

Algorithm: *Gradient projection method with BB steps (GPBB)*

Given a starting point $\mathbf{x}^0$, $\mathbf{x}^1$, $\epsilon > 0$, $\sigma \in (0,1)$, $t \in (0,1)$, and integer K

1. *For* $i = 1, \ldots$, repeat until $\|\nabla f(\mathbf{x}^i)_+\| \leq \epsilon$, where $\nabla f(\mathbf{x})_+$ is defined in (3.26). Set $\hat{\mathbf{x}}^0 = \mathbf{x}^{i-1}$ and $\hat{\mathbf{x}}^1 = \mathbf{x}^i$.

   ▷ *For* $j = 1, \ldots K$ *compute stepsizes:*

   $$\mathbf{g} = A^T A \cdot \nabla f(\mathbf{x}^{j-1})_+$$

   $$\alpha^j = \begin{cases} \dfrac{\|\nabla f(\mathbf{x}^{j-1})_+\|^2}{\nabla f(\mathbf{x}^{j-1})_+^T \cdot \mathbf{g}} & \text{for } j \text{ odd} \\ \dfrac{\nabla f(\mathbf{x}^{j-1})_+^T \cdot \mathbf{g}}{\|\mathbf{g}\|^2} & \text{for } j \text{ even} \end{cases}$$

   and update $\hat{\mathbf{x}}^j = \mathcal{P}_+(\hat{\mathbf{x}}^j - \beta^i \alpha^j \nabla f(\hat{\mathbf{x}}^j))$.

   ▷ *If* $f(\hat{\mathbf{x}}^c) - f(\hat{\mathbf{x}}^{c+K}) \geq \sigma \nabla f(\hat{\mathbf{x}}^c)^T (\hat{\mathbf{x}}^c - \hat{\mathbf{x}}^{c+K})$

   $$\mathbf{x}^{i+1} = \hat{\mathbf{x}}^K, \quad \beta^{i+1} = \beta^i$$

   ▷ *Else* Diminish Optimistically: $\beta^{i+1} = t\beta^i$

The GPBB algorithm uses alternating BB-stepsizes to accelerate convergence. Global convergence of the algorithm can only be guaranteed with backtracking linesearch. However, backtracking requires computational expensive function evaluations for each iteration, which would cause the algorithm to be inefficient. The GPBB algorithm basically checks a backtracking condition at every K iterations to reduce the number of function evaluations. If the backtracking conditions holds, the iterate is accepted, otherwise the parameter β is diminished and the process is repeated. The two algorithms GPL and GPBB mainly differ in the choice of stepsizes, yet efficiency of GPBB is seen to improve in many cases [21, 23].

The fast gradient methods promise an optimal $\mathcal{O}(1/\sqrt{\epsilon})$ rate of convergence [18]. This class of methods are similar to the gradient projection methods, yet they attain (in theory) a superior rate convergence. The fast gradient projection method (FGP) with constant stepsize $\alpha_k = \frac{1}{L}$ is given by [24]

Algorithm: *Fast gradient projection method (FGP)*

Given a starting point $\mathbf{x}^0$ set $\mathbf{y}^1 = \mathbf{x}^0$, $t_1 = 1$, and L the Lipschitz constant of ∇f. For $k \geq 1$ repeat until $\|\nabla f(\mathbf{x})_+\| \leq \epsilon$

1. *Update* $\mathbf{x}^k = \mathcal{P}_+(\mathbf{y}^k - \frac{1}{L}\nabla f(\mathbf{y}^k))$.
2. Calculate stepsize $t_{k+1} = \dfrac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$
3. Calculate auxiliary vector $\mathbf{y}^{k+1} = \mathbf{x}^k + \dfrac{t_k - 1}{t_{k+1}}(\mathbf{x}^k - \mathbf{x}^{k-1})$

The main difference between FGP, GPL, and GPBB, lies in the auxiliary point $\mathbf{y}$. Where the gradient methods takes one step in the direction of the negative gradient, FGP takes two, even if the second step takes the solution to a point that is not a minimum of the objective function in that direction. This multi-step

scheme has also been exploited in other algorithms and the extra term is sometimes denoted *momentum*. One such method is known as the heavy-ball method [25].

The FGP method is strictly speaking not a gradient *projection* method, since the gradient $\nabla f(\mathbf{y})$ is calculated without projection onto the feasible set. However, for the sake of completeness it is dubbed FGP throughout.

The FGP method with fixed stepsize $\alpha_k = \frac{1}{L}$ can be interpreted as minimizing a local quadratic approximation of f with Lipschitz constant L. This idea can be illustrated by Figure 3.2. Where GPL and GPBB take a step in the steepest descent direction determined by the first-order taylor approximation of f around $\mathbf{x}$, FGP makes use of the second-order Taylor approximation of a quadratic approximation of f around $\mathbf{x}$. The stepsize $\alpha_k = \frac{1}{L}$ is exactly what is required to attain a minimum of the local quadratic approximation of f. However, if the estimate of L is too conservative, the stepsize will be small and convergence will be slow.

Multi-step methods can exhibit non-monotonic descent, i.e. $f(\mathbf{x}^{k+1}) \not< f(\mathbf{x}^k)$, which is indeed the case for FGP. While non-monotonic descent is one of the properties that comes with the accelerated scheme it can be impractical for finding an optimal solution. The FGP method can easily be turned into a descent method by adding a logical statement after each update of $\mathbf{x}$: If $f(\mathbf{x}^k) > f(\mathbf{x}^{k-1})$ then set $\mathbf{x}^k = \mathbf{x}^{k-1}$. It can be shown that the descent version of FGP shares the convergence rate of FGP [26].

The Lipschitz constant L can be estimated explicitly in some cases and specifically for NNLS (3.16) by computing the largest eigenvalue of the Hessian A^TA. If the calculation of the Hessian is infeasible or impractical, the largest eigenvalue can be approximated by power iteration [27]: $L = \|A\|_2^2 = \lambda_{\max}(A^TA)$, given a random vector $\mathbf{x}^0$ iterate for $k = 1 \dots N-1$:

$$\mathbf{x}^{k+1} = \frac{A\mathbf{x}^k}{\|\mathbf{x}^k\|_2}, \tag{3.30}$$

then $L = \|\mathbf{x}^N\|_2^2$. Numerical tests show that 5-10 iterations are sufficient to obtain convergence. Note that the matrix-vector multiplication $A\mathbf{x}$ can be implemented as a convolution product via FFT, if the assumption of a shift-invariant point-spread function holds.

3.4.2 Other methods

The gradient methods introduced above fulfill the proposed conditions on an optimal method for solving NNLS (3.16). The convergence rates however, are sensitive to the condition number and scaling of the matrix A. In the following some alternatives to the gradient methods are introduced.

Conjugate-gradient (CG) methods are widely used for solving unconstrained optimization problems and are generally known to share the same non-optimal convergence rate of the ordinary gradient methods [18]. The sequence generated by the CG methods can be written as in (3.24)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{3.31}$$

where $\mathbf{d}_{k+1} = -\nabla f(\mathbf{x}_k) + \beta_k \mathbf{d}_k$ [16]. The step direction is no longer the steepest descent, but rather a linear combination thereof. Given a QP on the form (3.18), then $\mathbf{d}_{k-1}^T A^T A \mathbf{d}_k = 0$, i.e. $\mathbf{d}_{k-1}$ and $\mathbf{d}_k$ must be conjugate with respect to the Hessian A^TA. The stepsize α_k is usually found by exact linesearch along $\mathbf{d}_k$. The choice of β_k determines the specific CG method and is often based on the specific problem [28]. The convergence rate of CG methods are also sensitive to the condition number of the Hessian A^TA. In order to remedy this sensitivity, preconditioning of A is often used. A widely used algorithm for unconstrained optimization is given in [29]. For constrained optimization, the algorithm in [30] is generally known to provide the best results. One newly proposed method is given in [31].

The available algorithms for constrained optimization are rather complex and not easily turned matrix-free as it is required by the conditions for an efficient method for solving the NNLS problem, stated in Section 3.4.

The descent direction $\mathbf{d}_k$ from the basic iterative scheme (3.24) can be written in more general terms. Recall from (3.21) that the first-order Taylor approximation around $\mathbf{x}_k$ can be written

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k),$$

applying the gradient operator to both sides and letting $\nabla f(\mathbf{x}) = 0$ gives

$$\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k = -\nabla f(\mathbf{x}_k), \tag{3.32}$$

then $\mathbf{d}_k$ is the solution of the linear system of equations and furthermore the basis of Newton's method, which is generally known to be quadratically convergent [16]. Newton's method requires at most $\mathcal{O}(\frac{1}{3}n^3)$ flops in each step if $\nabla^2 f(\mathbf{x}^k)$ has no special structure such as Toeplitz or tridiagonal that can be exploited. This can be infeasible or inefficient for large scale problems such as NNLS (3.16). The gradient methods approximate the Hessian $\nabla^2 f(\mathbf{x}) \approx I$ as the identity matrix and does not depend on solving a linear system of equations for each iteration. However, there exist a certain class of methods suitable for large-scale problems that approximate the Hessian in a more computational efficient way. This class of methods are known as Quasi-Newton methods, where the most well-known is the BFGS update of $\mathbf{d}_k$ [16]. A limited-storage version of the BFGS update, known as L-BFGS, is more practical for large scale problems and a Projected Quasi-Newton method with this L-BFGS update, suitable for optimization of (3.16), is given in [32]. These methods are however more complex than the gradient methods and therefore not easily reprogrammable to be matrix-free as required by an optimal method for NNLS (3.16).

Interior point methods solve optimization problems iteratively by approaching the solution from the interior of the feasible set. Each iteration is relatively expensive – the order of Newton's method – but significant improvement can be made in a few iterations [16]. In general, this makes interior point methods impractical for large-scale problems, at least in terms of efficiency. However, some recently proposed improvements show promising performance for (very) large-scale problems [33, 34]. Specifically, ref. [34] proposes a matrix-free method, that could be efficient on the NNLS problem.

Similar to the CG methods, the algorithm is very complex and difficult to adapt to the NNLS problem.

In summary it is concluded that only the gradient methods fulfill the conditions on an optimal method for NNLS (3.16).

3.5 Model implementation

The deconvolution and beamforming algorithms introduced above as well as the computational model are implemented with MATLAB. First, the model geometry is established. The area of interest is assumed to be given at some discrete points (x, y), at a distance of z_0, represented by an equidistant grid of $N \times N$ points, given by

$$x = y = \left\{-\tfrac{d_0}{2}, -\tfrac{d_0}{2} + n\Delta x, \ldots, \tfrac{d_0}{2}\right\}, \quad \text{for } n = 1 \ldots, N-1$$

where $d_0 = 2z_0 \tan(\phi)$ is the area covered at distance z_0 with beamformer opening angle ϕ and $\Delta x = d_0/N$. The array geometry is either calculated or loaded from file and is represented by (x_0, y_0) coordinates of the M microphone positions in the observation plane. Second, the beamformer map is calculated. The steering vectors are calculated from Equation (2.27) and the point-spread function is given by Equation (3.1) with a single unit-strength centered source. The beamformer map is given by Equation (2.26) and calculated

with an experimentally derived cross-spectrum matrix (2.20) or a synthetic for simulation purposes (2.25). Third, the deconvolution process is started, working exclusively with 2D arrays. The beamformer map B, point-spread function PSF, and a starting guess X^0, all of size $N \times N$, are padded with zeros to a size of $2N \times 2N$ and supplied to each of the deconvolution algorithms. The deconvolution algorithms are also supplied with the function to be minimized – in this case the least-squares objective function in NNLS, given by Equation (3.16). The function takes inputs $X^{2N\times 2N}$, $B^{2N\times 2N}$ and $\text{PSF}^{2N\times 2N}$ and gives the residual $R^{2N\times 2N}$, objective function value f and gradient $\nabla f^{2N\times 2N}$ as outputs. The outputs are calculated by means of FFT and the relation between variables and matrix-vector notation is given below, where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denotes the Fast Fourier Transform and its inverse. The deconvolution algorithms iterate until a stopping criterion has been met and zero-padding is removed from the resulting X^k and given as output.

The relation between computations with 1D and 2D arrays are summarized in Table 3.4.

	Vector		Matrix		FFT
Residual	$\mathbf{r} = A\mathbf{x} - \mathbf{b}$	$\leftrightarrow$	$R = AX - B$	$\leftrightarrow$	$R = \mathcal{F}^{-1}\left(\mathcal{F}(X) \cdot \mathcal{F}(\text{PSF})\right) - B$
Objective function f	$\frac{1}{2}\|\mathbf{r}\|_2^2$	$\leftrightarrow$	$\frac{1}{2}\|R\|_F^2$	$\leftrightarrow$	$\frac{1}{2}\|R\|_F^2$
Gradient ∇f	$A^T\mathbf{r}$	$\leftrightarrow$	$A^T R$	$\leftrightarrow$	$\mathcal{F}^{-1}\left(\mathcal{F}(R) \cdot \mathcal{F}(\text{PSF}_{90})\right)$

Table 3.4: Relation between 1D & 2D computation of residual, objective function and gradient. PSF_{90} denotes the 90° rotated point-spread function.

The MATLAB implementation for calculation of the point-spread function, beamformer map and cross-spectrum matrix is given in Appendix A. Furthermore, the deconvolution algorithms GPL, GPBB and FGP are given as reference.

4 Simulation Study

Simulated data and models provide an easy accessible way to test various scenarios without time consuming experimental setups, measurements, and post processing. Some scenarios might even be infeasible to do experimentally, whereas simulation studies can provide an equally important insight to the performance of deconvolution algorithms. One important pitfall to keep in mind when working with reconstruction algorithms is that there is a risk of getting perfect, but unrealistic reconstructions due to inverse crime, that is, when simulated data resonates constructively with the reconstruction algorithm [35, 36]. Avoiding inverse crime require the forward model, that produces the simulated data, to be calculated on a different computational grid than the grid used in the reconstruction. In the following simulation study this is achieved by computing the point-spread function and beamformer map at two slightly different distances z_0 and z_{true}, respectively. The true distance is computed by adding 10% to z_0, $z_{\text{true}} = 1.1z_0$ and furthermore, following [35], the point-spread function is computed on a larger computational grid $1.3N \times 1.3N$ and interpolated to size $N \times N$ by the function `interp2` in MATLAB.

Benchmarking computational time of the deconvolution algorithms are performed with MATLAB functions `tic` and `toc`. The algorithms are optimized to avoid bottlenecks by the Profiler function in MATLAB and the structure of the different algorithms is streamlined. Still, the computational run time of a given algorithm is subject to uncertainties from overhead of initial memory allocation and parallelization of certain functions. Consequently, the computational times are only given with 2 significant digits and timing is performed after an initial run to reduce overhead. Furthermore, the benchmarks are derived on an Intel Core 2 Duo 2.4 GHz processor with 8 GB RAM, which consequently determines the outcome of the benchmarks. For that reason, the computational run time of one algorithm should only be assessed in relation to the other algorithms and not as an absolute measure of efficiency.

In the following, 3 different point source configurations are used to evaluate the performance and reconstruction quality of deconvolution algorithms GPL, GPBB and FGP. A complete summary and comparison of the tests in Chapter 4 and 5 are given in Chapter 6.

4.1 Single point source

Consider a single point source located at the center of an equidistant $N \times N$ grid. The distance z_0 to the point source, frequency f, and opening angle ϕ of the beamformer is varied in order to test the deconvolution algorithms in different situations. In general, the combination of the beamformer opening angle ϕ and observation distance z_0 can be used to impose different degrees of shift-variant point-spread functions. For instance, a large opening angle $\phi = 30^\circ$ and short observation distance $z_0 = 1\,\text{m}$ result in point-spread function that is more shift-variant than a point-spread function calculated with a small opening angle $\phi = 10^\circ$ and large observation distance $z_0 = 10\,\text{m}$. Examples of such were shown in Section 3.1. In practice, some degree of shift-variance will always be present, no matter how ϕ and z_0 are chosen.

In the following, the point-spread functions are denoted in a less strict sense, where a statement like: *The shift-invariant point-spread function*, refers to a point-spread function that to a good approximation is shift-invariant.

The main efficiency gain of the deconvolution algorithms is achieved by making use of the fact that matrix-vector multiplications $\mathbf{y} = A\mathbf{x}$ can be replaced by a linear convolution with the shift-invariant point-spread function $Y = X * \text{PSF}$. For this reason it is of great interest to consider shift-invariant point-spread functions. In simulation studies, this is an easy property to achieve, as long as the beamformer angle is chosen small enough. In practice however, the measurement setup might restrict the measurement objective to some fixed distance and the beamformer angle must be chosen appropriately. The assumption of shift-invariance is used throughout in the following, however since some degree of shift-variance will inevitably be present, a mismatch between the model (shift-invariant) and data (shift-variant) arises, that affects the performance of the algorithms and resolution of the deconvolved maps. This mismatch is referred to as a model-data mismatch.

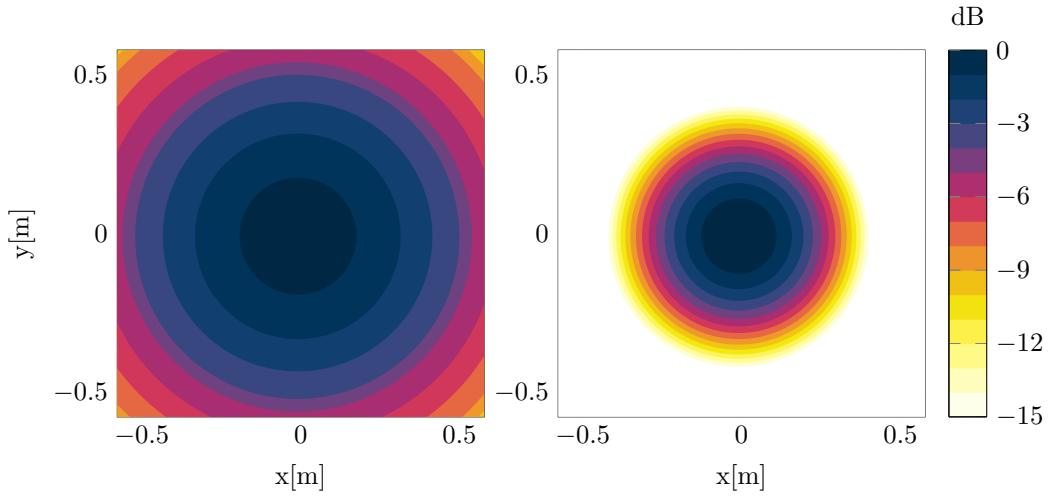
According to the convolution theorem [6], convolution in the spatial domain is equivalent to multiplication in the frequency domain,

$$f * g = \mathcal{F}^{-1}\left(\mathcal{F}(f) \cdot \mathcal{F}(g)\right), \tag{4.1}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denotes the Fourier Transform and its inverse. This property can be implemented with the Fast Fourier Transform (FFT) to calculate the convolution product efficiently. The FFT performs a circular convolution under the assumption that f and g are periodic. This is in general not the case and zero-padding is applied in order to reduce wrap-around effects. This means that a $N \times N$ grid is padded with zeros to size $2N \times 2N$.

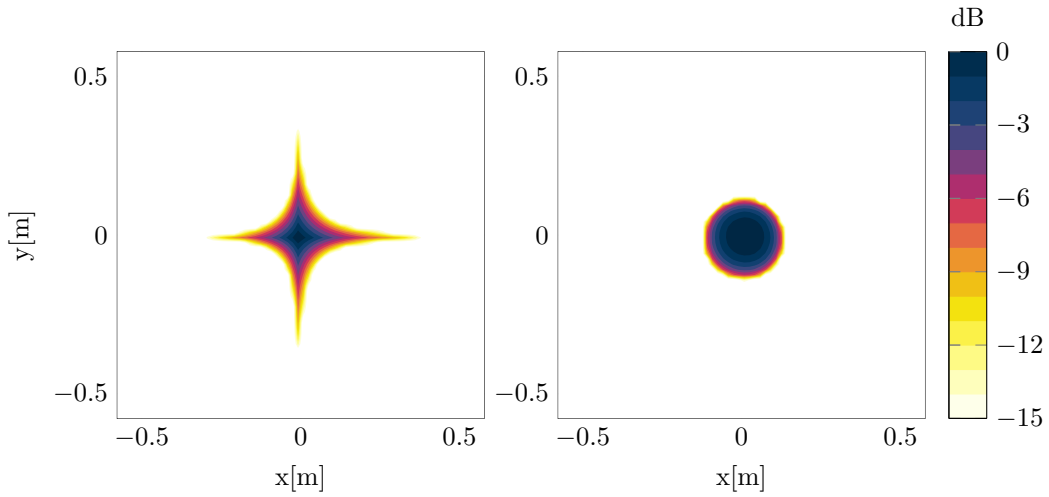
Adding zero-padding can also be seen as imposing a boundary condition on the point-spread function and beamformer map. Consider the beamformer maps given by a centered point source in a $N \times N$ grid with $N = 70$ located at $z_0 = 1\,\text{m}$, at frequencies $f = 500\,\text{Hz}$ and $f = 1000\,\text{Hz}$ depicted in Figure 4.1. Since only a single point source is considered, the point-spread function and beamformer maps are equivalent. The physical interpretation of the zero boundary condition depends on the level of the beamformer maps at the boundary. Adding a zero boundary condition to the beamformer map at $f = 500\,\text{Hz}$ will produce a discontinuity at the boundary, causing an unphysical situation in which the beamformer's response to a point source suddenly drops to zero outside the $N \times N$ grid. The map at $f = 1000\,\text{Hz}$ has a much lower level at the boundary, so adding a zero boundary condition will comply more reasonably with the actual physical situation. The effect of zero-padding is depicted in Figure 4.2, where the deconvolved maps for $f = 500\,\text{Hz}$ and $f = 1000\,\text{Hz}$ are shown. The discontinuity caused by zero-padding is clearly affecting the deconvolved map at $f = 500\,\text{Hz}$. The levels at the edge of the maps in Figure 4.2 are approximately $-5\,\text{dB}$ at $500\,\text{Hz}$ and approaching $-\infty$ dB at $1000\,\text{Hz}$. This suggests that there must be a lower frequency limit of deconvolution in which boundary effects are dominating the reconstructed maps. The exact lower limit is eventually determined by the source distribution and beamformer configuration. From the present investigation, a proposed rule of thumb could be that the maximum normalized level at the boundary of a beamformer map be less than $-10\,\text{dB}$ to avoid boundary effects.

In this particular case there is a lower frequency limit of $f \approx 800\,\text{Hz}$. One way of improving this limit is to impose other boundary conditions. One option is to make a periodic extension of the beamformer map and point-spread function to avoid discontinuities [37]. The point-spread function with periodic boundary condition and a deconvolved map at $f = 500\,\text{Hz}$ is shown in Figure 4.3. The imposed boundary condition improves the deconvolved map in comparison to the zero boundary case in Figure 4.2a, although boundary artifacts are still visible by the asymmetric appearance.



(a) 500 Hz (b) 1000 Hz

Figure 4.1: Two beamformer maps. Point source located at $z_0 = 1\,\mathrm{m}$ and beamformer opening angle $\phi = 30°$.

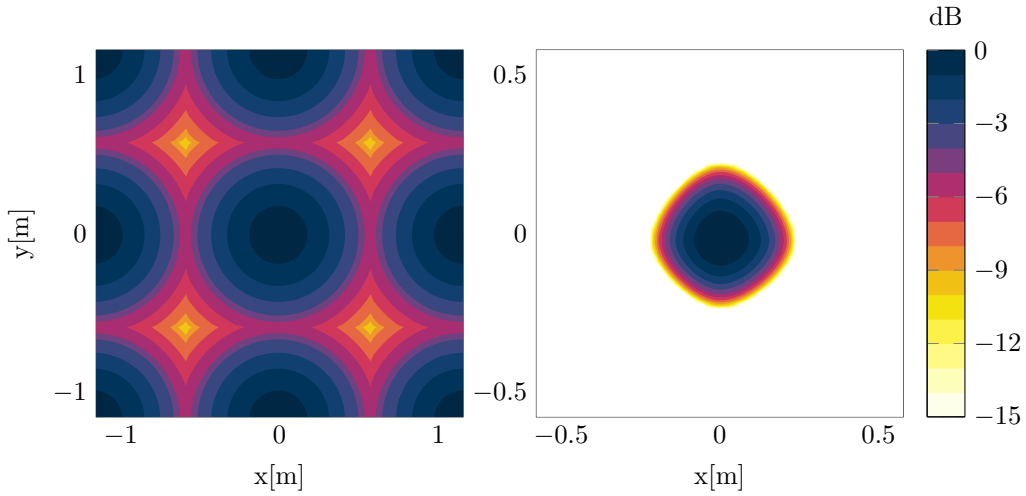


(a) 500 Hz (b) 1000 Hz

Figure 4.2: Two deconvolved maps produced by 200 iterations with deconvolution algorithm GPL.

The physical interpretation of imposing a periodic boundary condition is not meaningful when dealing with sound fields, it merely serves as a tool for visual improvement of the deconvolved map and for that reason it will not be used in the deconvolution of beamforming maps henceforth. The zero boundary condition is neither meaningful when approaching the lower frequency limit, therefore deconvolution will only be considered at frequencies for which the maximum normalized level at the boundary of beamformer map is not too large.

The performance of the deconvolution algorithms is evaluated by two measures: the decrease in norm of the normalized projected gradient $\|\nabla f(\mathbf{x}^k)_+\|/\|\nabla f(\mathbf{x}^0)_+\|$ and the decrease in objective function value $\left(f(\mathbf{x}^k) - f(\mathbf{x}^*)\right)/\|\mathbf{x}^0 - \mathbf{x}^*\|^2$. The former provides a measure of closeness to the optimal solution $\mathbf{x}^*$, whereas the latter is proportional to the rate of convergence. This can be seen from the upper complexity



(a) Point-spread function with periodic boundary condition.

(b) Deconvolved map

Figure 4.3: Point-spread function and deconvolved map with periodic boundary condition at $f = 500\,\text{Hz}$.

bound of FGP, stated in Equation (3.25)

$$\frac{f(\mathbf{x}^k) - f(\mathbf{x}^*)}{\|\mathbf{x}^0 - \mathbf{x}^*\|_2^2} \leq \frac{2L}{(k+1)^2}.$$

The right hand side $2L/(k+1)^2$ will be shown as a dashed line in the following where convergence plots of $\left(f(\mathbf{x}^k) - f(\mathbf{x}^*)\right)/\|\mathbf{x}^0 - \mathbf{x}^*\|^2$ are shown. Only FGP is guaranteed to convergence at a rate faster than that line. The optimal solution $\mathbf{x}^*$ is chosen to be the solution at the last iteration of an algorithm.

Take as an example the beamformer map in Figure 4.1b with $z_0 = 1\,\text{m}$, $\phi = 30^\circ$ and $f = 1000\,\text{Hz}$. The deconvolution algorithms are applied with two stopping criteria: Stop if $\|\nabla f(\mathbf{x}^k)_+\|/\|\nabla f(\mathbf{x}^0)_+\| \leq 10^{-4}$ or the number of iterations reach $k = 300$. The former stopping criterion is referred to as the gradient tolerance $\text{TOL}\nabla f(\mathbf{x})_+$ henceforth. The number of iterations and computational times are summarized in Table 4.1. Figure 4.4 shows the norm of the projected gradient and objective function values as function of iterations.

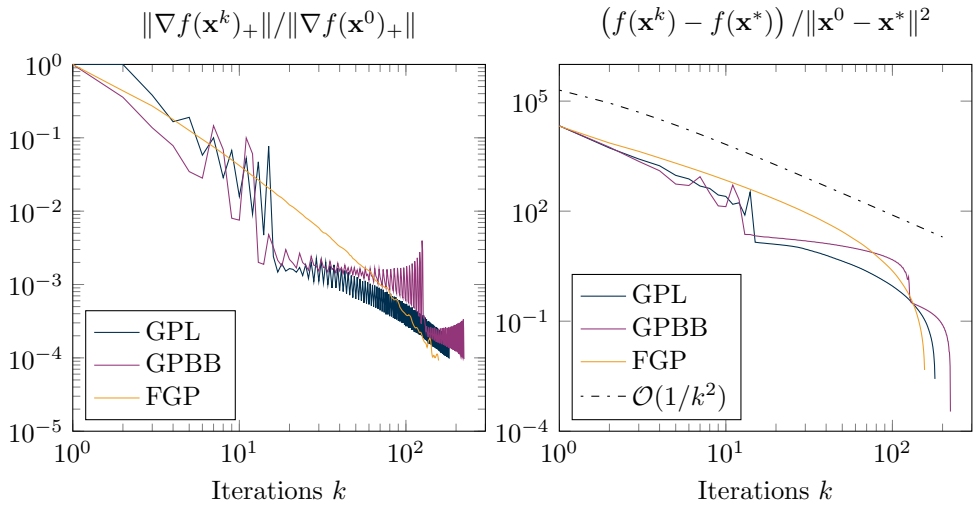
Method	Iterations k	Time $[s]$
GPL	181	1.1
GPBB	223	1.7
FGP	156	1.3

Table 4.1: Results summary for a single point source located at $z_0 = 1\,\text{m}$ with $f = 1000\,\text{Hz}$ and $\phi = 30^\circ$.

FGP use fewer iterations than GPL and GPBB to satisfy the stopping criterion and can therefore be said to be more efficient in this particular case. The oscillatory behavior exhibited in the norm of the projected gradient of GPL and GPBB in Figure 4.4a, provides a good visual explanation for the cause of the higher iteration count. The convergence rates of the GPL, GPBB and FGP can be seen in Figure 4.4b. The dashed black line represents the theoretical upper bound on the convergence rate of FGP, stated in Equation (3.25). It is expected to see a $\mathcal{O}(1/k^2)$ upper bound on the rate of convergence of FGP, that is, FGP is at all times bound to stay below that line.

In this case all methods perform satisfactory with GPL and GPBB initially attaining a better rate of convergence than FGP. After about 10 iterations, GPL and GPBB stagnate and converge very slowly while FGP attains an increasingly superior rate of convergence.

A similar picture is seen in the shift-invariant case, where the point source is moved to $z_0 = 10\,\mathrm{m}$ at $f = 3000\,\mathrm{Hz}$ and with a beamformer opening angle of $\phi = 10^\circ$. The deconvolution results are summarized in Table 4.2 and Figure 4.5. The rate of convergence of GPL and GPBB is more in line with that of FGP and no stagnation is seen in this case. GPBB in particular requires much fewer iterations to reach the stopping criterion.



(a) Decrease in norm of projected gradient.

(b) Decrease in objective function value.

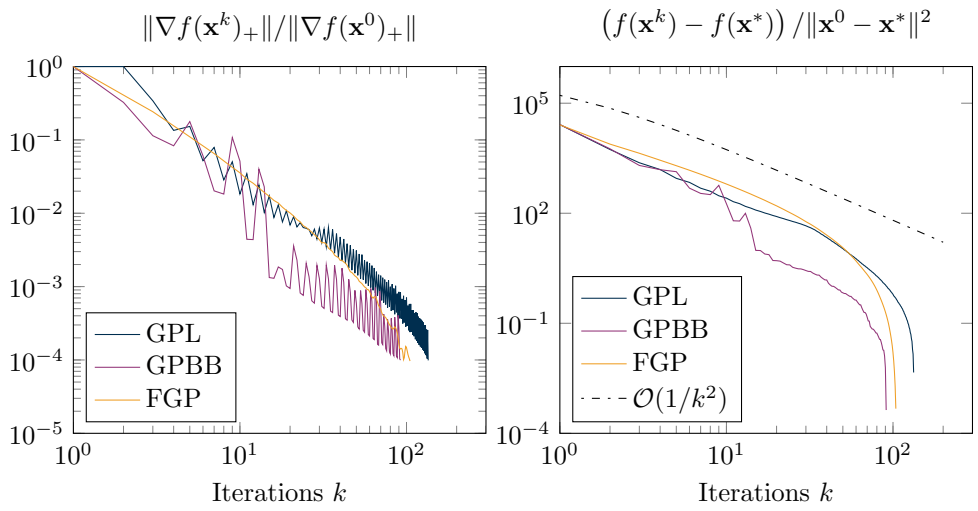
Figure 4.4: Results summary as function of iterations for a single point source located at $z_0 = 1\,\mathrm{m}$ with $f = 1000\,\mathrm{Hz}$ and $\phi = 30^\circ$.

Method	Iterations k	Time $[s]$	$f(\mathbf{x}^*)$
GPL	134	0.9	0.57
GPBB	91	0.8	0.57
FGP	104	1.0	0.57

Table 4.2: Results summary for a single point source located at $z_0 = 10\,\mathrm{m}$ with $f = 3000\,\mathrm{Hz}$ and $\phi = 10^\circ$.

In general these two examples show, that the efficiency of the deconvolution algorithms improves when the point-spread function is shift-invariant. This tendency is further established for a wide frequency range from $f = 2000\,\mathrm{Hz}$ to $4000\,\mathrm{Hz}$ and is summarized in Figure 4.6. The number of iterations is seen to decrease from an average of 200 iterations to around 100 iterations with the shift-invariant point-spread function. Moreover, the number of iterations is dependent on frequency in the case of a shift-variant point-spread function, while it is independent in the shift-invariant case.

The FGP algorithm, described in Section 3.4.1, can be optimized to perform more efficiently since evaluations of $\|\nabla f(\mathbf{x})_+\|$ are only used as a check of convergence and not explicitly used in the computation of $\mathbf{x}$. On average, a decrease in computational time of 40-50% is seen if $\|\nabla f(\mathbf{x})_+\|$ is only computed at every 20 iteration. For instance, the computational times of FGP in Table 4.1 and 4.2 are improved to $0.7\,\mathrm{s}$ and $0.6\,\mathrm{s}$

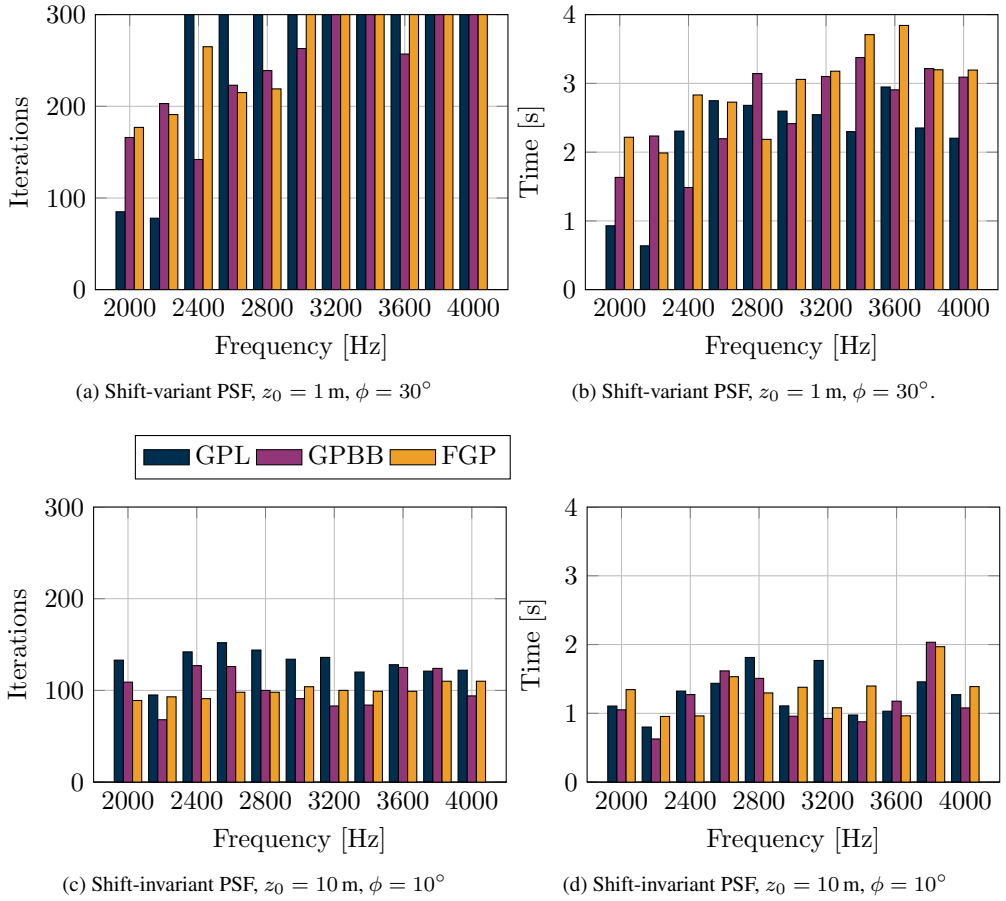


(a) Decrease in norm of projected gradient.

(b) Decrease in objective function value.

Figure 4.5: Results summary as function of iterations for a single point source located at $z_0 = 10\,\text{m}$ with $f = 3000\,\text{Hz}$ and $\phi = 10°$.

respectively. The optimized version of FGP will be used whenever computational run times are stated in the following sections.



(a) Shift-variant PSF, $z_0 = 1\,\mathrm{m}$, $\phi = 30^\circ$

(b) Shift-variant PSF, $z_0 = 1\,\mathrm{m}$, $\phi = 30^\circ$.

(c) Shift-invariant PSF, $z_0 = 10\,\mathrm{m}$, $\phi = 10^\circ$

(d) Shift-invariant PSF, $z_0 = 10\,\mathrm{m}$, $\phi = 10^\circ$

Figure 4.6: Comparison of number of iterations and computational time for deconvolution algorithms applied to models with different point-spread functions. One point-spread function at $z_0 = 10\,\mathrm{m}$ is predominantly shift-invariant.

4.2 Two point sources

Recall that the resolution of the frequency-domain beamformer, given in Equation (2.10), states

$$R = \frac{az\lambda}{\cos^3(\theta)D} = \frac{az_0c}{\cos^3(\theta)fD}.$$

This equation provides an analytical expression for the beamformer's ability to distinguish two point source. Deconvolution is applied in order to gain spatial resolution. The degree of resolution improvement is however dependent on the deconvolution algorithm, number of iteration, frequency, and source configuration. In an attempt to quantify the achievable resolution of the deconvolution process, two point sources are placed first at $z_0 = 3\,\text{m}$ and then at $z_0 = 5\,\text{m}$, separated by $\Delta x \approx 0.4\,\text{m}$, with beamformer opening angles $\phi = 30°$ and $\phi = 10°$. The beamformer resolution at $f = 2000\,\text{Hz}$ is

$$R = \frac{1.22z_0c}{\cos^3(\theta_{sep})f} = \begin{cases} 0.63\,\text{m}, & z_0 = 3\,\text{m}, \phi = 30° \\ 1.04\,\text{m}, & z_0 = 5\,\text{m}, \phi = 10°. \end{cases} \tag{4.2}$$
$$\theta_{sep} = \arctan\left(\frac{\Delta x}{2z_0}\right)$$

At this frequency the beamformer resolution is poorer than the source separation Δx, why the sources are indistinguishable by the beamformer. The beamformer resolution relative to separation distance $R/\Delta x$ provides a dimensionless quantity, that is less than or equal to unity when the two points can be distinguished in the beamformer map. When $R/\Delta x > 1$, the points are indistinguishable by the beamformer.

A quantitative measure of the achievable resolution improvement by means of deconvolution, is given by computing the level of the point exactly between the two point sources as function of $R/\Delta x$. A beamformer map of the two point source distribution is given in Figure 4.7, with the center point marked by a white cross.

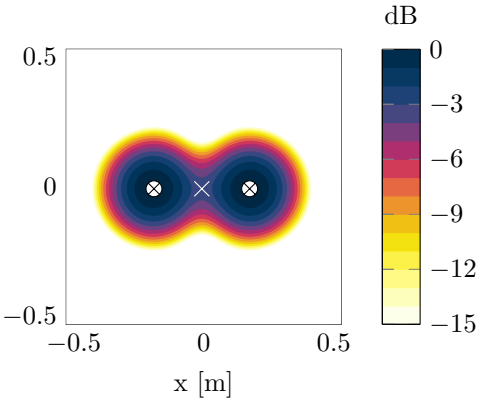
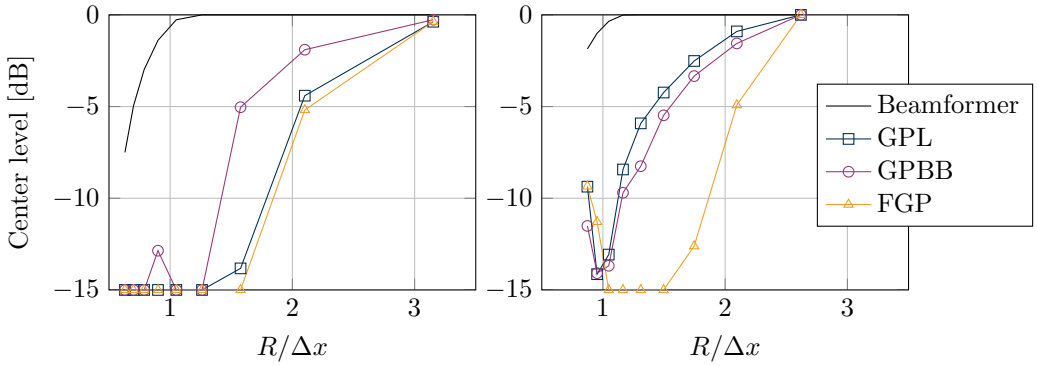


Figure 4.7: Beamformer map with center point indicated by the symbol $\times$ and point source locations with symbol $\otimes$.

The deconvolution algorithms are applied to beamformer maps at frequencies ranging from $1500\,\text{Hz}$ to $6000\,\text{Hz}$ and given stopping criteria of $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ and iteration maximum of 300. The center level and $R/\Delta x$ is stored at each frequency. The results are shown in Figure 4.8.

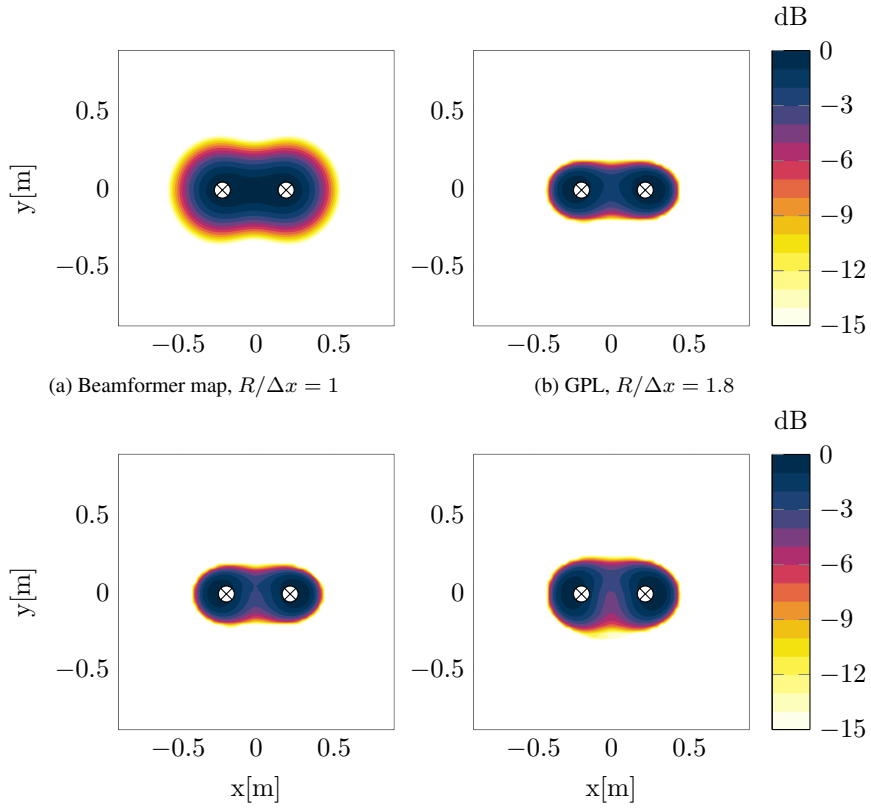
The beamformer resolution $R/\Delta x = 1$, when the center level is approximately $-0.5\,\text{dB}$, why this level serves as a heuristic resolution criterion alike Rayleigh's. It is seen in both cases that this criterion is reached in the deconvolved maps at or above $R/\Delta x = 2$. Deconvolved maps at $z_0 = 5\,\text{m}$ are shown in Figure 4.9 near the resolution limit.



(a) $z_0 = 3\,\text{m}$, $\phi = 30°$.

(b) $z_0 = 5\,\text{m}$, $\phi = 10°$.

Figure 4.8: Center level as function of resolution for a two point source configuration.



(a) Beamformer map, $R/\Delta x = 1$

(b) GPL, $R/\Delta x = 1.8$

(c) GPBB, $R/\Delta x = 1.8$

(d) FGP, $R/\Delta x = 2.1$

Figure 4.9: Beamformer and deconvolved maps at resolution limit. $z_0 = 5\,\text{m}$, $\phi = 10°$, $\otimes$ designates the location of a point source.

From a visual perspective, the point sources are well separated in the deconvolved maps with a 3 dB decrease in level at the center point between the two sources. This support that the heuristic resolution limit, a center level less than $-0.5\,\text{dB}$, is a fair criterion of resolution. The center level is not monotonically decreasing as expected, which could be due to some residual left in the deconvolved maps at high frequencies.

Comparing the center level of the deconvolution algorithms in Figure 4.8, the resolution histories of FGP and GPBB seem to be rather independent on the source configuration and shift-variance of the point-spread

function, while GPL attains a better resolution limit in the shift-invariant case. This suggests that the well known trade-off between spatial resolution and shift-invariance of the point-spread function, also extends to deconvolution and specifically in the case for GPL.

The center level resolution criterion is rather coarse, since it only depends on one point in the map. In an attempt to provide a better measure, the spatial resolution of the deconvolved maps is assessed by the peak-to-average ratio, defined by

$$\text{PAR} = \frac{\min(X_{i,j})}{\overline{|X_{i,j}|^2}}, \tag{4.3}$$

where $X_{i,j}$ is a map in dB, normalized to the maximum value, given at indices i, j, defined by the line connecting the position of the two point sources. As an example, the beamformer map in Figure 4.10 is assigned with $\otimes$ symbols that designate the location of the point sources and a white line connects the two points. The line is given by the indices i, j. The peak-to-average ratio is the minimum value on the line divided by the mean-square value along the line.

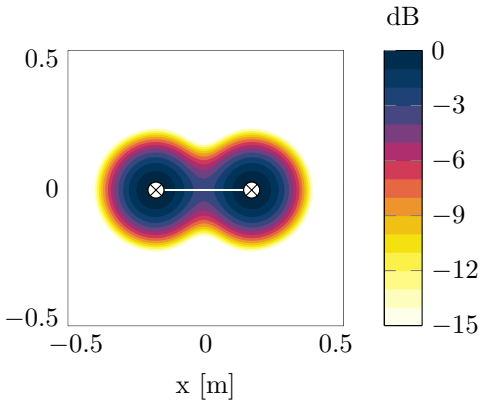
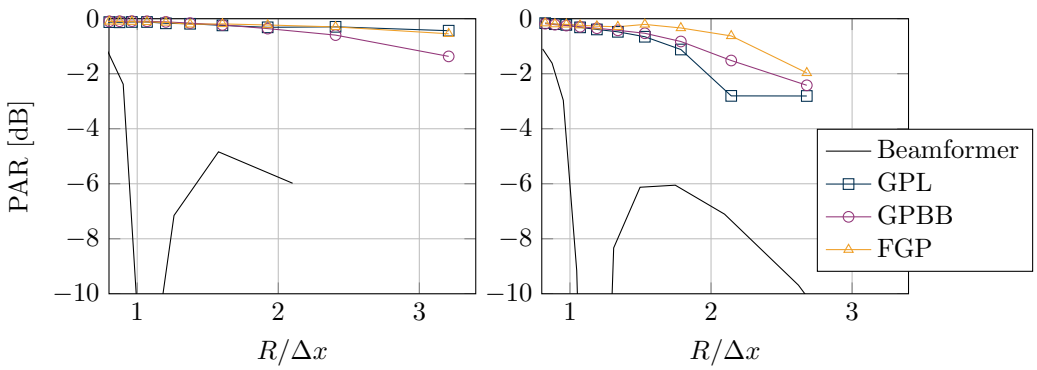


Figure 4.10: Beamformer map that illustrate the peak-to-average measure.

The resulting peak-to-average ratios as function of resolution $R/\Delta x$, for a identical setup as described above, are shown in Figure 4.11. The beamformer map reach $R/\Delta x = 1$ at a peak-to-average ratio of $-6\,\text{dB}$ at $z_0 = 3\,\text{m}$ and $-10\,\text{dB}$ at $z_0 = 5\,\text{m}$. The deconvolved maps never reach this limit which suggest that the points are clearly separated. The limit does not comply with a visual inspection of the deconvolved maps and the previous result using the center level. The peak-to-average ratio is therefore not a good measure of resolution.



(a) $z_0 = 3\,\text{m}$, $\phi = 30°$.

(b) $z_0 = 5\,\text{m}$, $\phi = 10°$.

Figure 4.11: Peak-to-average values as function of beamformer resolution per source separation.

4.3 Warm-start

The rate of convergence of the deconvolution methods GPL, GPBB and FGP can be improved by a warm-start, that is, when a realistic guess of the true solution is supplied as the initial solution x^0. This can be seen be the upper complexity bounds previously stated in Equation (3.25):

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \frac{L}{2k}\|\mathbf{x}^0 - \mathbf{x}^*\|_2^2, \qquad \text{Gradient method}$$
$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \frac{2L}{(k+1)^2}\|\mathbf{x}^0 - \mathbf{x}^*\|_2^2, \qquad \text{Accelerated gradient method.}$$

The term $\|\mathbf{x}^0 - \mathbf{x}^*\|^2$ describes how close the initial guess is to the optimal solution $\mathbf{x}^*$. If this norm is made smaller by proper choice of $\mathbf{x}^0$, the deconvolution algorithms can be expected to reach an ϵ-optimal solution $f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \epsilon$ in fewer iterations and thereby improve efficiency[1].

To assess the achievable efficiency gain by using warm-start, a two point source configuration is used once again. Consider two point sources placed at $z_0 = 3\,\text{m}$, separated by $\Delta x \approx 0.8\,\text{m}$, with beamformer opening angle $\phi = 30°$. The frequency range of interest is $f = 1000\,\text{Hz}$ to $5000\,\text{Hz}$.

As a first assessment, it is of interest to see how the deconvolution algorithms react to a wrong starting guess. Consider a beamformer map produced by only the left of the two point sources and supply that map to the deconvolution algorithms. The initial run is given zeros as a starting guess with the usual stopping criteria, $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ and iteration maximum of 300. The resulting deconvolved maps of the left point are then used as a starting guess for a run where the beamformer map of only the right point source is supplied. A wrong starting guess is given on purpose, in order to test how fast the deconvolution algorithms adapt to the correct solution. By measuring the normalized level at the exact position of the point sources, as function of iterations, the adaptation rate is asserted by the decrease in level of the left point source. As an example, Figure 4.12 show the peak level of the left and right point source as function of iterations. At $k = 0$, the level of the left point source attains a maximum since this is the supplied starting guess, and the level of the right point source is given by the supplied beamformer map. With increasing number of iterations, the level decreases at the left point source and increases at the right point source, as expected. Deconvolved maps produced by GPL at iterations $k = 1$ and $k = 7$ are shown in Figure 4.13.

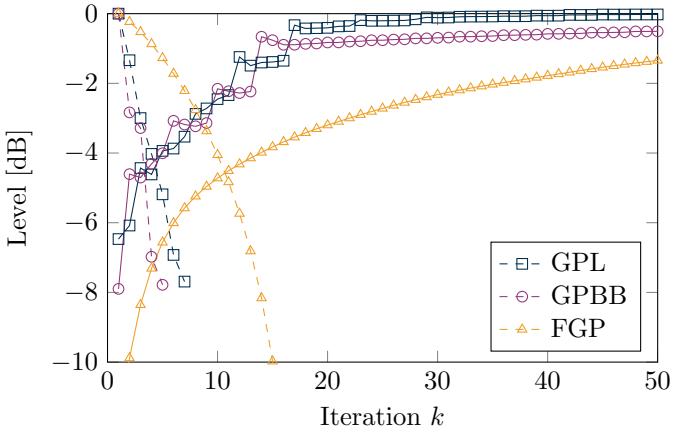


Figure 4.12: Peak level of left (dashed) and right (solid) point source as function of iterations. $f = 3000\,\text{Hz}$.

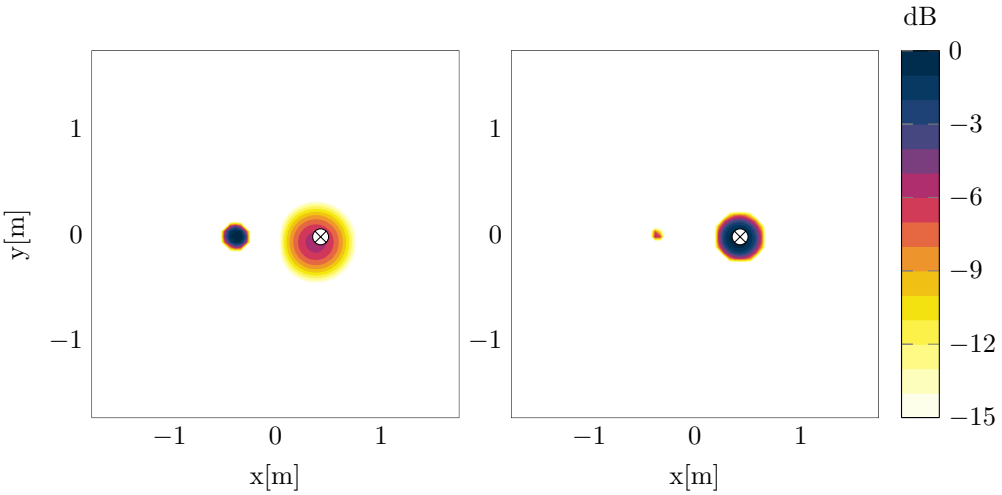
Let iteration k_{adapt}, at which the level of the left point source is below $-10\,\text{dB}$, define the limit where residual from the starting guess is no longer dominant in the deconvolved map. The frequency dependent iterate k_{adapt}

[1] This idea is well-known in mathematical litterature although, to the authors best knowledge, it has not yet been applied to beamforming deconvolution.

is stated in Table 4.3. FGP need noticeably more iterations than GPL and GPBB, to reduce residual from the starting guess. This agrees well with the momentum interpretation of the method discussed in Section 3.4.1. Although a momentum based method can attain improved convergence rates when the distance to the optimal solution is small, it can exhibit slow convergence, as in this case, when a bad initial guess is supplied. The residual of the left point source is carried along for more iterations than the non-momentum based methods GPL and GPBB.

f [Hz]	1000	2000	3000	4000	5000
GPL	7	7	8	6	6
GPBB	8	4	6	8	10
FGP	19	12	16	23	36

Table 4.3: Iteration k_{adapt} at which residual from the starting guess is no longer dominant.



(a) $k = 1$ (b) $k = 7$

Figure 4.13: GPL at iteration $k = 1$ and $k = 7$, $\otimes$ designates the location of a point source. $f = 3000\,\text{Hz}$.

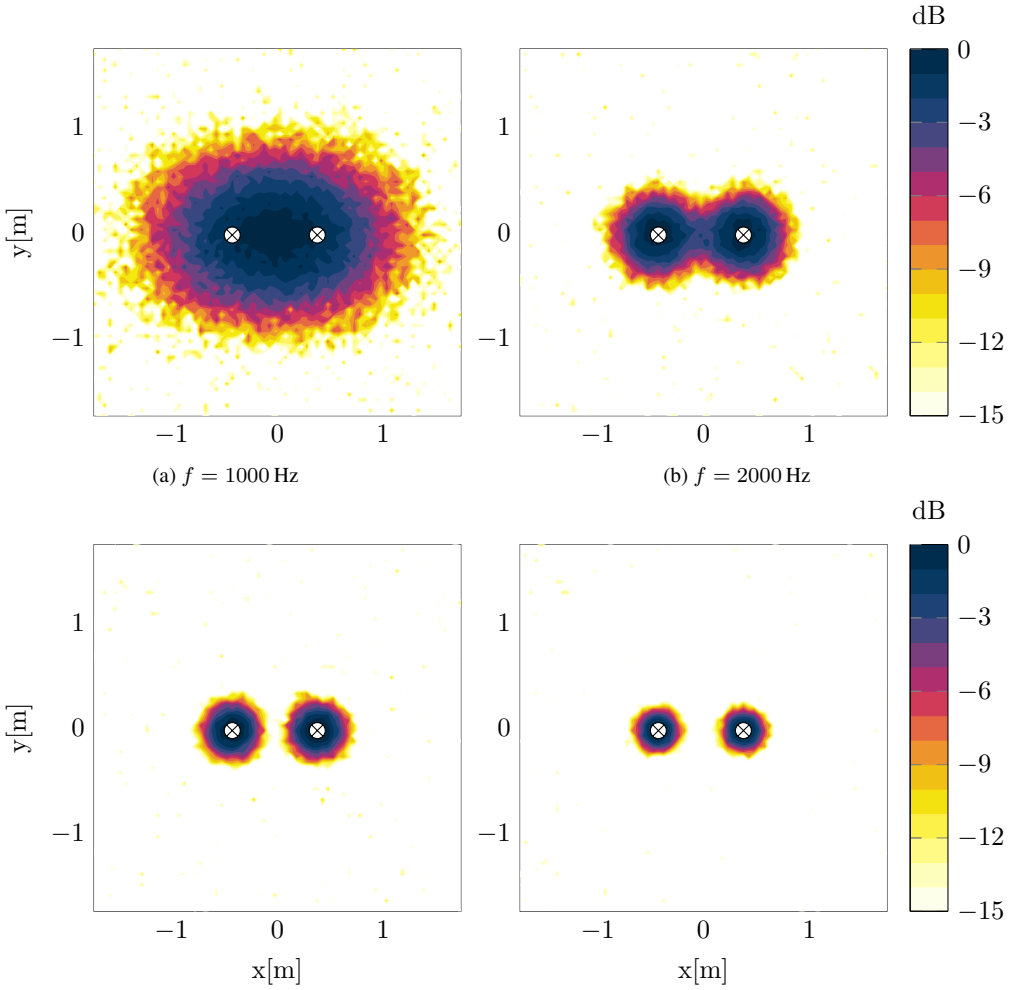
Keeping this first result of warm-starting in mind, the efficiency of using warm-start in connection with deconvolution of several subsequent frequency bands is investigated. Consider the beamformer maps of the two point source configuration introduced above for $f = 1000\,\text{Hz}$ to $4000\,\text{Hz}$. To simulate a more realistic situation, Gaussian white noise is added to the beamformer map to yield a signal-to-noise ratio of $15\,\text{dB}$. The beamformer maps are depicted in Figure 4.14. Recall that the resolution of the beamformer maps increases with frequency and assume therefore that the beamformer map at high frequency is closer to the true source distribution, which is two delta functions. That is, $\|\mathbf{x}^0 - \mathbf{x}^*\|$ gets smaller with increasing frequency. Then in line with that thought, efficiency could be improved by looping through subsequent frequency bands from high frequencies to low, supplying a realistic starting guess by means of the solution in the previous frequency band.

To test this idea, the beamformer map in the frequency band $f = 4000\,\text{Hz}$ and a zero starting guess is supplied to the deconvolution algorithms with the usual stopping criteria. The solution $\mathbf{x}^*$ at $f = 4000\,\text{Hz}$ is supplied to the deconvolution algorithms in the next frequency band $f = 4000\,\text{Hz} - \Delta f$, and so forth. The stopping criterion for subsequent bands is lowered to $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-2}$, due to the fact that a

smaller decrease in norm is expected since a good starting guess is supplied. The frequency spacing between subsequent bands Δf is chosen to $1000\,\text{Hz}$ in this particular case.

The deconvolution algorithms produce similar results at all frequencies, three examples of deconvolved maps computed without warm-staring are shown in Figure 4.15. As expected, the frequency band at $f = 1000\,\text{Hz}$ proves most difficult to reconstruct. When warm-starting is applied, efficiency is seen to improve greatly, Figure 4.17 summarize the number of iterations and computational time spend in each frequency band. No improvements are seen at $f = 4000\,\text{Hz}$, because this is the initial frequency band where the zero starting guess is supplied. Examples of the deconvolved maps with warm-start, equivalent to Figure 4.15, are shown in Figure 4.16. The reconstructions are almost perfect at all frequencies, which show that warm-starting is capable of providing both efficiency and resolution improvements.

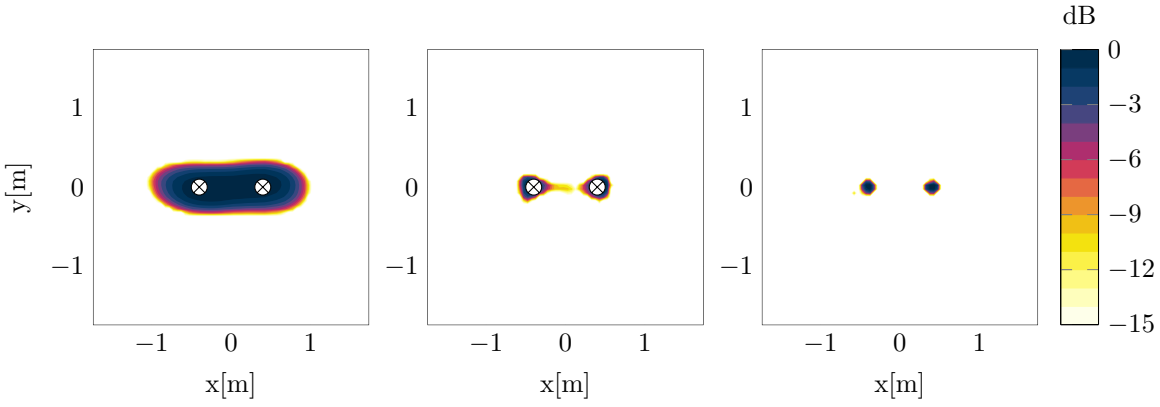
In this example, the source distribution is identical for all frequencies. The underlying assumption of using warm-start is, that the spectrum between subsequent frequency bands does not alter much. The extreme case was considered above, where the source position changed from one location to another between subsequent runs. The main conclusion was, that the deconvolution algorithms needed only a small number or iterations to adapt to the actual solution. In that case, efficiency would necessarily not improve as much as seen in Figure 4.17, since a number of adaptation iterations would be required. Additionally, applying warm-starting is only beneficial if deconvolution of multiple frequency bands are considered.



(a) $f = 1000\,\text{Hz}$ (b) $f = 2000\,\text{Hz}$

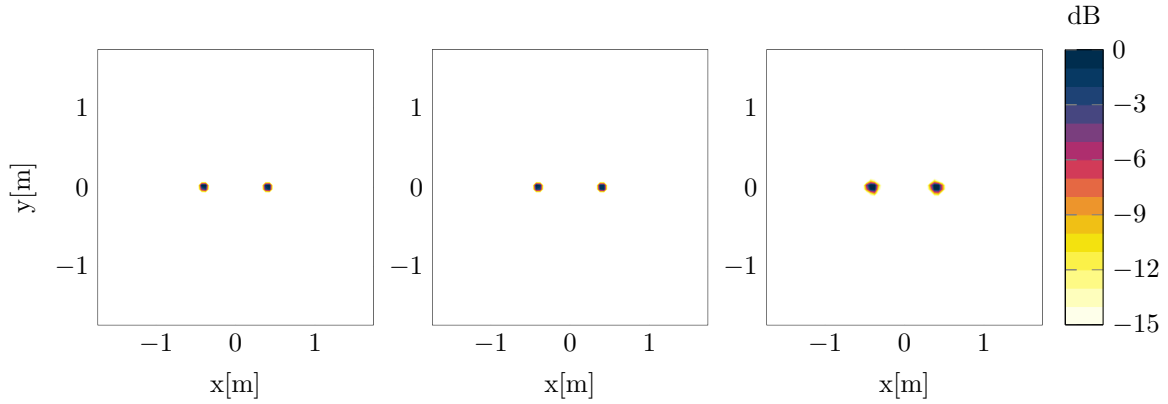
(c) $f = 3000\,\text{Hz}$ (d) $f = 4000\,\text{Hz}$

Figure 4.14: Beamformer maps at frequencies $f = 1000\,\text{Hz}$ to $4000\,\text{Hz}$, $\otimes$ designates the location of a point source.



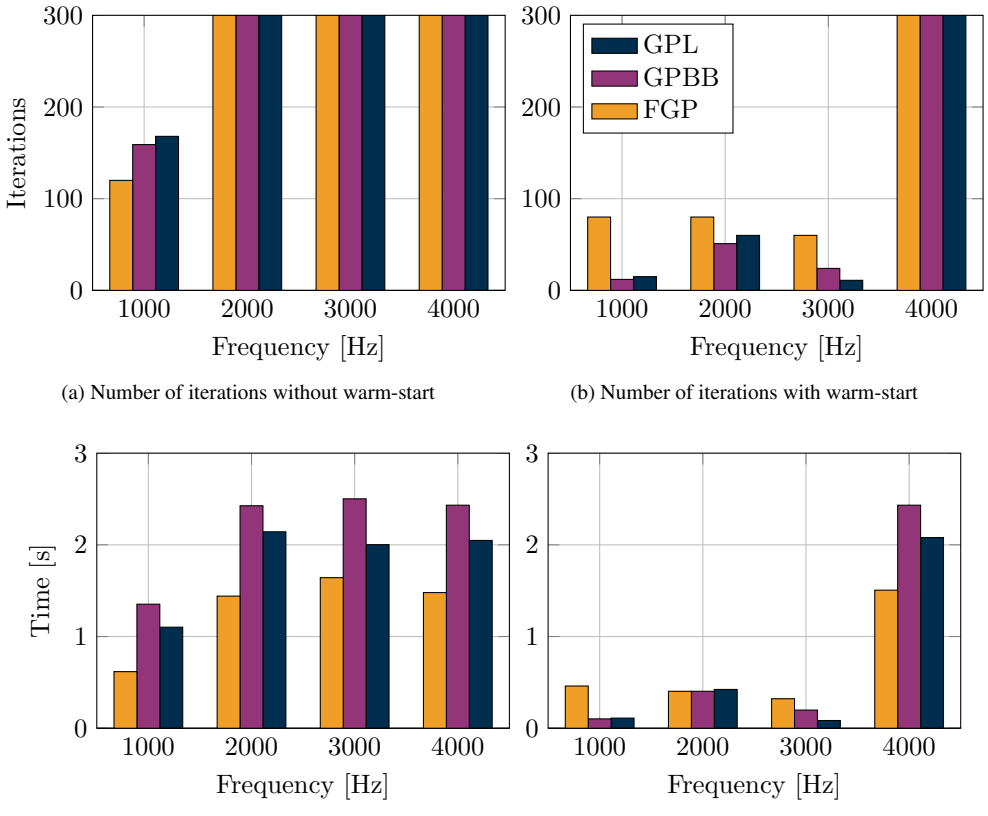
(a) FGP $f = 1000\,\text{Hz}$ (b) GPBB $f = 2000\,\text{Hz}$ (c) GPL $f = 3000\,\text{Hz}$

Figure 4.15: Deconvolved maps without warm-starting of two point source configuration at $z_0 = 3\,\text{m}$, separated by $\Delta x \approx 0.8\,\text{m}$. $\otimes$ designates the location of a point source, although it is omitted in (c).



(a) FGP $f = 1000\,\text{Hz}$ (b) GPBB $f = 2000\,\text{Hz}$ (c) GPL $f = 3000\,\text{Hz}$

Figure 4.16: Deconvolved maps with warm-starting of two point source configuration at $z_0 = 3\,\text{m}$, separated by $\Delta x \approx 0.8\,\text{m}$.



(a) Number of iterations without warm-start (b) Number of iterations with warm-start

(c) Computational time without warm-start (d) Computational time with warm-start

Figure 4.17: Comparison of number of iterations and computational run time for the deconvolution of a two point source configuration at $z_0 = 3\,\text{m}$, separated by $\Delta x \approx 0.8\,\text{m}$. A sequence of 4 frequency bands in deconvolved, where warm-starting is applied to the one.

4.4 Multiple sources

Turning to more complicated source distributions, consider 8 point sources arranged in a circle at a distance $z_0 = 5\,\mathrm{m}$ with varying source strengths ranging from $0\,\mathrm{dB}$ to $-7\,\mathrm{dB}$ in normalized level, depicted in Figure 4.18. The circular distribution assures that the point-spread function is almost identical for all point sources positions. Or rather, that the model–data mismatch is equal for all point sources in the map. The right most point source is assigned with maximum strength, whence normalized to $0\,\mathrm{dB}$ and the subsequent points sources are assigned with decreasing strength in intervals of $-1\,\mathrm{dB}$ in counterclockwise ordering.

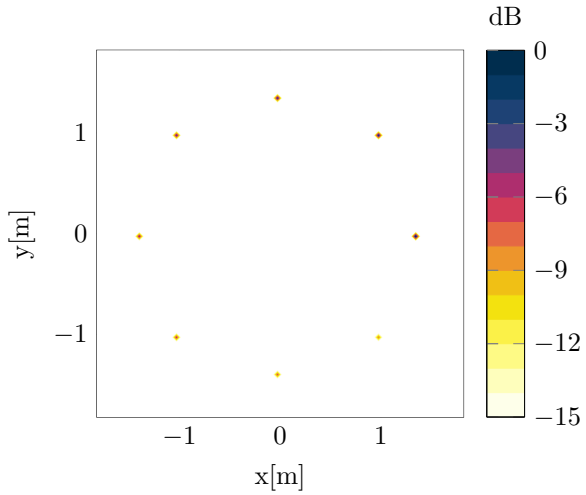


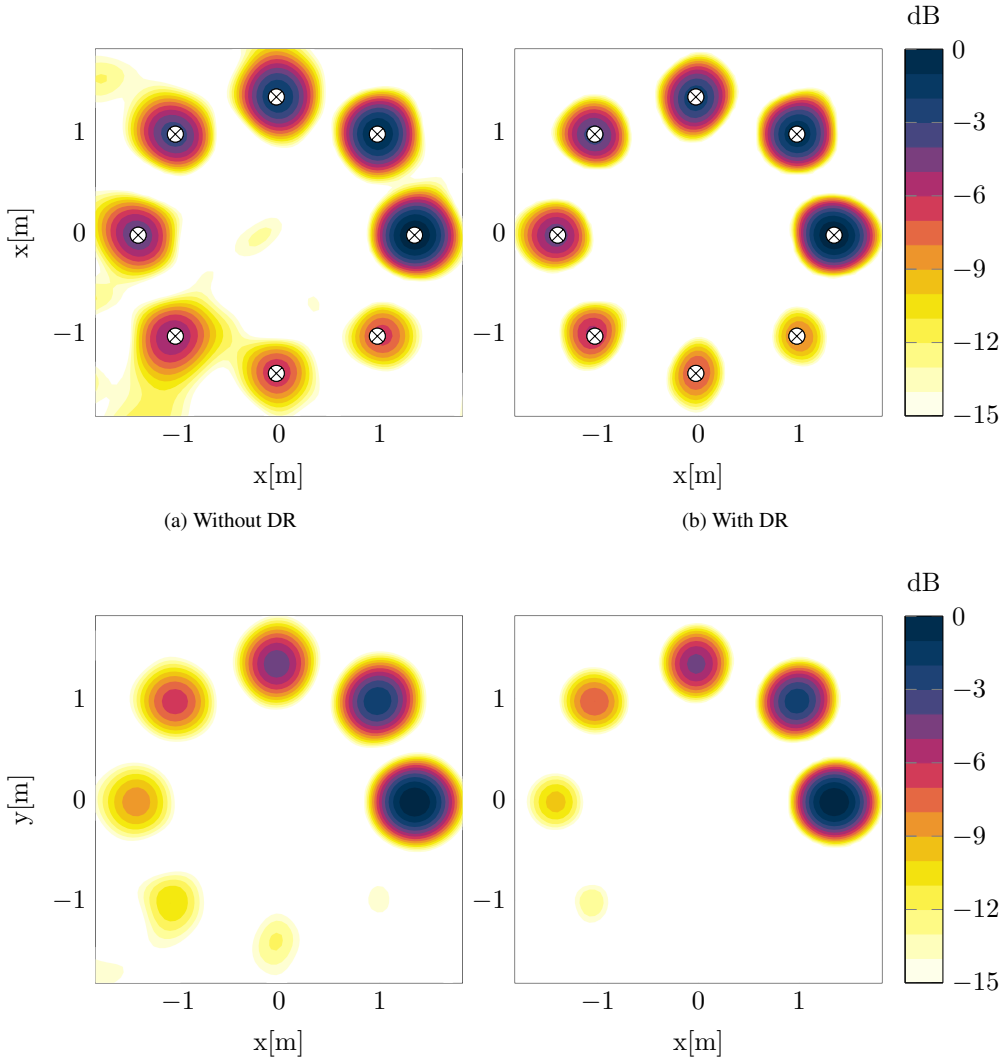
Figure 4.18: Point source distribution

The beamformer map is, as usual, constructed from a synthetic cross-spectrum matrix, given in Equation (2.26). However, to simulate more realistic conditions, Gaussian white noise is added to the cross-spectrum matrix before calculation of the beamformer map. This agrees well with actual measurement conditions, where the cross-spectrum matrix is derived from captured time data that inevitably is contaminated with noise.

4.4.1 Diagonal Removal

The beamformer maps considered so far have all been derived using diagonal removal (DR) to increase the signal-to-noise ratio by removing self-noise. One possible side effect is however, that low-level source contributions might also be removed by applying DR. To assess the effect of applying DR, consider beamformer maps with $\phi = 20^\circ$, calculated from a cross-spectrum matrix with added Gaussian white noise at a signal-to-noise ratio of $0\,\mathrm{dB}$, depicted in the top panel of Figure 4.19. All point sources are visible in the beamformer maps and in this case DR has no effect on the low level source at $-7\,\mathrm{dB}$. Increasing the dynamic range of the source strengths to $0\,\mathrm{dB}$ to $-14\,\mathrm{dB}$ and calculating the beamformer maps, this time without added noise to the cross-spectrum matrix, the lower panel in Figure 4.19 shows, that the level of the two point sources, at $-14\,\mathrm{dB}$ and $-12\,\mathrm{dB}$, is decreased to a level below $-15\,\mathrm{dB}$.

Peak levels at the point source positions are given in Table 4.4 with and without DR and at the two dynamic ranges considered. The low level sources at $-14\,\mathrm{dB}$ and $-12\,\mathrm{dB}$ are decreased in level to $-20\,\mathrm{dB}$ and $-14\,\mathrm{dB}$ respectively.



(a) Without DR

(b) With DR

(c) Without DR, high dynamic range

(d) With DR, high dynamic range

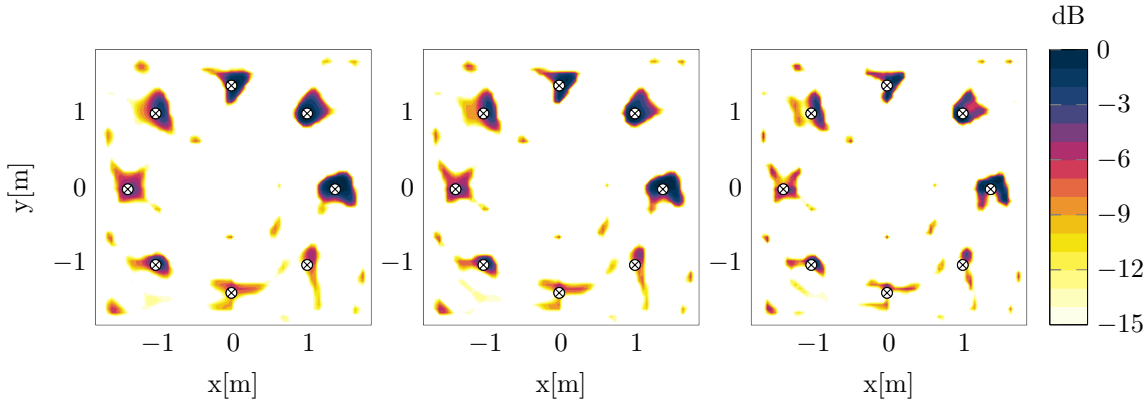
Figure 4.19: Circular source distribution $f = 4000\,\text{Hz}$. *Top panel:* Beamformer maps calculated from a noisy cross-spectrum matrix with signal-to-noise ratio of 0 dB. *Bottom panel:* Beamformer maps with higher dynamic range, calculated from cross-spectrum matrix without added noise.

	Normalized level [dB]							
Without DR	-7	-6	-5	-4	-3	-2	-1	0
With DR	-7.91	-6.59	-5.44	-4.39	-3.26	-2.13	-1.07	0
Without DR	-14	-12	-10	-8	-6	-4	-2	0
With DR	-20.0	-14.4	-11.3	-8.92	-6.56	-4.25	-2.11	0

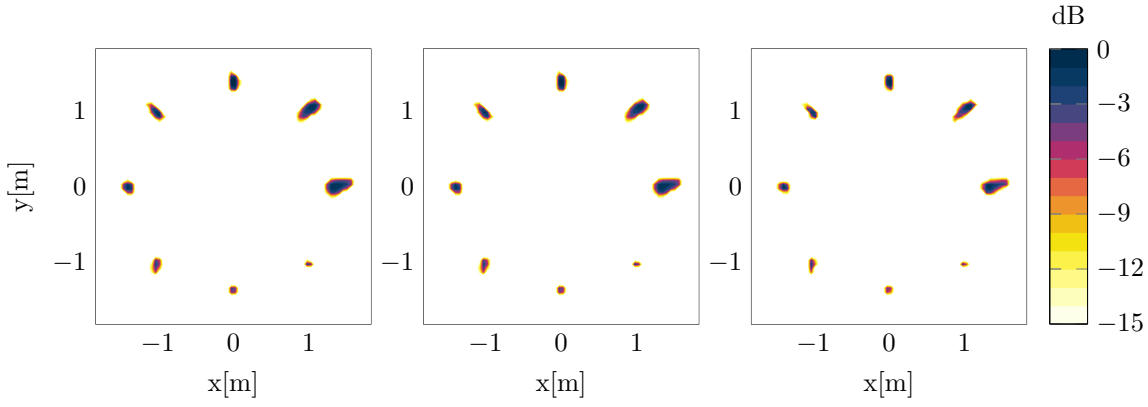
Table 4.4: Level of beamformer map at point source positions.

The deconvolution algorithms are applied to the two cases with the usual stopping criteria $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ and a maximum of 300 iterations. The deconvolved maps at dynamic ranges -7 dB and -14 dB are shown in figures 4.20 and 4.21 respectively. In the former case, all points are resolved and DR has a clear effect on the quality of the deconvolved maps. The latter case however, only points at a level above -10 dB are resolved. The usual performance measures $\|\nabla f(\mathbf{x}^k)_+\|/\|\nabla f(\mathbf{x}^0)_+\|$ and $\left(f(\mathbf{x}^k) - f(\mathbf{x}^*)\right)/\|\mathbf{x}^0 - \mathbf{x}^*\|^2$ are shown in figures 4.22 and 4.23 respectively. It is seen that the initial convergence rate of the case without DR is faster compared to the case with DR. Some degree of stagnation seems to arise after about 20 iterations without DR, while the case with DR exhibits a more stable rate of convergence. The deconvolution algorithms exhibit similar convergence rates without DR, while GPL attains a superior rate with DR.

From this investigation it is clear, that both efficiency of the deconvolution algorithms and spatial resolution of the maps are greatly improved with diagonal removal is applied. Source distributions with a high dynamic range might suffer from loss of low-level noise contributions. This is a trade-off that should be assessed by the priority and objective of a given measurement condition.

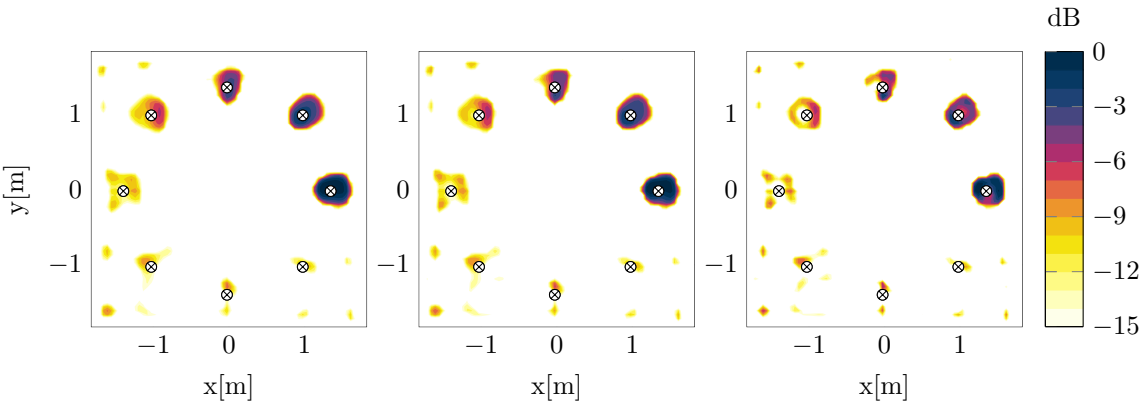


(a) GPL, without DR (b) GPBB, without DR (c) FGP, without DR

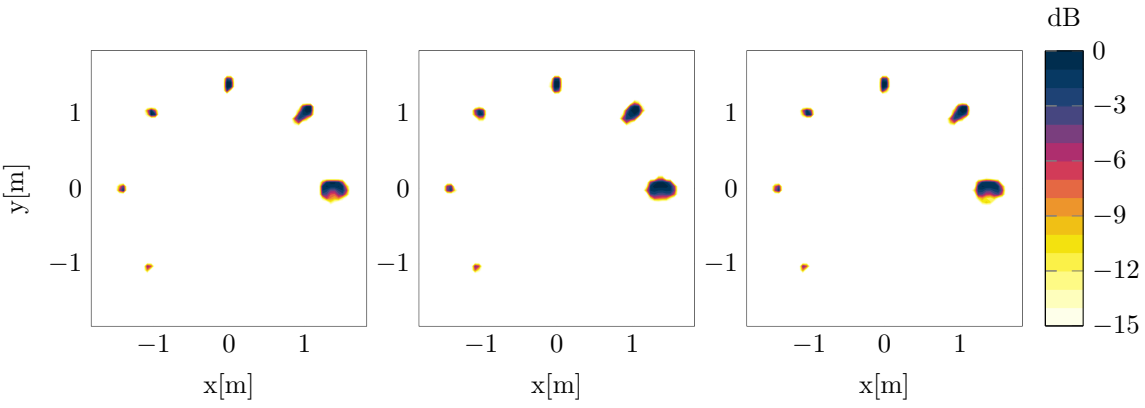


(d) GPL, with DR (e) GPBB, with DR (f) FGP, with DR

Figure 4.20: Deconvolved maps with -7 dB dynamic range in source distribution.

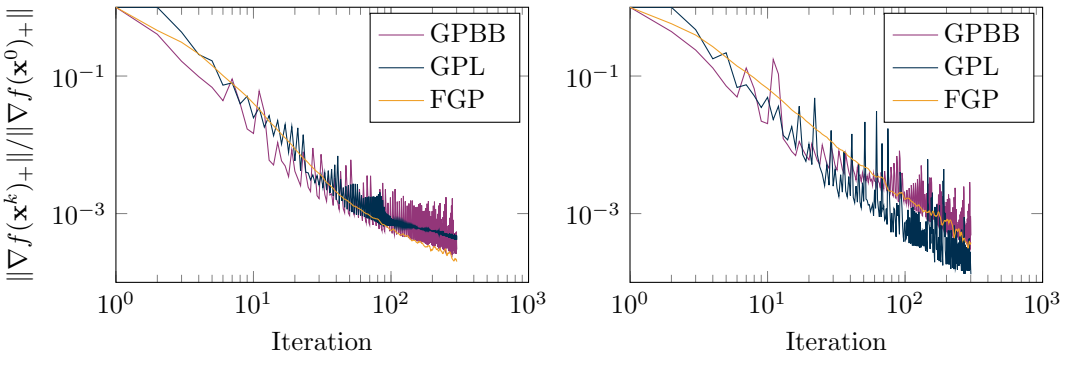


(a) GPL, without DR (b) GPBB, without DR (c) FGP, without DR



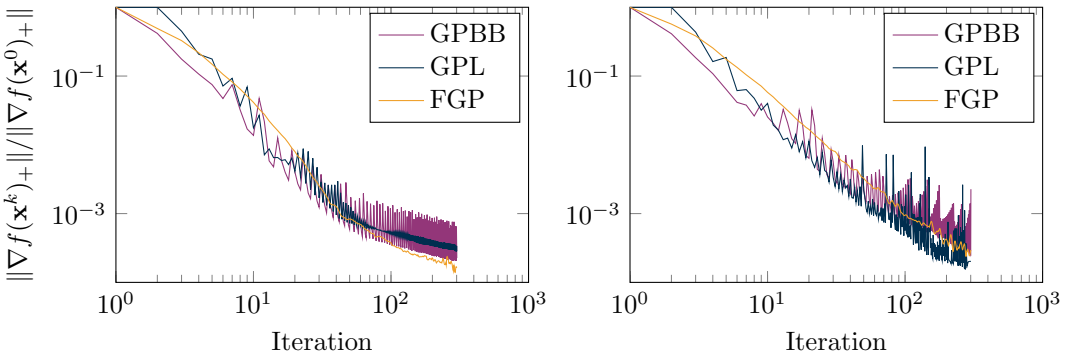
(d) GPL, with DR (e) GPBB, with DR (f) FGP, with DR

Figure 4.21: Deconvolved maps with -14 dB dynamic range in source distribution.



(a) Without DR

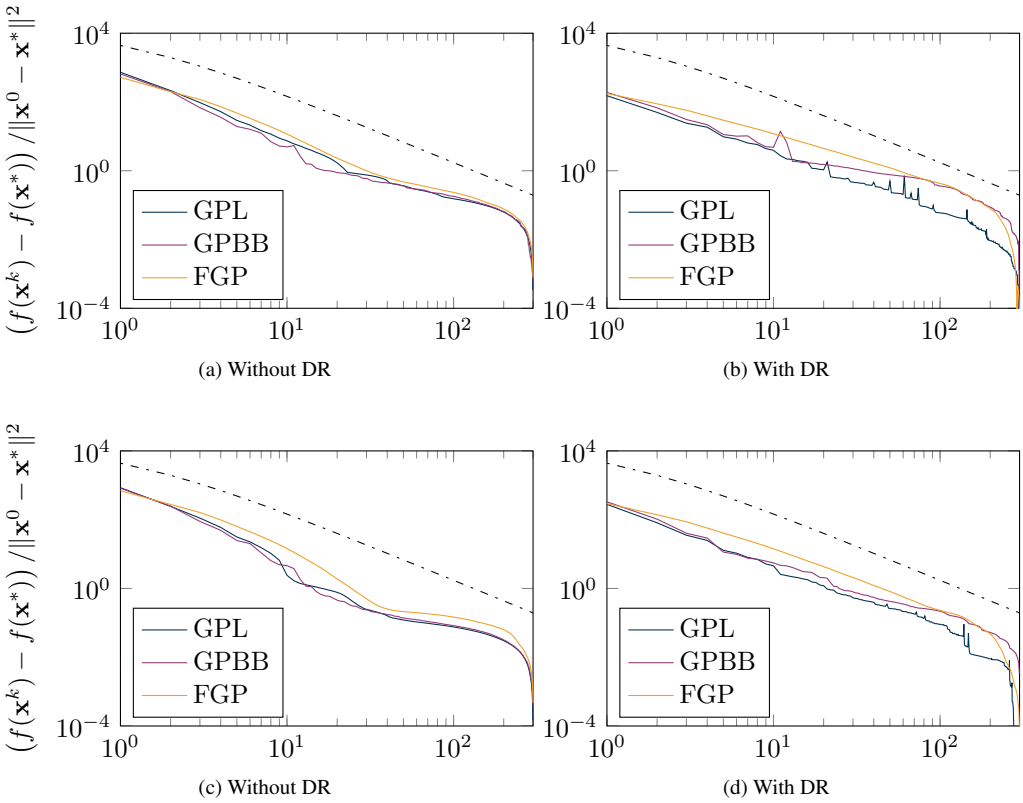
(b) With DR



(c) Without DR

(d) With DR

Figure 4.22: Decrease in norm of projected gradient at dynamic range $-7\,\mathrm{dB}$ (top panel) and $-14\,\mathrm{dB}$ (bottom panel)



(a) Without DR (b) With DR

(c) Without DR (d) With DR

Figure 4.23: Decrease in objective function value at dynamic range -7 dB (top panel) and -14 dB (bottom panel). The dash-dotted line represents the upper complexity bound of FGP $\mathcal{O}(1/k^2)$.

5 Experimental Study

The simulation study provided an insight to the performance and achievable reconstruction quality of deconvolution algorithms. Now, this insight is put to practical use by testing the algorithms on experimental data.

5.1 Measurement Setup

The measurements considered in the following are conducted in the large anechoic chamber at DTU. This $1000\,\mathrm{m}^3$ room is used to imitate acoustical free space with a lower limiting frequency of about $50\,\mathrm{Hz}$. Analogous to the simulation study are monopole point sources considered: Two loudspeakers, Omnisource Type 4295, separated by $0.70\,\mathrm{m}$ are placed directly in front of an irregular 60 channel microphone array, B&K P60 D100. The speakers are optimized to model the radiation characteristics of a monopole. Each speaker is connected to a noise generator that produce Gaussian white noise. The gain is adjusted such that the sound pressure level produced by each speaker is approximately $80\,\mathrm{dB}$ re $20\,\mu\mathrm{Pa}$, $1\,\mathrm{m}$ in front of the microphone array. The two speakers thus deliver incoherent random noise with monopole-like radiation characteristics. The data acquisition is controlled outside the anechoic chamber by a laptop with PULSE Labshop software connected to a B&K front-end, which handles the digital signal processing. A schematic of the experimental setup is shown in Figure 5.1 and photos are shown in Figure 5.2. A full list of measurement equipment is given in Appendix B.

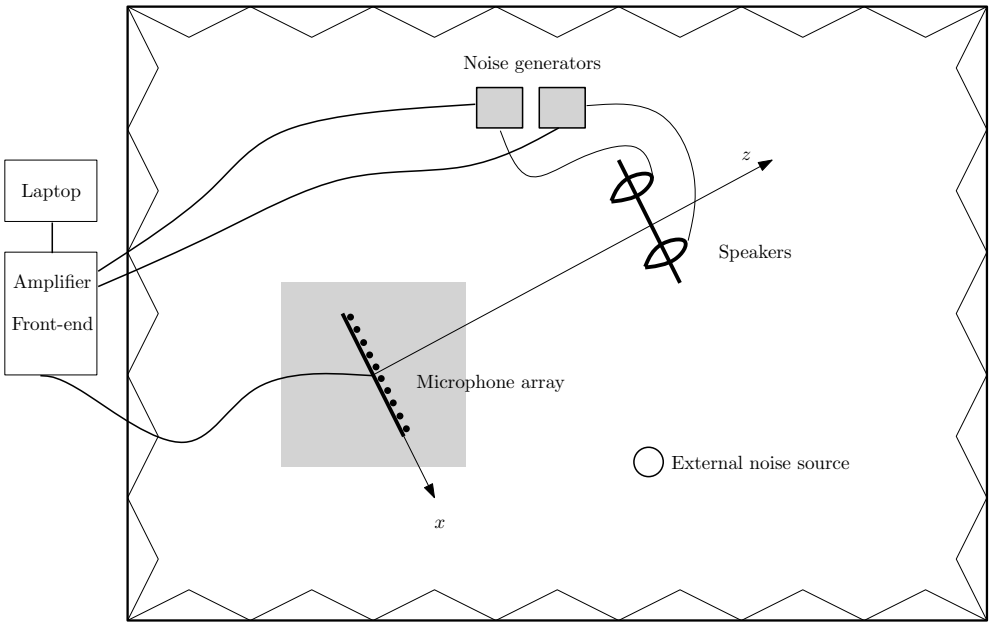
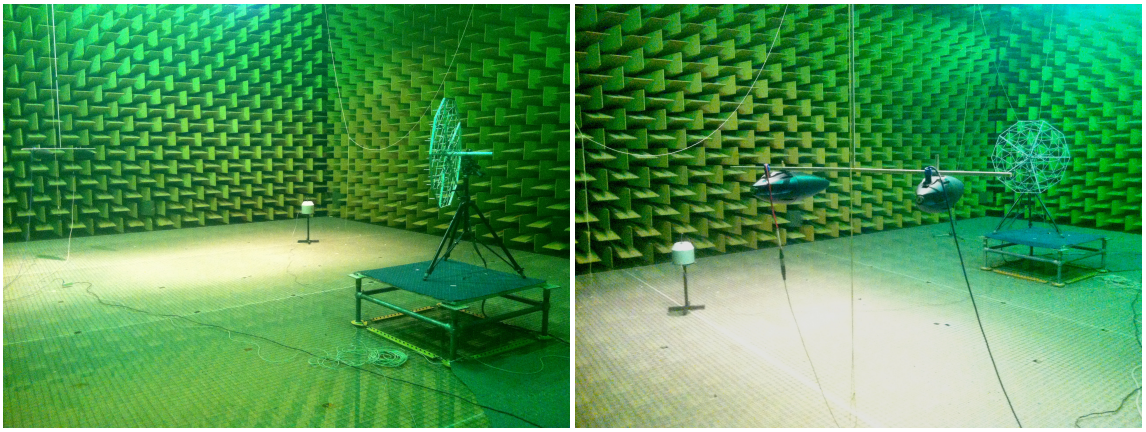


Figure 5.1: Experimental setup in anechoic chamber. View from above.



(a) View from side (b) Behind speakers

Figure 5.2: Experimental setup in anechoic chamber. In photos, the microphone array, two speakers and external noise source are seen.

The measurement setup is validated to ensure that the captured data can be trusted. The validation is done by checking the calibration of the microphones, estimating the coherence between the output of the noise generators and the input from the microphones, and calculating the autospectrum of the speakers and noise floor.

The 60 $1/4$" microphones in the microphone array are automatically detected by the front-end and individual sensitivities are set automatically from a database. This auto-calibration is checked on a sample of 10 microphones. The calibration is done by using a B&K Sound Level Calibrator Type 4231, that delivers a 1 kHz tone at 94 dB re 20 µPa. A FFT-analysis is carried out in PULSE Labshop with a flat top window, 1 Hz spectral resolution, no overlapping and 10 averages of 1 s segments. The calculated sound pressure level at 1 kHz is noted and the calibration process is repeated for the remaining 9 microphones, chosen at random over the array.

Microphone	1	2	17	35	36	37	41	44	49	60	Mean
dB SPL	93.1	93.4	93.0	93.2	93.3	93.3	93.3	93.1	93.2	93.1	93.2
Deviation dB	0.9	0.6	1.0	0.8	0.7	0.7	0.9	0.8	0.9	0.9	0.8

The results show that the 10 microphones on average measure 93.2 dB $\pm$ 0.2 dB re 20 µPa, which is 0.8 dB below the reference level of the calibrator. This deviation is acceptable for a beamforming application and due to the small variance $\pm$ 0.2 dB, the microphone calibration is trustworthy.

The coherence is defined as the ratio between the absolute squared cross-spectrum and the product of the autospectra,

$$\gamma_{xy}^2(\omega) = \frac{|S_{xy}(\omega)|^2}{S_{xx}(\omega)S_{yy}(\omega)}. \tag{5.1}$$

where $0 \leq \gamma_{xy}^2 \leq 1$. The coherence provides a measure of the overall quality of the measurement chain. Damaged cables, non-linearities in amplifier or speaker, clipping of signals or extraneous noise sources would all result in coherence less than unity. The coherence is estimated, with one speaker at a time, by making 20 s recordings of each speaker being fed with random noise. The speakers are located $z_0 = 565$ cm from the array. The recorded time data for all microphones and direct output from the noise generators are

saved and exported to MATLAB. Cross-spectral matrices are computed from the 20 s recordings, split up in time segments of 1 s (16384 samples) multiplied with a Hanning window and using 66% overlap between time segments. This results in a spectral averaging over 57 segments. With a resulting spectral resolution of 1 Hz, the cross-spectrum matrix attain dimensions of $60 \times 60 \times 8192$. The spectral averaging procedure described here is used for all subsequent results.

The coherence is averaged over all microphones and the variance is checked to reveal any outliers. In addition, an external noise source is placed in the room to test the coherence in the presence of different noise levels.

A word on notation: The external noise source is usually used as a sound power reference source, why the settings on it are given in sound power level re 1 pW. These settings are referred to in the following, although the sound power is of no interest in this measurement scenario.

The coherence as function of frequency is shown in Figure 5.3 for both loudspeakers. The coherence is practically unity when no external noise source is present and becomes increasingly degraded for higher noise levels. This ensures that the measurement chain is in good condition when external noise sources are not present.

The autospectra of the speakers are retrieved in the same manner as the coherence – in fact the diagonal of the cross-spectral matrix represents the autospectrum of each microphone at that particular frequency. The autospectra are shown in Figure 5.4a. The resonances in the autospectra are due to the construction of the loudspeaker tubes, that are designed to have uniform directionality rather than a flat frequency response.

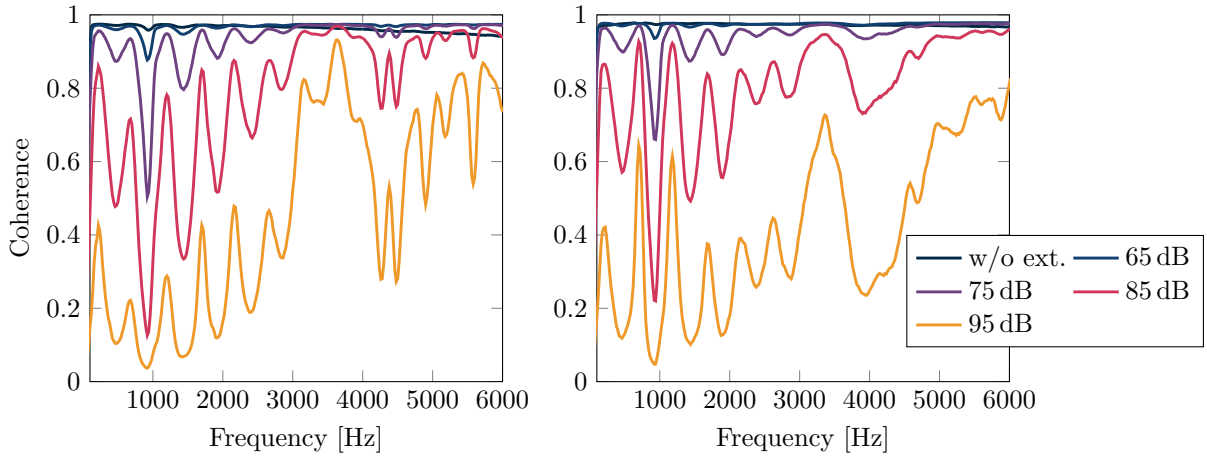
The background noise is measured without any active sound sources and is approximately 35 dB re 20µPa in overall sound pressure level. The noise floor, in presence of an external noise source, is measured at different noise levels, represented by the settings on the sound power reference source: 65 dB re 1pW to 95 dB re 1pW. The derived autospectra of the background noise and noise floor are shown in Figure 5.4b and the signal-to-noise ratios (SNR) for each source are summarized in Table 5.1.

SNR [dB]	Source 1	Source 2
Background noise level	27.8	31.8
Ext. noise source setting:		
65 dB re 1pW	23.7	27.7
75 dB re 1pW	17.8	21.8
85 dB re 1pW	8.2	12.5
95 dB re 1pW	-1.8	2.2

Table 5.1: Signal-to-noise ratio at $z_0 = 5.65$ m. The noise levels are referred to by the settings on a sound power reference source, hence the reference in dB re 1pW. The SNR of Source 1 and 2 are given in dB.

5.2 Effects of shift-variant point-spread functions

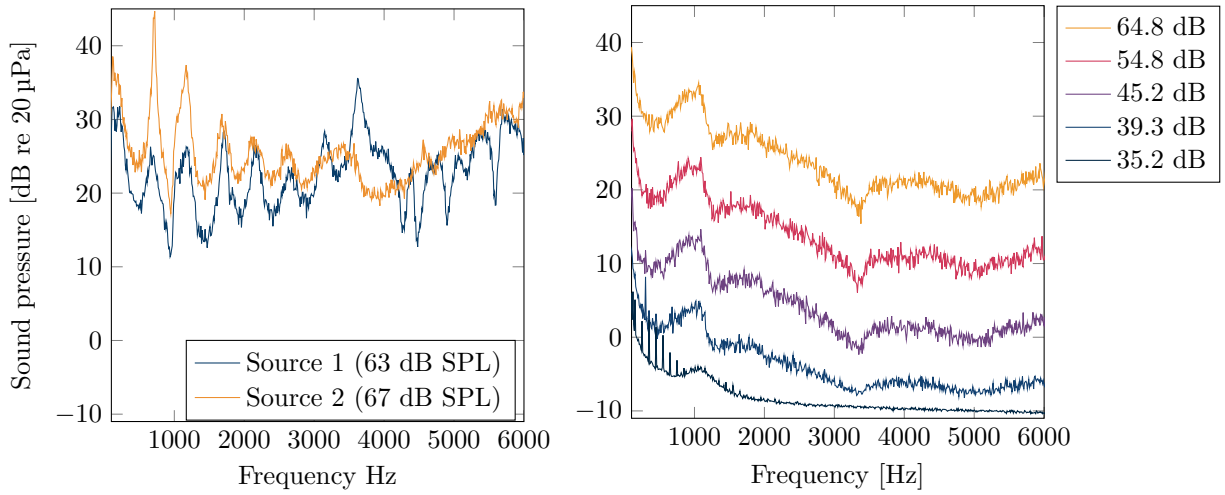
The effects of shift-variant point-spread functions was shown to worsen efficiency of deconvolution algorithms and limit the spatial resolution in the case of simulated point sources. In this section, the effect is investigated with experimental data, by considering two speakers separated by $\Delta x = 0.7$ m at two different measurement distances: $z_0 = 1.05$ m and $z_0 = 5.65$ m. A these measurement distances, the point-spread functions can be assumed to be shift-variant and shift-invariant by proper choice of beamformer opening angle ϕ. With a measurement area of interest of $d_0 = 1.4$ m, the opening angles are respectively $\phi = 35^\circ$



(a) Source 1

(b) Source 2

Figure 5.3: Average coherence $\gamma^2_{xy}(\omega)$ over all microphones of Source 1 and 2 at $z_0 = 5.65\,\text{m}$ with and without external noise source. The levels are given in dB re 1pW, related to the settings on the external noise source.



(a) Average autospectra over all microphones of two Omnisource Type 4295.

(b) Average autospectra over all microphones of background noise and external noise source.

Figure 5.4: *Left panel:* Autospectra of two sound sources at 5.65 m. The overall sound pressure level is 63 dB SPL and 67 dB SPL for Source 1 and 2 respectively. *Right panel:* Autospectra of background noise and external noise. Levels in legend are given as overall sound pressure level dB re 20µPa.

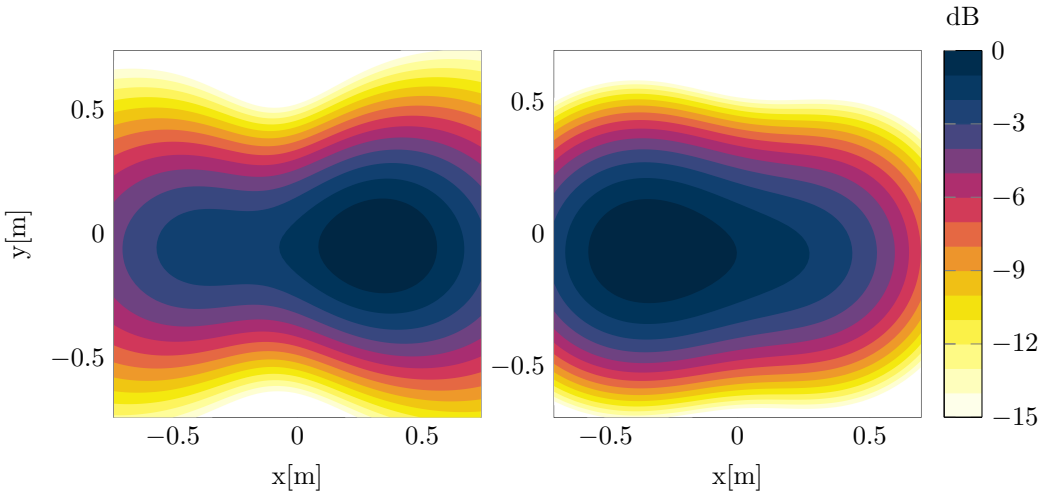
and $\phi = 7°$. The beamformer resolution limit $R/\Delta x = 1$ occur at frequency

$$R/\Delta x = 1 = \frac{1.22 z_0 c}{\Delta x \cos^3(\theta_{\text{sep}}) f}, \quad \theta_{\text{sep}} = \arctan\left(\frac{\Delta x}{2 z_0}\right)$$

$$f = \frac{1.22 z_0 c}{\Delta x \cos^3(\theta_{\text{sep}})} = \begin{cases} 697\,\text{Hz}, z_0 = 1.05\,\text{m} \\ 3391\,\text{Hz}, z_0 = 5.65\,\text{m}, \end{cases}$$

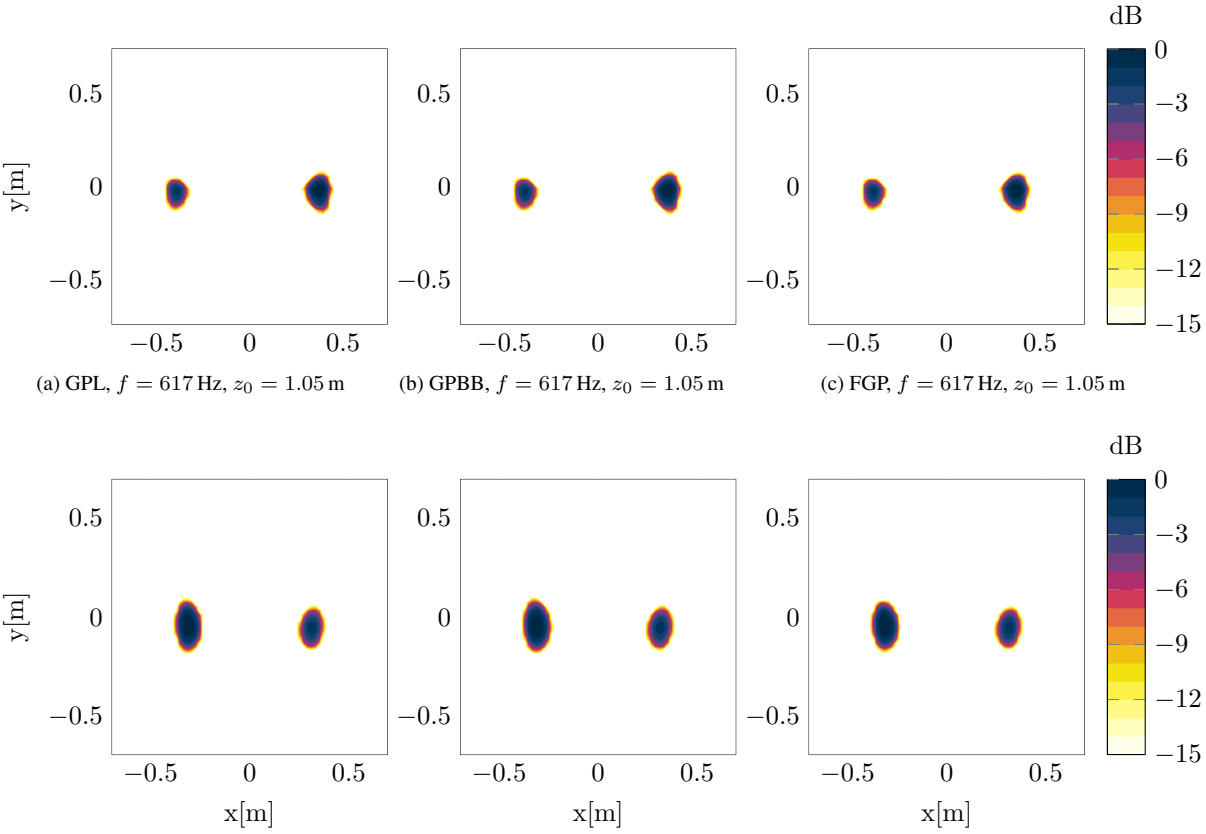
where θ_{sep} is the angular separation of the point sources. The overall sound pressure level at measurement distance $z_0 = 1.05\,\text{m}$ is 80 dB SPL and 68 dB SPL at $z_0 = 5.65\,\text{m}$. With no external noise source, this setup produce signal-to-noise ratios of 45 dB and 33 dB, respectively.

Consider first, frequencies at which $R/\Delta x > 1$. For $z_0 = 1.05\,\text{m}$, $f = 617\,\text{Hz}$ is considered and for $z_0 = 5.65\,\text{m}$, $f = 2951\,\text{Hz}$ is considered. The beamformer resolution is $R/\Delta x = 1.13$ and $R/\Delta x = 1.15$ respectively. The two beamformer maps are shown in Figure 5.5. The two sources cannot be resolved by the beamformer at these frequencies, thus deconvolution can be applied to increase spatial resolution. The deconvolution algorithms are given stopping criteria $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ and a maximum of 1000 iterations. The deconvolved maps, shown in Figure 5.6, obtain the same overall quality of reconstruction. The two sources are well separated in both cases, although at $z_0 = 5.65\,\text{m}$ the resolution is slightly poorer.



(a) $f = 617\,\text{Hz}$, $z_0 = 1.05\,\text{m}$ (b) $f = 2951\,\text{Hz}$, $z_0 = 5.65\,\text{m}$

Figure 5.5: Beamformer maps of two point sources separated by $\Delta x = 0.7\,\text{m}$ at two different distances. The map on the right has a predominantly shift-invariant point-spread function, while this is not the case for the map on the left. The beamformer resolutions are $R/\Delta x = 1.13$ in the left panel and $R/\Delta x = 1.15$ in the right.

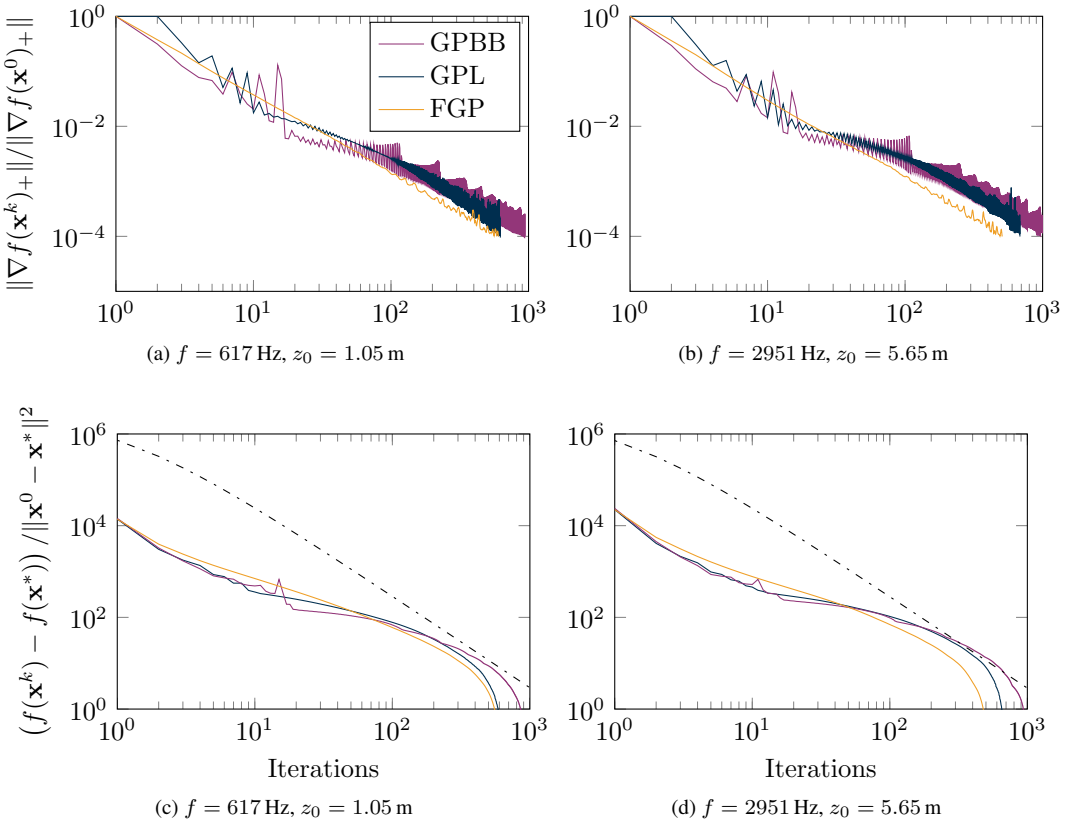


(a) GPL, $f = 617\,\text{Hz}$, $z_0 = 1.05\,\text{m}$ (b) GPBB, $f = 617\,\text{Hz}$, $z_0 = 1.05\,\text{m}$ (c) FGP, $f = 617\,\text{Hz}$, $z_0 = 1.05\,\text{m}$

(d) GPL, $f = 2951\,\text{Hz}$, $z_0 = 5.65\,\text{m}$ (e) GPBB, $f = 2951\,\text{Hz}$, $z_0 = 5.65\,\text{m}$ (f) FGP, $f = 2951\,\text{Hz}$, $z_0 = 5.65\,\text{m}$

Figure 5.6: Deconvolved maps of two point sources separated by $\Delta x = 0.7\,\text{m}$ at two different distances.

Despite the same overall reconstruction quality, the deconvolution algorithms perform differently in terms of efficiency. The performance measures, $\|\nabla f(\mathbf{x}^k)_+\|/\|\nabla f(\mathbf{x}^0)_+\|$ and $\left(f(\mathbf{x}^k) - f(\mathbf{x}^*)\right)/\|\mathbf{x}^0 - \mathbf{x}^*\|^2$, shown in Figure 5.7, reveal that FGP attains a superior rate of convergence and a faster decrease in the norm of the projected gradient. In particular for $z_0 = 5.65\,\text{m}$, where the point-spread function is predominantly shift-invariant. This result agrees well with the simulation study in Section 4.1. The computational time of the deconvolution algorithms are given in tables 5.2a and 5.2b.



(a) $f = 617\,\text{Hz}$, $z_0 = 1.05\,\text{m}$

(b) $f = 2951\,\text{Hz}$, $z_0 = 5.65\,\text{m}$

(c) $f = 617\,\text{Hz}$, $z_0 = 1.05\,\text{m}$

(d) $f = 2951\,\text{Hz}$, $z_0 = 5.65\,\text{m}$

Figure 5.7: Norm of the projected gradient and decrease in objective function value for two point sources separated by $\Delta x = 0.7\,\text{m}$ at two different distances. The dash-dotted line represent the upper complexity bound of FGP $\mathcal{O}(1/k^2)$.

Method	Iterations k	Time $[s]$
GPL	623	4.3
GPBB	1000	7.5
FGP	620	3.1

(a) $f = 617\,\text{Hz}$ at $z_0 = 1.05\,\text{m}$.

Method	Iterations k	Time $[s]$
GPL	683	4.6
GPBB	1000	7.9
FGP	569	2.7

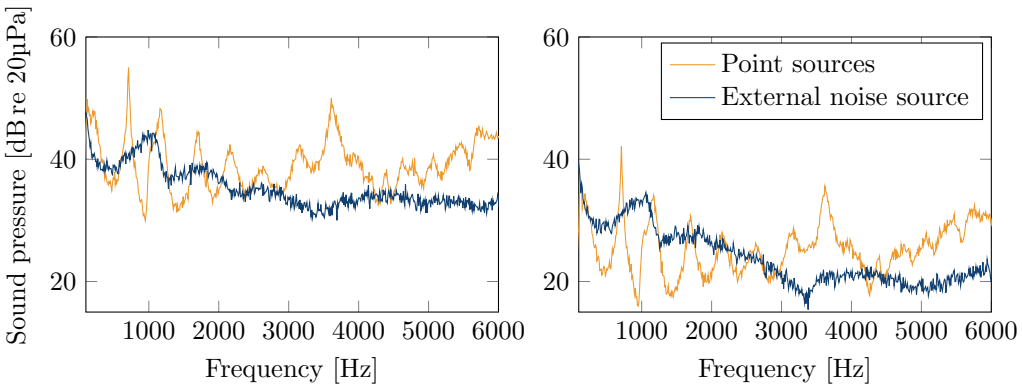
(b) $f = 2951\,\text{Hz}$ at $z_0 = 5.65\,\text{m}$.

Table 5.2: Number of iterations and computational run time.

5.2.1 With external noise source

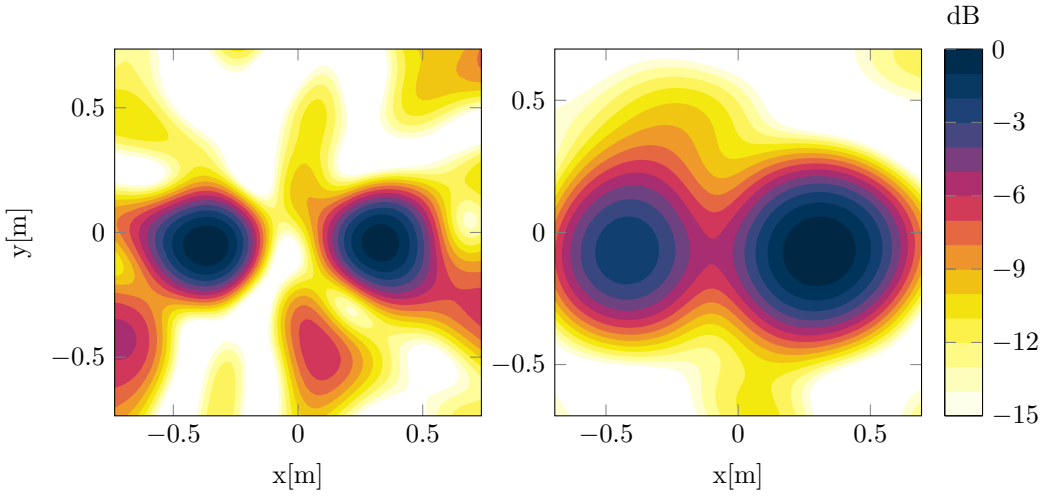
In an attempt to challenge the deconvolution algorithms, a strong external noise source is added to the two point source configuration. The autospectra of the external noise source and two sources are given in Figure 5.8. The overall sound pressure level at $1.05\,\text{m}$ is $75.1\,\text{dB}$ SPL and $64.8\,\text{dB}$ SPL at $5.65\,\text{m}$. The

resulting overall signal-to-noise ratios are 4.9 dB and 3.2 dB respectively. The considered beamformer maps are calculated without diagonal removal at $f = 1448$ Hz for $z_0 = 1.05$ m and $f = 4455$ Hz for $z_0 = 5.65$ m. The signal-to-noise ratio at the considered frequencies are -5 dB at $f = 1448$ Hz and 0 dB at $f = 4455$ Hz. The beamformer maps are depicted in Figure 5.9 and their resolution at these frequencies are $R/\Delta x = 0.48$ and $R/\Delta x = 0.76$ respectively.



(a) $z_0 = 1.05$ dB (b) $z_0 = 5.65$ dB

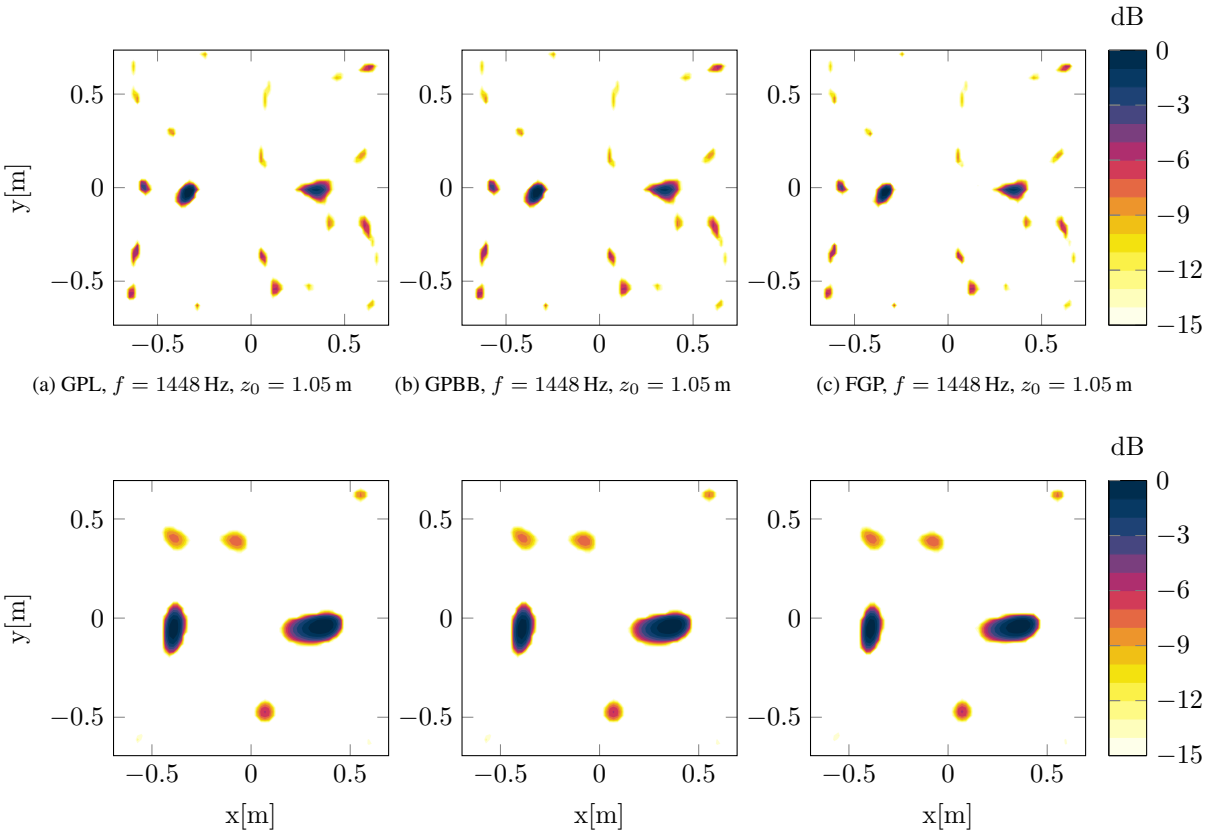
Figure 5.8: Individual autospectra of the two point sources and external noise source. The overall signal-to-noise ratios are 4.9 dB and 3.2 dB respectively.



(a) $f = 1448$ Hz, $z_0 = 1.05$ m, $R/\Delta x = 0.48$ (b) $f = 4455$ Hz, $z_0 = 5.65$ m, $R/\Delta x = 0.76$

Figure 5.9: Beamformer maps computed without diagonal removal. An external noise source is present, which yields a SNR of -5 dB (left) and 0 dB (right).

The deconvolved maps are shown in Figure 5.10 and performance measures in Figure 5.11. The overall quality of the reconstruction is worse than previously seen, this is due to the added external noise and especially the omission of diagonal removal. Yet, the strongest responses in the deconvolved maps arise from the two sources. Keeping in mind that the signal-to-noise ratios are below 0 dB, the results are quite extraordinary. This demonstrates quite well, the beamformer's ability to filter out unwanted noise by focusing only in the area of interest. The performance measures, $\|\nabla f(\mathbf{x}^k)_+\|/\|\nabla f(\mathbf{x}^0)_+\|$ and $\left(f(\mathbf{x}^k) - f(\mathbf{x}^*)\right)/\|\mathbf{x}^0 - \mathbf{x}^*\|^2$ show specifically two interesting things: First, the convergence rates of the deconvolution methods are quite slow, which suggests that the deconvolution problem is quite ill-posed, and second, GPL and GPBB are



(a) GPL, $f = 1448\,\text{Hz}$, $z_0 = 1.05\,\text{m}$ (b) GPBB, $f = 1448\,\text{Hz}$, $z_0 = 1.05\,\text{m}$ (c) FGP, $f = 1448\,\text{Hz}$, $z_0 = 1.05\,\text{m}$

(d) GPL, $f = 4455\,\text{Hz}$, $z_0 = 5.65\,\text{m}$ (e) GPBB, $f = 4455\,\text{Hz}$, $z_0 = 5.65\,\text{m}$ (f) FGP, $f = 4455\,\text{Hz}$, $z_0 = 5.65\,\text{m}$

Figure 5.10: Deconvolved maps of two point sources separated by $\Delta x = 0.7\,\text{m}$ at two different distances based on beamformer maps without DR. An external noise source is present, which yields a SNR of $-5\,\text{dB}$ in the top panel and $0\,\text{dB}$ in the bottom panel.

seen to cross the upper complexity bound of FGP. Since FGP is guaranteed to stay below this limit, it will always perform better than GPL and GPBB, if they at a point cross that bound. The superior efficiency is also seen from the number of iterations and computational run time in Table 5.3. GPL use the maximum number of iterations in both cases, while GPBB and FGP require fewer at $z_0 = 5.65\,\text{m}$, 30% and 20% fewer respectively. As previously stated, this is likely to be caused by a smaller model–data misfit, since the assumption of shift-invariance of the point-spread function is more correct at $z_0 = 5.65\,\text{m}$. Although, a direct comparison is more difficult than for simulated sources, because the frequency difference is quite large and the signal-to-noise ratios are not equal.

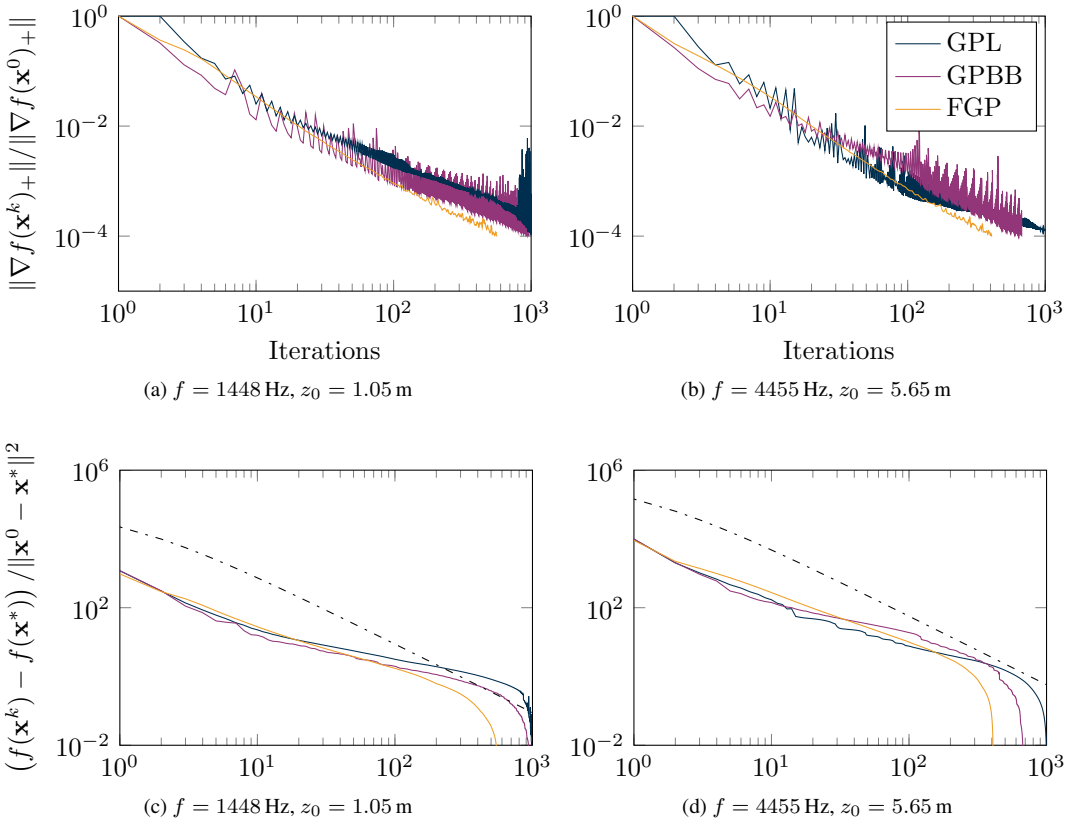
Method	Iterations k	Time $[s]$
GPL	1000	7.0
GPBB	960	7.9
FGP	580	2.9

(a) $z_0 = 1.05\,\text{m}$ at $f = 1448\,\text{Hz}$.

Method	Iterations k	Time $[s]$
GPL	1000	7.1
GPBB	675	5.6
FGP	460	2.4

(b) $z_0 = 5.65\,\text{m}$ at $f = 4455\,\text{Hz}$.

Table 5.3: Number of iterations and computational run time for two point source configuration in presence of noise.



(a) $f = 1448\,\text{Hz}$, $z_0 = 1.05\,\text{m}$

(b) $f = 4455\,\text{Hz}$, $z_0 = 5.65\,\text{m}$

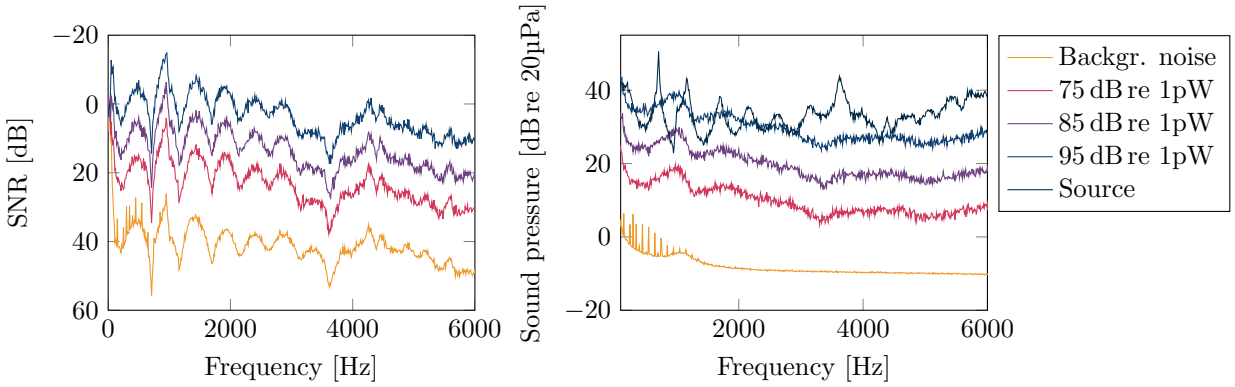
(c) $f = 1448\,\text{Hz}$, $z_0 = 1.05\,\text{m}$

(d) $f = 4455\,\text{Hz}$, $z_0 = 5.65\,\text{m}$

Figure 5.11: Norm of the projected gradient and decrease in objective function value for two point sources separated by $\Delta x = 0.7\,\text{m}$ at two different distances. The dash-dotted line represent the upper complexity bound of FGP $\mathcal{O}(1/k^2)$.

5.3 Two point sources with external noise $z_0 = 270\,\text{cm}$

The effect on efficiency of the deconvolution methods in presence of noise is investigated more thoroughly in this section. Consider again, two sources separated by $\Delta x = 0.7\,\text{m}$, located now $z_0 = 2.70\,\text{m}$ from the microphone array. An external noise source is positioned outside the field of view of the microphone array as depicted in Figure 5.1, driven with Gaussian white noise at various noise levels, described by sound power level settings 75, 85, and 95 dB re 1pW. The combined autospectrum of the sources without the external noise source is shown in Figure 5.12b along with the individual autospectra of the external noise source at three different noise level settings. The corresponding signal-to-noise ratios as function of frequency are given in Figure 5.12a.



(a) Signal-to-noise ratio.

(b) Autospectra

Figure 5.12: Signal-to-noise ratio and mean autospectra over all microphones of sources at $z_0 = 2.70\,\text{m}$ and external noise source at various noise level settings.

The considered frequencies are chosen to cover a wide range of signal-to-noise ratios, beamformer resolutions $R/\Delta x$, and additionally, to ensure that both sources have a strong response. The latter case can be assessed by studying the individual autospectra of the sources in Figure 5.4a and corresponding beamformer maps. The chosen frequencies are listed in Table 5.4 along with the beamformer resolution and signal-to-noise ratios. The beamformer opening angle is $\phi = 15^\circ$ giving an observation area of $1.5 \times 1.5\,\text{m}$. Beamformer maps at $f = 1358$, 1627, and $2480\,\text{Hz}$ are depicted in Figure 5.13.

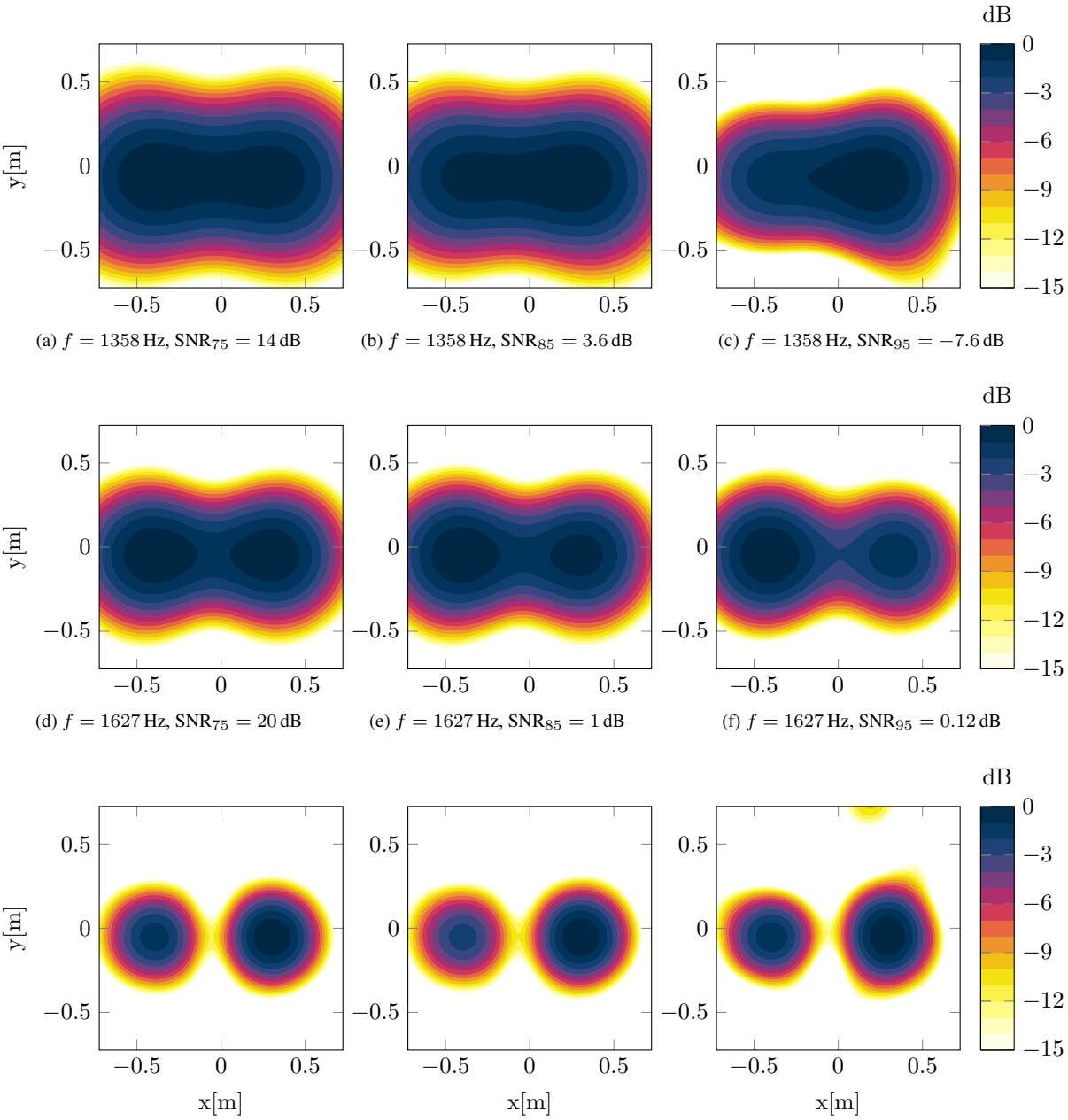
f	$R/\Delta x$	SNR_{75}	SNR_{85}	SNR_{95}
1358	1.2	14	3.6	-7.6
1627	1.0	20	11	0.12
2009	0.8	18	7.4	-2.2
2480	0.7	19	8.9	-0.27
2974	0.6	22	10.9	1.37

Table 5.4: Considered frequencies, beamformer resolution and signal-to-noise ratios.

The beamformer maps are supplied to the deconvolution algorithms with stopping criteria $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ and a maximum of 1000 iterations. A selection of deconvolved maps are shown in Figure 5.14 and the number of iterations and computational time are plotted against beamformer resolution $R/\Delta x$ and signal-to-noise ratio in Figure 5.15.

In general, GPL is clustered towards the bottom with fewest number of iterations and shortest run time. FGP

is in most cases at an intermediate number of iterations, while GPBB is predominantly located in the top with most iterations and longest run times. No general trend between $R/\Delta x$, SNR, number of iterations, and computational time seem to exist. This is somewhat unexpected since the signal-to-noise ratio is presumed to affect the ill-posedness of the deconvolution problem and thus the number of iterations.

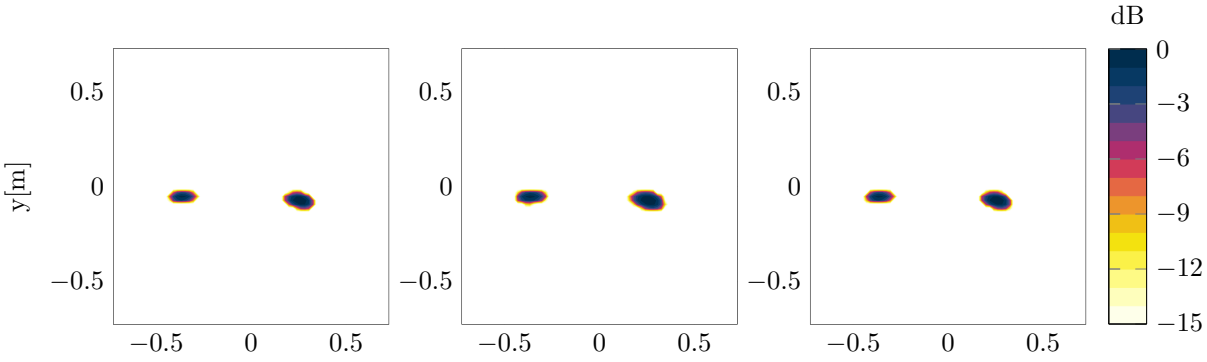


(a) $f = 1358\,\text{Hz}$, $\text{SNR}_{75} = 14\,\text{dB}$ (b) $f = 1358\,\text{Hz}$, $\text{SNR}_{85} = 3.6\,\text{dB}$ (c) $f = 1358\,\text{Hz}$, $\text{SNR}_{95} = -7.6\,\text{dB}$

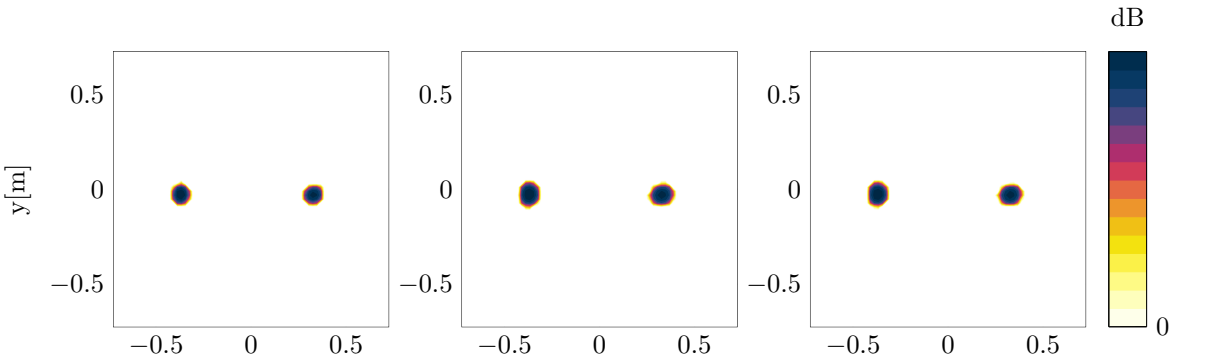
(d) $f = 1627\,\text{Hz}$, $\text{SNR}_{75} = 20\,\text{dB}$ (e) $f = 1627\,\text{Hz}$, $\text{SNR}_{85} = 1\,\text{dB}$ (f) $f = 1627\,\text{Hz}$, $\text{SNR}_{95} = 0.12\,\text{dB}$

(g) $f = 2480\,\text{Hz}$, $\text{SNR}_{75} = 19\,\text{dB}$ (h) $f = 2480\,\text{Hz}$, $\text{SNR}_{85} = 8.9\,\text{dB}$ (i) $f = 2480\,\text{Hz}$, $\text{SNR}_{95} = -0.27\,\text{dB}$

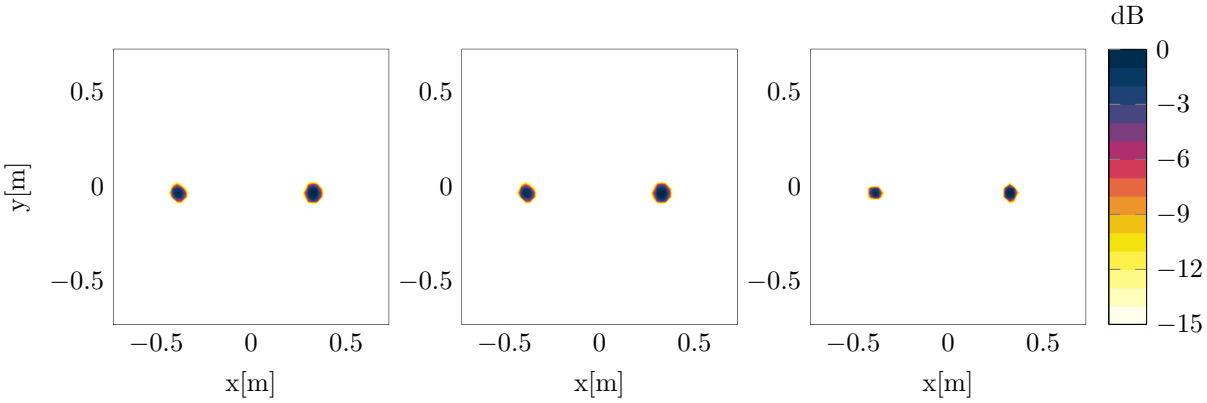
Figure 5.13: Beamformer maps of two point sources at $z_0 = 2.70\,\text{m}$. $f = 1358$, 1627, and 2480 Hz, corresponding to resolutions $R/\Delta x = 1.2$, 1.0, and 0.7 respectively.



(a) GPL $f = 1358\,\text{Hz}$, $\text{SNR}_{95} = -7.6\,\text{dB}$ (b) GPBB $f = 1358\,\text{Hz}$, $\text{SNR}_{95} = -7.6\,\text{dB}$ (c) FGP $f = 1358\,\text{Hz}$, $\text{SNR}_{95} = -7.6\,\text{dB}$



(d) GPL $f = 1627\,\text{Hz}$, $\text{SNR}_{85} = 1\,\text{dB}$ (e) GPBB $f = 1627\,\text{Hz}$, $\text{SNR}_{85} = 1\,\text{dB}$ (f) FGP $f = 1627\,\text{Hz}$, $\text{SNR}_{85} = 1\,\text{dB}$



(g) GPL $f = 2480\,\text{Hz}$, $\text{SNR}_{75} = 19\,\text{dB}$ (h) GPBB $f = 2480\,\text{Hz}$, $\text{SNR}_{75} = 19\,\text{dB}$ (i) FGP $f = 2480\,\text{Hz}$, $\text{SNR}_{75} = 19\,\text{dB}$

Figure 5.14: Deconvolved maps of two point sources at $z_0 = 2.70\,\text{m}$. $f = 1358$, 1627, and 2480 Hz, corresponding to resolutions $R/\Delta x = 1.2$, 1.0, and 0.7 respectively.

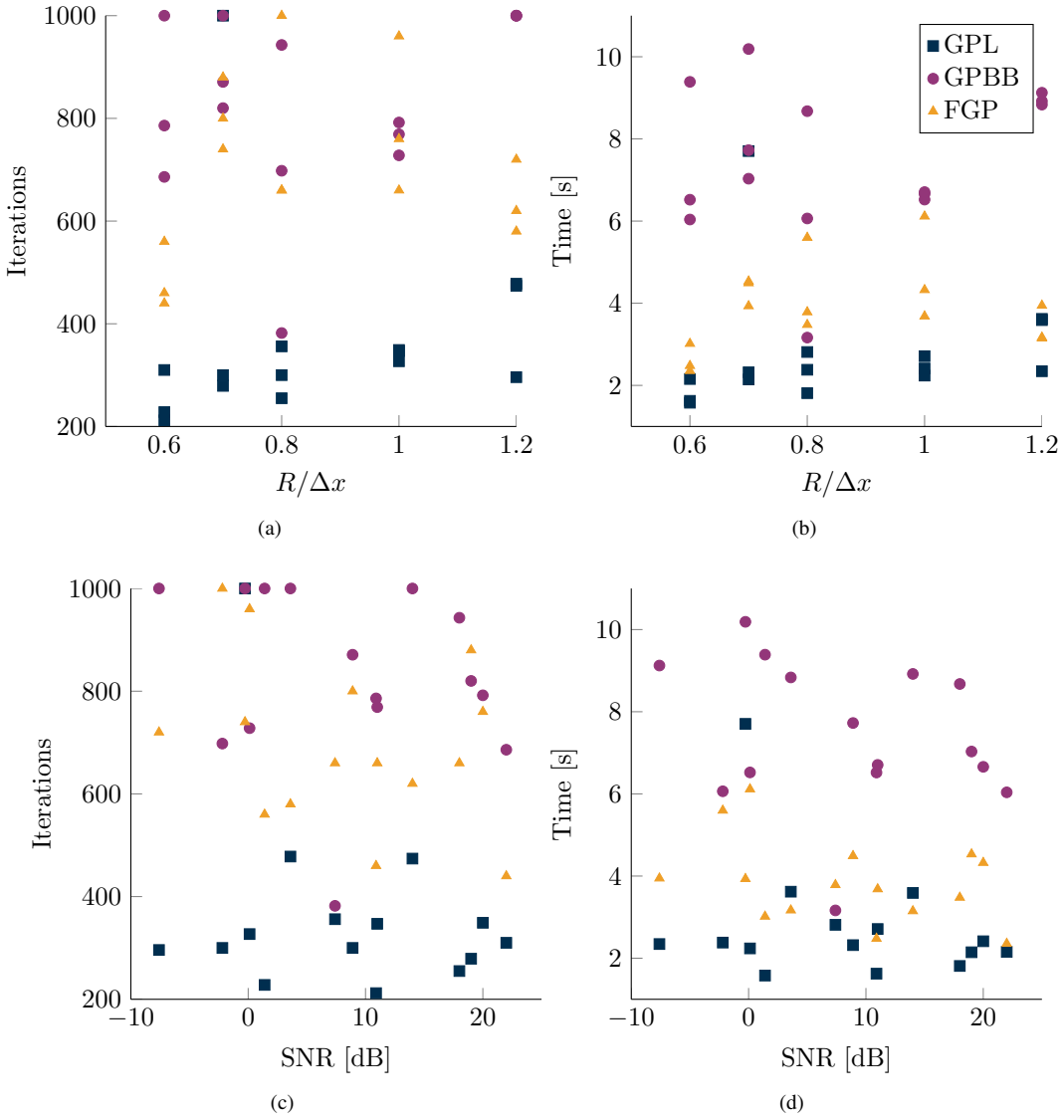
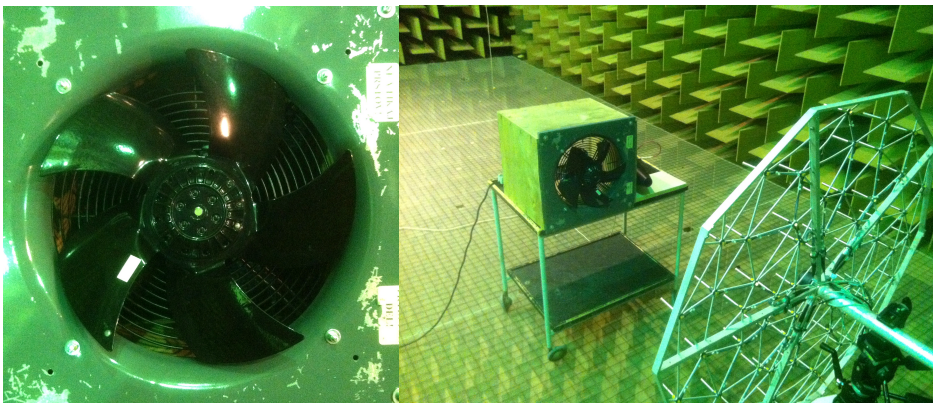


Figure 5.15: Performance comparison of deconvolution algorithms for different signal-to-noise ratios and beamformer resolutions.

5.4 Fan source

The source distributions considered so far have been simple point-like sources, well-suited for testing of deconvolution algorithms. Real-world applications are rarely as simplistic, why the two next sections are dedicated to measurements of more complicated sources.

Figure 5.16 shows a reconstructed fan, originally build as an air fan. The fan rotates at about 1330 RPM and the air flow rearwards. The distance from the center of the rotor to the tip of the blades is approximately 13.5 cm and a small metallic bolt is attached at the outermost edge of one of the blades, marked with a piece of white tape seen in Figure 5.16a. The bolt makes this fan an interesting measurement object, since the spectrum at high frequencies is expected to be dominated by noise generated by the bolt due to turbulence. The autospectrum is shown in Figure 5.17. At frequencies below 1000 Hz, modal components are strongly present, presumably arising from the motor itself.



(a) (b)

Figure 5.16: Measurement setup with fan.

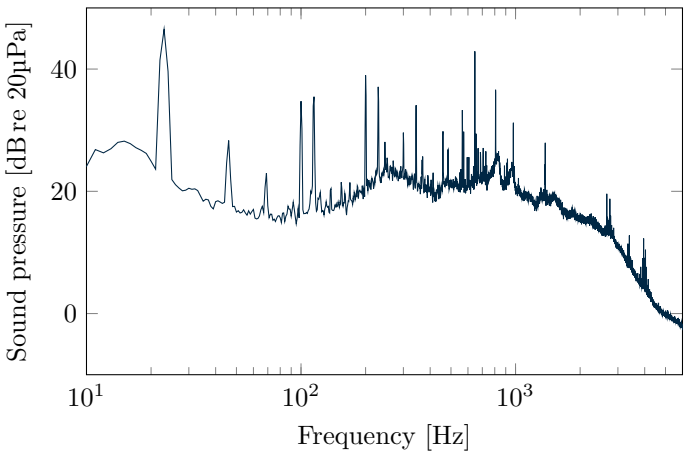


Figure 5.17: Mean autospectrum over all microphones of fan at $z_0 = 0.72\,\text{m}$.

The cross-spectrum matrix is calculated by the previously stated procedure and the results thus describe a spatial average. The considered area of interest is $0.5 \times 0.5\,\text{m}$ located at $z_0 = 0.72\,\text{m}$, giving an opening angle of $\phi = 20°$. The frequencies considered are above approximately 2000 Hz, where the beamformer resolution is $R \leq 0.15\,\text{m}$ and summarized in Table 5.5. The beamformer maps in Figure 5.18 show, that

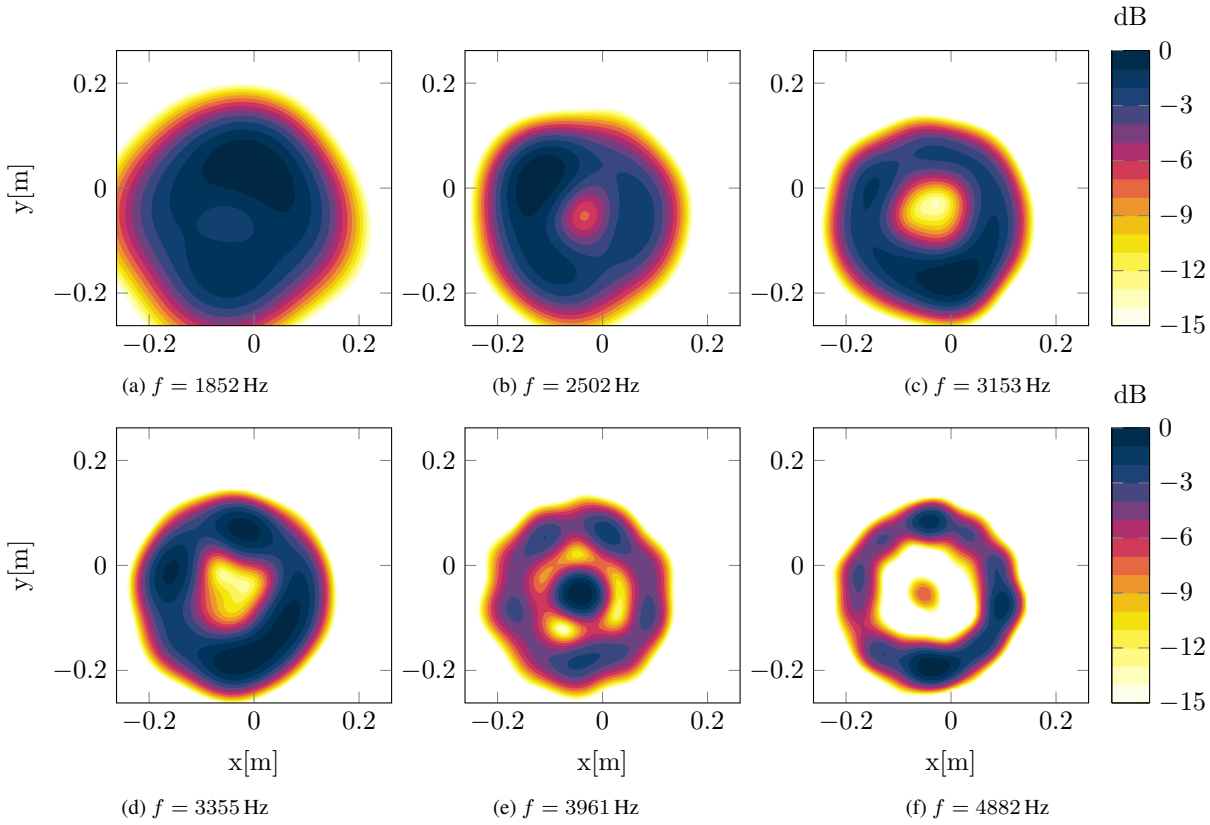
main noise contribution arise from a broad ring or annulus with an outer diameter of about $35\,\text{cm}$ at all the considered frequencies.

The deconvolution algorithms are applied with stopping criteria $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ and a maximum of 1000 iterations. Deconvolved maps at selected frequencies $f = 1852\,\text{Hz}$, $3355\,\text{Hz}$, and $3961\,\text{Hz}$ are shown in Figure 5.19. The noise contribution is now narrowed down to a small ring with a diameter of about $28\,\text{cm}$, which is a close match to the position of the metallic bolt that is located at a radius of $13.5\,\text{cm}$. The source distribution at $f = 3961\,\text{Hz}$, depicted in the lower panel of Figure 5.19 has been split up into small line segments, which agrees quite well with the beamformer map in Figure 5.18e. The noise contribution is not totally uniform, which could be due to some resonance in the spectrum. Consulting the autospectrum reveal that there are some peaks around $4000\,\text{Hz}$ that could be resonances or some interaction between the frame and passing blades. Moreover, the level at high frequencies are quite low and since diagonal removal is applied, the level might be too low in some regions to be fully included in the deconvolved map.

The number of iterations of computational run time is shown in Figure 5.20, where also the usual performance measures are shown for $f = 3153\,\text{Hz}$. The deconvolution algorithms generally use fewer iterations in the low frequency bands. This can be caused by the shift-variant point-spread function that induce a larger model–data mismatch at high frequencies, thus slowing down convergence. The convergence rates show, that FGP attain a superior rate of convergence, while GPL and GPBB once again are seen to cross the upper complexity bound of FGP.

f [Hz]	1852	2502	3153	3355	3961	4882
R [m]	0.18	0.14	0.11	0.10	0.09	0.07

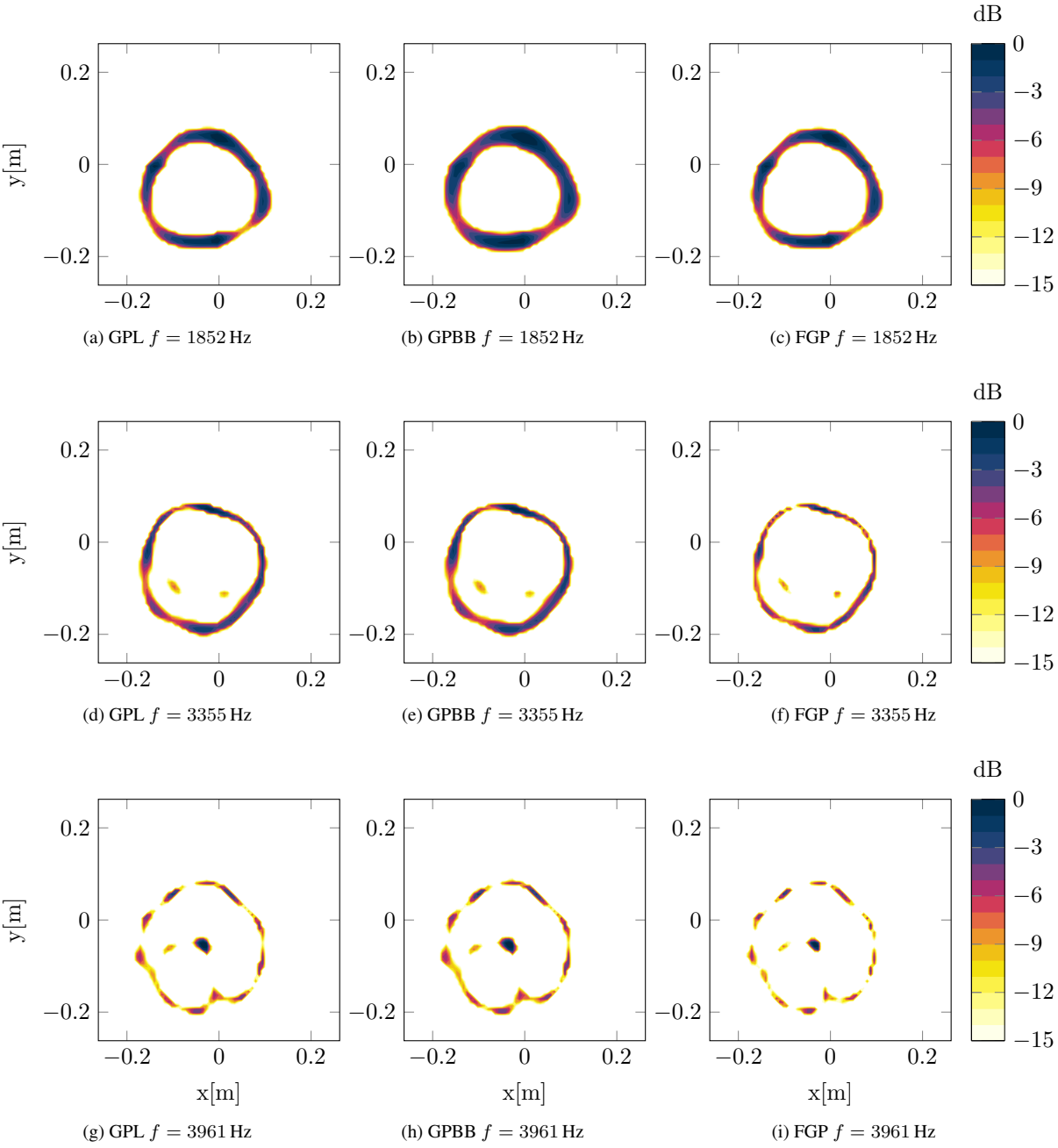
Table 5.5: Considered frequencies and beamformer resolution.



(a) $f = 1852\,\text{Hz}$ (b) $f = 2502\,\text{Hz}$ (c) $f = 3153\,\text{Hz}$

(d) $f = 3355\,\text{Hz}$ (e) $f = 3961\,\text{Hz}$ (f) $f = 4882\,\text{Hz}$

Figure 5.18: Beamformer maps of fan located at $z_0 = 0.72\,\text{m}$ at 6 different frequencies.

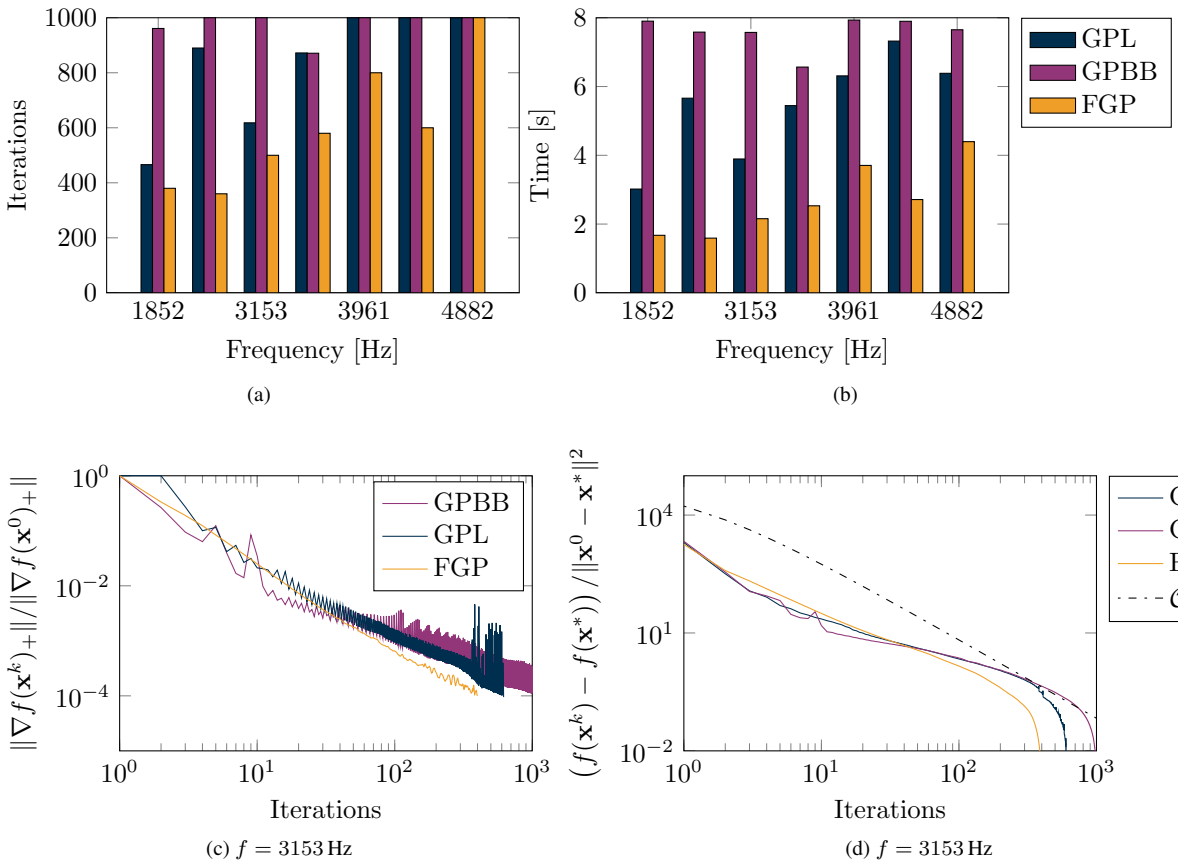


(a) GPL $f = 1852$ Hz (b) GPBB $f = 1852$ Hz (c) FGP $f = 1852$ Hz

(d) GPL $f = 3355$ Hz (e) GPBB $f = 3355$ Hz (f) FGP $f = 3355$ Hz

(g) GPL $f = 3961$ Hz (h) GPBB $f = 3961$ Hz (i) FGP $f = 3961$ Hz

Figure 5.19: Deconvolved maps at $f = 1852, 3355, 3961$ Hz.



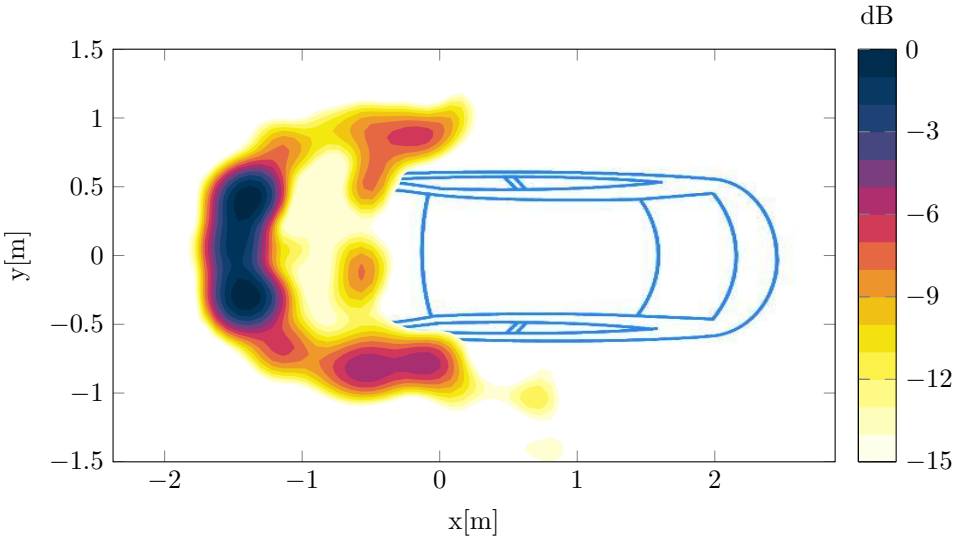
(a) (b)

(c) $f = 3153\,\mathrm{Hz}$ (d) $f = 3153\,\mathrm{Hz}$

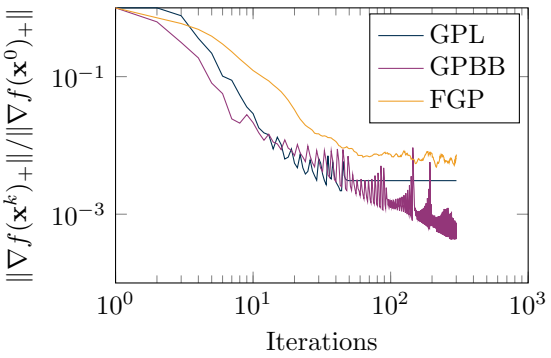
Figure 5.20: Results summary for deconvolution algorithm applied to beamformer maps of a fan at $z_0 = 0.72\,\mathrm{m}$.*Top panel:* Number of iterations and computational run time. *Bottom panel:* Performance measures: Norm of projected gradient and decrease in objective function value.

5.5 Car in Wind Tunnel

The performance of the deconvolution algorithms has been assessed by different, although simple, source distributions, and different aspects have been highlighted by construction of specific tests. In order to get a complete picture of the performance and quality of the deconvolution algorithms, measurements of a car in a wind tunnel is used[1]. The measurements considered was obtained in a wind tunnel experiment with a 114 channel microphone array placed $z_0 = 5\,\mathrm{m}$ above a car. Such a measurement scenario is performed to evaluate the exterior aerodynamic noise contributions on the car, caused by wind resistance when the car is moving. A beamformer map at $f = 2304\,\mathrm{Hz}$ is shown in Figure 5.21a[2]. The air flow on the front spoiler and side view mirrors are the primary exterior noise source. The angle covered by the beamformer in this map is approximately $\phi = 30^\circ$ in the horizontal direction. At this opening angle, the point-spread function is rather shift-variant, thus inducing a model–data mismatch that affects the resolution limit of the deconvolution algorithms.



(a) Beamformer map at $f = 2304\,\mathrm{Hz}$ of car in a wind tunnel.



(b) Decrease in norm of projected gradient.

Figure 5.21: Beamformer map and performance of deconvolution algorithms at $f = 2304\,\mathrm{Hz}$.

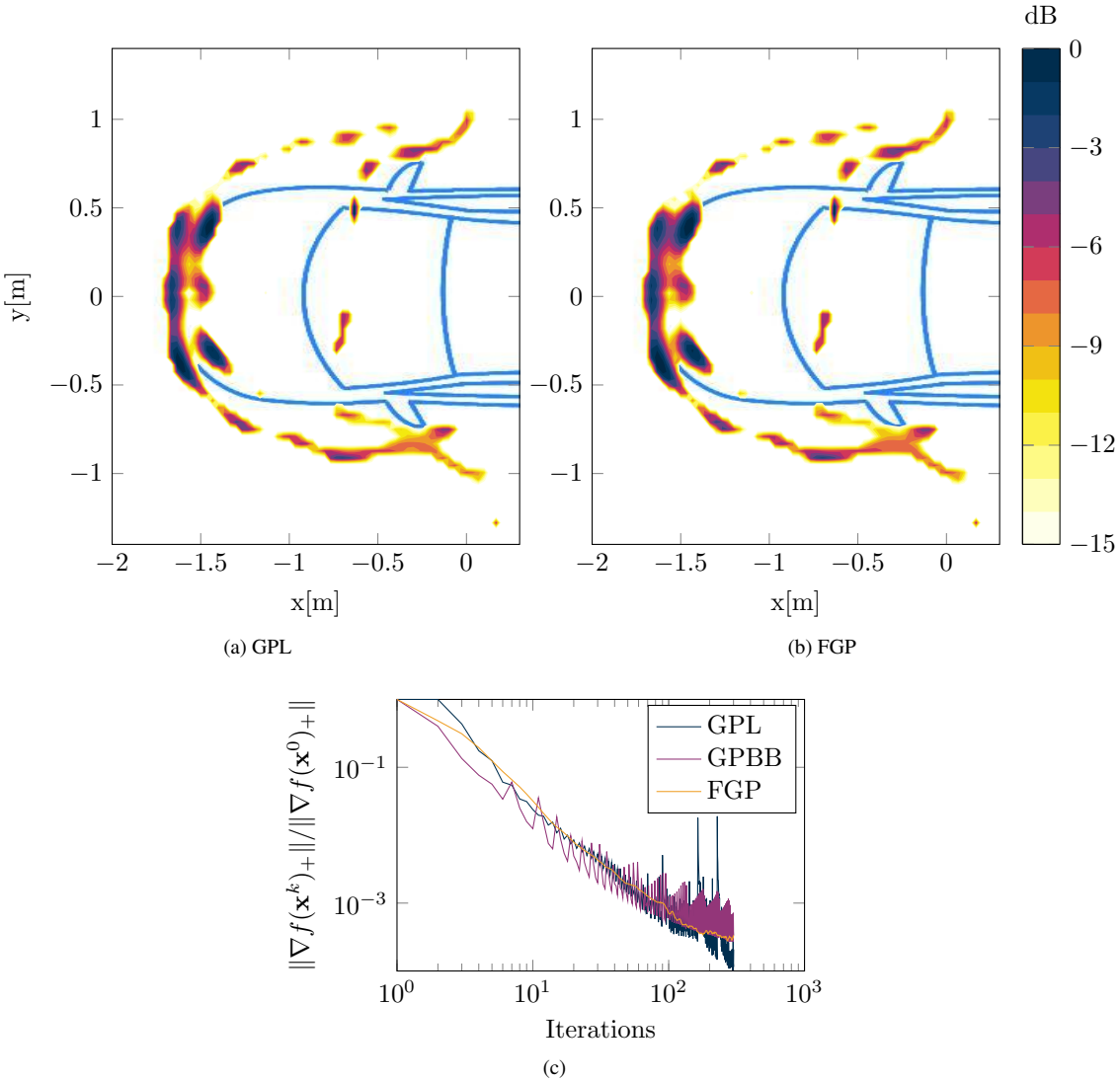
The performance of the deconvolution algorithms applied to the beamformer map in Figure 5.21a is shown in Figure 5.21b. The deconvolved maps are not shown here. The stopping criteria are $\mathrm{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$

[1] Measurement data is kindly supplied by Jørgen Hald, Brüel & Kjær

[2] The car sketch and beamformer/deconvolved maps might not be perfectly aligned in the following, why the maps should be interpreted with caution.

and a maximum of 300 iterations. After a number of iterations the convergence seem to stagnate, especially for GPL and FGP. After about 50-60 iterations no real progress is made towards the optimal solution and GPL in particular, exhibits a total stagnation. This is contributed to the analytical stepsizes of GPL, that attain values less than floating-point accuracy in MATLAB: `eps` $= 2^{-52} \approx 10^{-16}$, thus effectively taking stepsizes of zero length. Interestingly, GPBB seems to be unaffected and the decrease continue until the maximum number of iterations is reached. This stagnation effect is contributed to the model–data mismatch caused by the shift-invariant assumption of the point-spread function in spite of it is actually shift-variant.

By considering a smaller area than the one depicted in Figure 5.21a, the opening angle is reduced to about $\phi = 15^\circ$ and the model–data misfit improves. Deconvolved maps of GPL and FGP are shown in Figure 5.22 along with the norm of the projected gradient. The convergence rate is clearly improved by reducing the beamformer opening angle.



(a) GPL

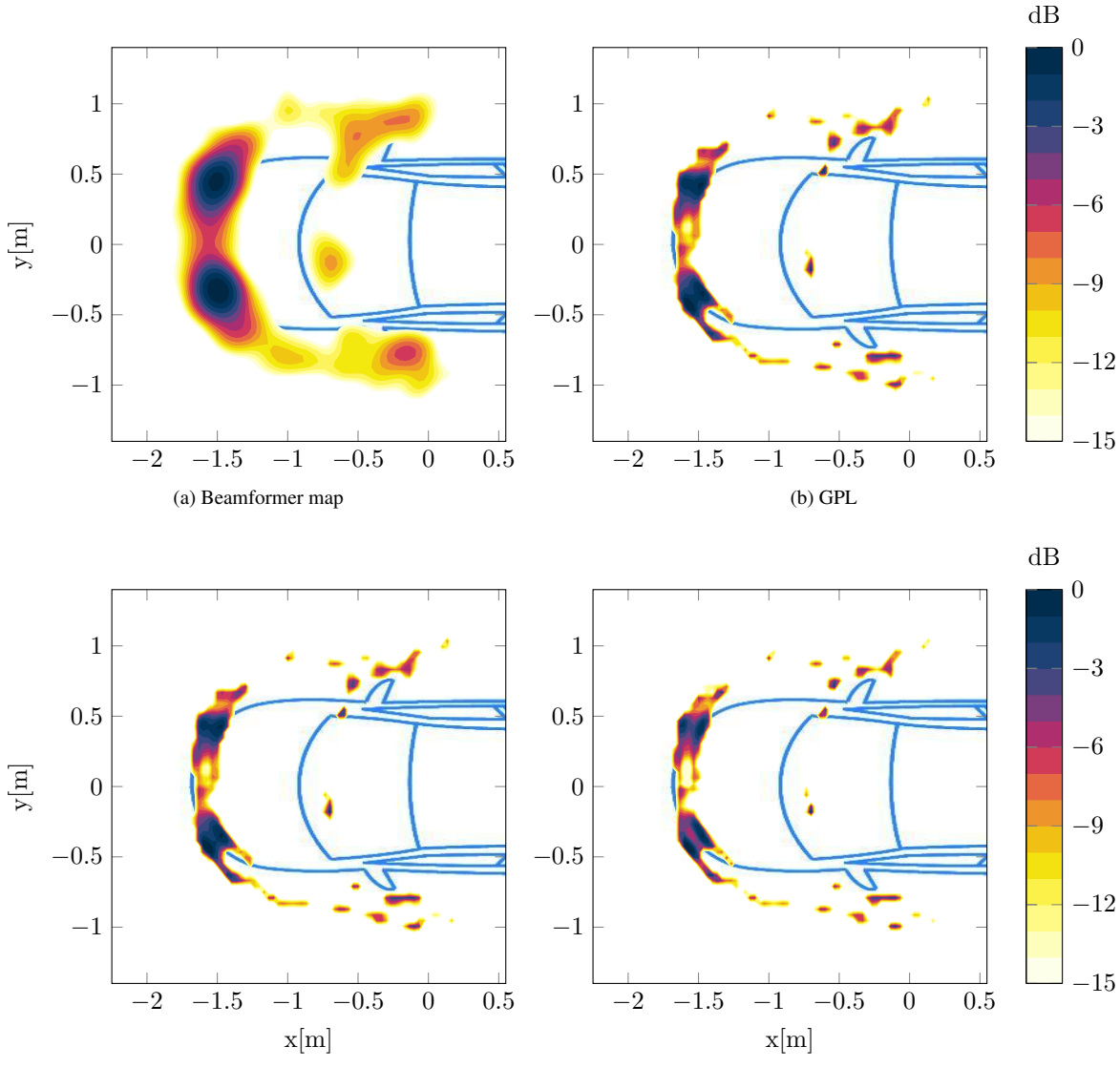
(b) FGP

(c)

Figure 5.22: Deconvolved maps and performance of deconvolution methods at $f = 2304\,\text{Hz}$.

There appears still to be an upper frequency limit, even for the smaller area of interest, around 3000 Hz, where the convergence rate of GPL and FGP stagnates. Consider for instance the beamformer map at

$f = 3072\,\text{Hz}$, depicted in Figure 5.23a. Once again, the main noise contributions are located at the front spoiler and side view mirrors, however in contrast to Figure 5.21a, the rather uniform line source at the front spoiler has divided into two well separated sources. The deconvolved maps are shown in Figure 5.23b-d. The overall quality is quite good and spatial resolution is clearly improved. The norm of the projected gradient, depicted in Figure 5.24, shows a stagnation in convergence rate for GPL after about 100 iterations.



(a) Beamformer map

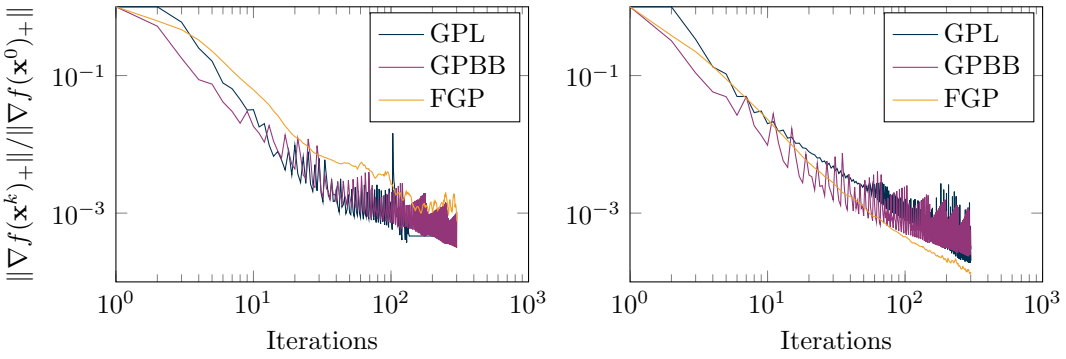
(b) GPL

(c) GPBB

(d) FGP

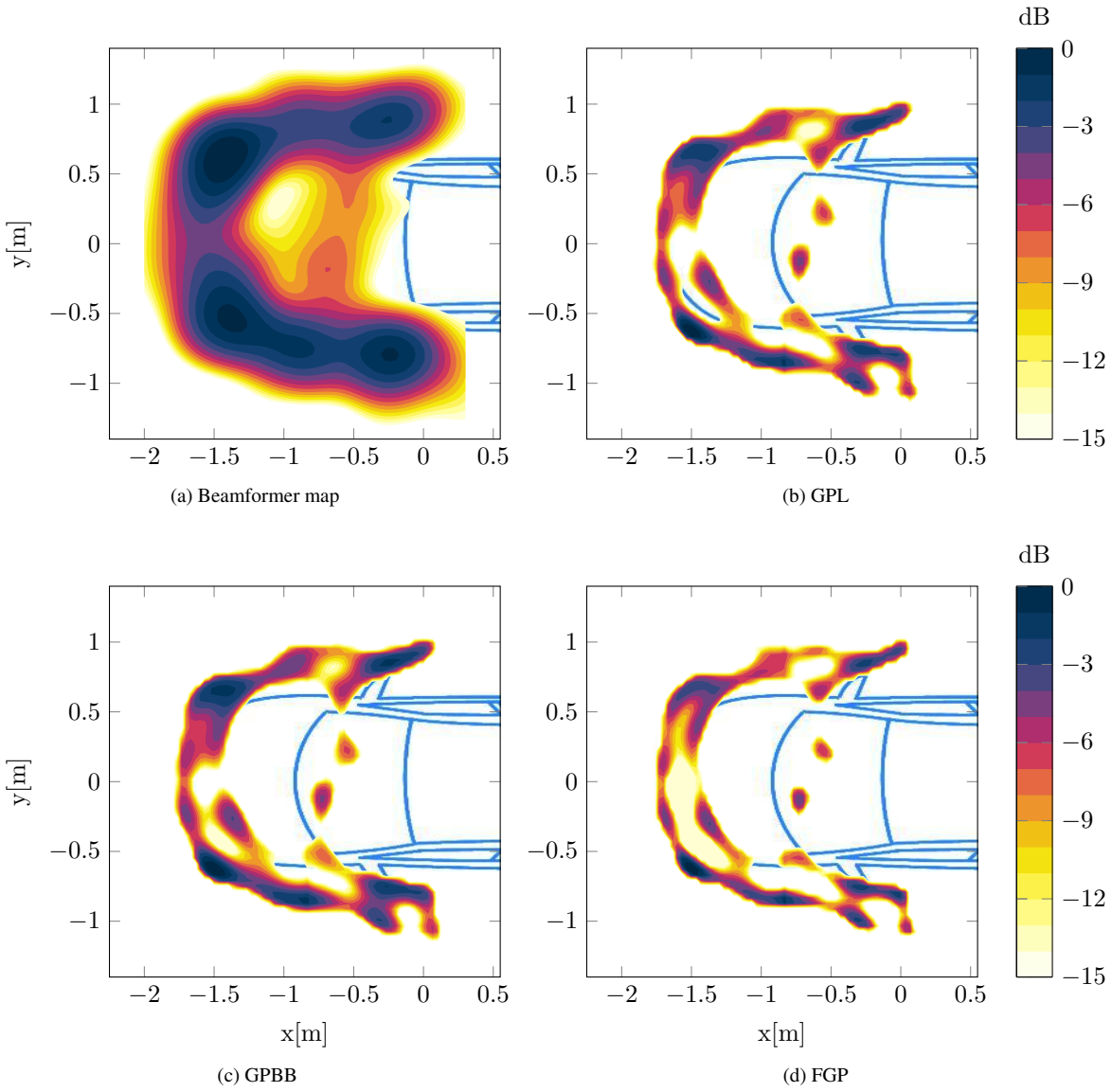
Figure 5.23: Beamformer map and deconvolved maps at $f = 3072\,\text{Hz}$.

At lower frequencies, where the spatial resolution of the beamformer maps are considerably lower, the directivity of the point-spread function is less pronounced, and the deconvolution algorithms perform better. Consider for instance the beamformer map at $f = 1664\,\text{Hz}$, depicted in Figure 5.25 along with the deconvolved maps. The norm of the projected gradient is shown in Figure 5.24b. Comparing the decrease in norm of projected gradient, GPL and FGP perform much better at low frequency, while GPBB is mostly the same.



(a) $f = 3072\,\text{Hz}$ (b) $f = 1664\,\text{Hz}$

Figure 5.24: Norm of the projected gradient.



(a) Beamformer map (b) GPL

(c) GPBB (d) FGP

Figure 5.25: Beamformer map and deconvolved maps at $f = 1664\,\text{Hz}$.

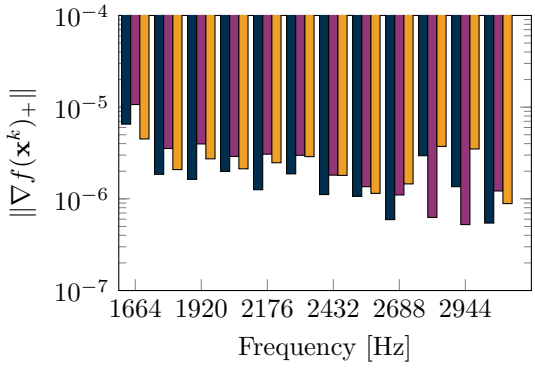
5.5.1 Warm-start

The concept of warm-starting the deconvolution algorithms, considered in Section 4.3, showed promising results in terms of efficiency and resolution on a simple two point source distribution. The source positions and strengths were independent of frequency which is rarely the case in real-world applications, such as the wind tunnel experiment shown above. To complete the investigation of warm-starting, measurements from the wind tunnel are considered at frequencies,

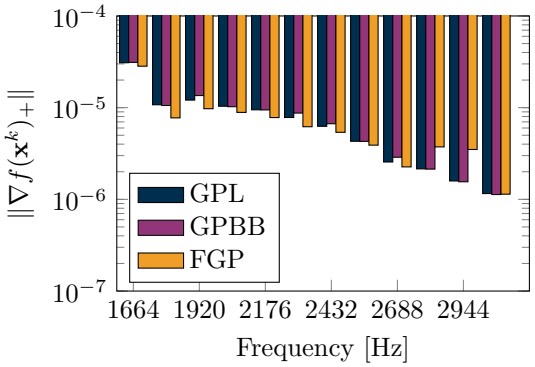
f [Hz]	1664	1792	1920	2048	2176	2304	2432	2560	2688	2816	2944	3072

Where the bandwidth is 128 Hz. Starting at 3072 Hz, the beamformer map and zero starting guess is supplied to the deconvolution algorithms. The solution at this frequency band is then supplied as a starting guess to the next frequency band 2944 Hz, and so forth, until the frequency band at 1664 Hz is reached. The efficiency is evaluated by number of iterations, computational time, and absolute norm of the projected gradient of the last iterate in each frequency band. The stopping criteria are $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ and a maximum of 300 iterations. It is assumed that when a proper starting guess is supplied, the gradient tolerance need not be higher than 10^{-3}. The deconvolved map in the frequency band at 1664 Hz serves as a reference of achievable reconstruction quality and was shown in Figure 5.25. Two sequences are compared with and without warm-starting, however both with $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$. The norm of the projected gradient and number of iterations are summarized in Figure 5.26 and the deconvolved maps are shown in Figure 5.27.

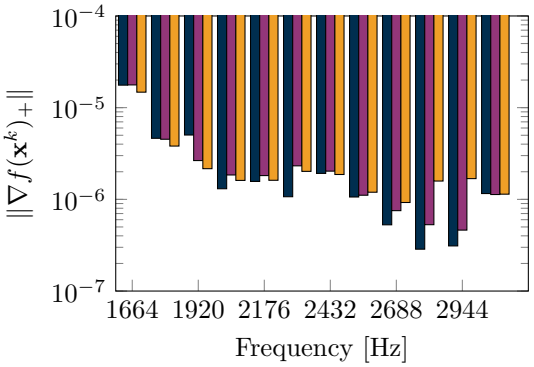
The sequence without warm-starting requires, for GPL and GPBB, less than 100 iterations in each frequency band to reach $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$. The FGP method requires nearly 300 iterations in the first 3 frequency bands but reduce the number of iterations with decreasing frequency. The sequence with warm-starting requires more iterations, on average between 100 and 300 iterations in each frequency band, why it is more time consuming than the sequence without warm-starting. As a reference, a sequence with a gradient tolerance of 10^{-4} requires 300 iterations in all frequency bands. The deconvolved maps in Figure 5.27 reveal that a clear resolution improvement is achieved with warm-starting. The resolution is visually at a level comparable to the maps in Figure 5.25 and with a slightly higher resolution. This is attained with fewer iteration overall compared to a sequence with $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$ that required 300 iterations in each frequency bands. The improvement can also be seen from the norm of the projected gradient, which is somewhat lower in most frequency bands compared to the sequence with $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$, and in fact also slightly lower than the sequence with $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$.



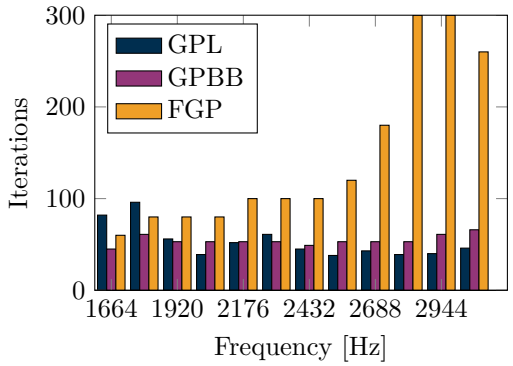
(a) Without warm start, $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-4}$



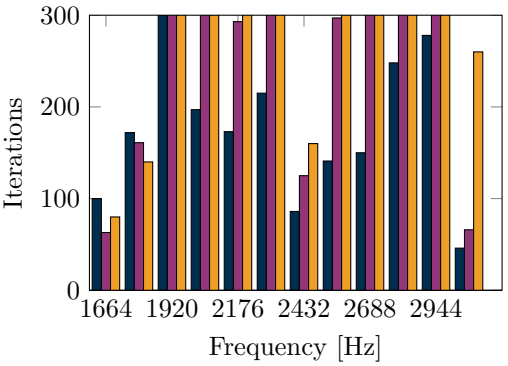
(b) Without warm start, $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$



(c) With warm start,$\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$

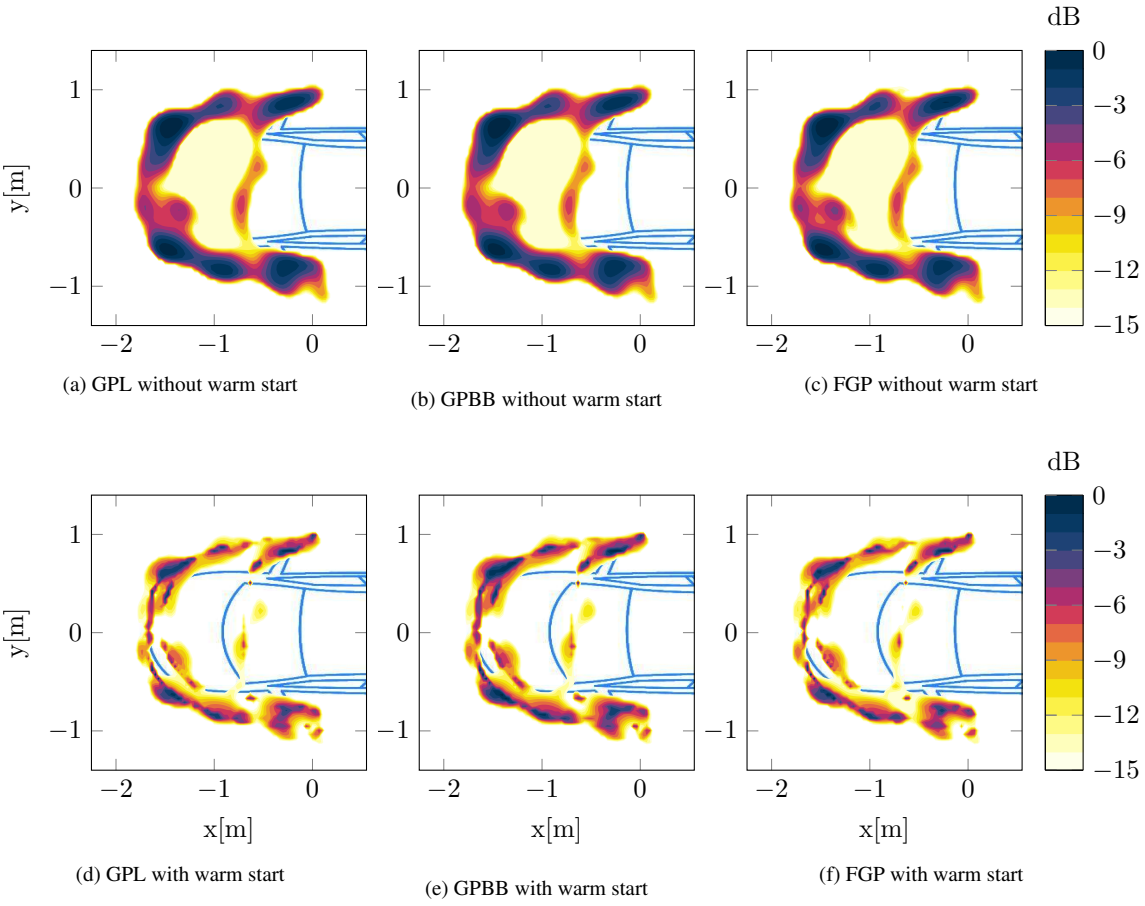


(d) Without warm start, $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$



(e) With warm start, $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$

Figure 5.26: Comparison of performance measures with and without warm-start.



(a) GPL without warm start

(b) GPBB without warm start

(c) FGP without warm start

(d) GPL with warm start

(e) GPBB with warm start

(f) FGP with warm start

Figure 5.27: Deconvolved maps with $\text{TOL}\nabla f(\mathbf{x})_+ \leq 10^{-3}$ calculated with and without warm-starting.

6 Summary & Discussion

A total of 10 tests have been carried out to investigate various aspects of the deconvolution algorithms GPL, GPBB, and FGP. In the following, the main results are discussed and compared. The tests are assigned a capital letter for easier referencing in the discussion and summarized in Table 6.2 with source configuration, frequency, main objective, and section numbering.

Test	**Source configuration**	**Frequency** f [Hz]	**Objective**	**Section**
		Simulation study:		
A	Single point source $z_0 = 1\,\text{m}$, $\phi = 30°$ $z_0 = 10\,\text{m}$, $\phi = 10°$	1000-4000	Efficiency of shift-variant and shift-invariant point-spread functions	4.1.
B	Two point sources $z_0 = 3\,\text{m}$, $\phi = 30°$ $z_0 = 5\,\text{m}$, $\phi = 10°$	1500-6500	Resolution of deconvolution methods	4.2
C	Two point sources $z_0 = 3\,\text{m}$, $\phi = 30°$	1000-4000	Warm-start	4.3
D	Multiple point sources $z_0 = 5\,\text{m}$, $\phi = 20°$	4000	Diagonal removal	4.4
		Experimental study:		
E	Two point sources $z_0 = 1.05\,\text{m}$, $\phi = 35°$ $z_0 = 5.65\,\text{m}$, $\phi = 7°$	617, 2951	Efficiency of shift-variant and shift-invariant point-spread functions	5.2
F	Two point sources $z_0 = 1.05\,\text{m}$, $\phi = 35°$ $z_0 = 5.65\,\text{m}$, $\phi = 7°$	1448, 4455	Added noise	5.2.1
G	Two point sources $z_0 = 2.70\,\text{m}$, $\phi = 15°$	1358-2974	Robustness towards external noise	5.3
H	Fan source $z_0 = 0.72\,\text{m}$, $\phi = 20°$	1852-4882	Performance on real-world source	5.4
I	Car in wind tunnel $z_0 = 5\,\text{m}$, $\phi = 15°$	1664, 2304, 3072	Performance on real-world application	5.5
J	Car in wind tunnel $z_0 = 5\,\text{m}$, $\phi = 15°$	1664 - 3072	Warm-start	5.5.1

Table 6.2: Summary of tests.

It should be recalled that the initial objective for applying deconvolution to beamformer maps was to improve spatial resolution. In general, all tests have shown that this is in fact the case. In test B, a two-point source configuration was used to quantify the achievable resolution by applying deconvolution. The main result was that a resolution gain of at least a factor of 2 is achievable, meaning that the limits on resolution of the beamformer map, such as lower frequency limit or maximum measurement distance, is halved and doubled respectively with deconvolution. The test used a somewhat coarse measure of resolution, the center point level between the source positions, why another measure, the peak-to-average ratio was considered. However, this measure did not provide a useful description of the attainable resolution. The resolution limit was eventually evaluated visually and shown to comply with the initially proposed resolution measure.

Tests A and E showed the dependence of the performance of the deconvolution methods, on the degree of shift-variance of the point-spread function. Measurement conditions with large beamformer opening angles ϕ evidently have shift-variant point-spread functions, why poor performance and stagnation can occur, as it was seen in tests A and I. Equally important, the considered frequency was seen to affect the shift-variance of the point-spread function in test A, where the performance at low frequencies was comparable between the shift-variant and shift-invariant case. This suggests that at low frequencies, where the point-spread function covers a large area, it does not affect the performance of the deconvolution methods as much as higher frequencies, where it is confined to a smaller area.

In test A, all methods needed fewer iterations to converge at $z_0 = 10\,\mathrm{m}$ compared to $z_0 = 1\,\mathrm{m}$, and particularly towards higher frequencies, fewer iterations was needed. In test E, a similar picture was seen for FGP, while GPL needed more iterations at the far distance $z_0 = 5.65\,\mathrm{m}$. The GPBB method required the fewest iterations compared to GPL and FGP in test A, while it required most in test E. The performance of GPL and FGP are similar in tests A and E with FGP being slightly faster in both cases. Overall, FGP performed most consistently over the two tests showing the same expected behavior while GPBB jumped from the best to worst performance over the two tests.

The results of tests A and E agree well with that of test F, where noisy beamformer maps were considered. The efficiency of GPL and GPBB was seen to cross the theoretical upper bound of FGP and thus provide the first demonstration of its theoretically superior convergence rate. The same crossing was also observed with the fan in test H, where FGP attained a superior convergence rate for all frequency bands. This superior rate of convergence seems to really stand out in tests with 1000 iterations. This is in agreement with the momentum interpretation of FGP, where a starting guess, that is far from the optimum, slows down the initial rate of convergence until enough momentum is gathered to attain an accelerated convergence rate compared to GPL and GPBB. In some applications more than 300 iterations might be too time consuming when many frequency bands are to be calculated. However, due to the improved computational run time seen from FGP, this could be achieved at approximately the same overall run time and at the same time provide increased resolution.

Warm-starting has been assessed in tests C and J. The simple setup in test C showed that the computational time can be reduced significantly when the solution in one frequency band is supplied as a starting guess to the succeeding band. This idea leverages on the fact that the convergence rate is improved when the distance from the starting guess to the optimum is small. The efficiency improvements were also seen with the real-world measurements in test J. Specifically GPL required less than 200 iterations in most frequency bands, compared to the reference run without warm-starting that required 300 iterations in all bands. A total of 1300 fewer iterations over 11 frequency bands were required by GPL. The source distribution was almost perfectly reconstructed in test C when applying warm-starting. Since precautions has been made to avoid inverse crime, the explanation must be that since the source distribution does not change with

frequency, a supplied starting guess will be very close to the optimum, why the methods converge to such good reconstructions. The quality of the reconstruction in test J is more difficult to interpret, since the true source distribution in unknown. However from a visual perspective, the deconvolved maps are similar to the maps derived without warm-starting and no artifacts are present. This suggest that in real-world applications, warm-starting can be applied to a sequence of frequency bands under the assumption that the spectra does not change too much between bands – the bandwidth considered in test J was $128\,\text{Hz}$. The optimal bandwidth and stopping criteria naturally depend on the specific measurement objective, why it might be difficult to implement in a general NNLS solver. With some manual tweaking, warm-starting can supply great reconstructions, than might even obtain a better resolution than the non warm-starting counterpart.

The effects of applying diagonal removal to the cross spectrum matrix was assessed in tests D and F. The main benefit of diagonal removal are less noisy beamformer maps and consequently also deconvolved maps. The tests showed clear improvements in resolution of the deconvolved maps, however test D also showed that the level is decreased in the process and sources with low power levels might be effectively removed from the maps. The performance measures of the deconvolution methods showed that the initial convergence rate was better in the case without DR and all methods performed equally well. After about 30-40 iterations the methods stagnated and little progress was made for the remaining iterations. This was most pronounced in the case where the source strengths ranged from 0 to $-14\,\text{dB}$. The stagnation did not seem to occur in the case with DR. The initial convergence is slower but constant throughout all iterations. GPL performed slightly better than GPBB and FGP in the case with DR. In test F, this stagnation was also seen, specifically for the case at short distance.

The robustness towards external noise was assessed in test G. Five different frequency bands each at three different signal-to-noise ratios were used to test if the performance of the deconvolution methods is dependent on the noise level of external sources. With signal-to-noise ratios ranging from -10 to $20\,\text{dB}$ and resolution ratios $R/\Delta x$ from 0.6 to 1.2, no general trend was observed for either measure, however, a linear regression analysis could be applied to determine if this is truly the case. This would on the other hand require more measurement data to improve the statistical significance.

The limiting factors of deconvolution are mainly posed by the boundary conditions enforced by zero-padding, and the degree of shift-variance of the point-spread function. The former was shown to create strong artifacts in the deconvolved maps at low frequencies, to an extent where the reconstruction is useless. One remedy was proposed by enforcing periodic boundary conditions, where zero-padding was replaced by a periodic extension of the map. This boundary condition is one of several used in Image Deblurring to reduce boundary artifacts [38]. The physical interpretation of these are, regardless of the visual improvements they may pose, not meaningful in beamforming applications. One possible method, originally proposed for Nearfield acoustic holography (NAH) [39], leverages on the fact that some functions in the deconvolution problem are known analytically, why they could be calculated in the full $2N \times 2N$ domain instead of padding with zeros. This idea could be transferred to beamforming deconvolution by calculating the point-spread function in the full $2N \times 2N$ domain, since it is computed analytically, and possibly reduces boundary artifacts.

The second limiting factor of deconvolution is due to shift-variant point-spread functions. The effects on performance have been seen throughout the tests. Several solutions have been proposed in acoustical litterature, for instance coordinate transformation [40], where the idea is to replace the regular equidistant grid with a transformed grid that ensures that the point-spread function is shift-invariant at all grid points. The grid density in such a transformed grid is not uniform, it has high density at the center and low density towards the edges. The achievable resolution is therefore dependent on the local grid density and position of

the sources. Another proposed method, described in [8], embeds the deconvolution algorithms in a loop, where the point-spread functions are calculated for each grid point and used in the deconvolution process. This method ensures that the shift-variance of the point-spread function is taken into account, however at the cost of a substantially higher computational work. Lastly, a method used in Image Deblurring [41], describes how to section the grid into a number of sub grids in which the point-spread function is assumed constant. This method is similar to the embedded, however with a much lower computational cost. The number of sub grids depend on the degree of shift-variance of the point-spread function and is a trade off between resolution and computational time.

Measuring efficiency by the number of iterations an algorithm require to achieve the stopping criteria and awarding each algorithm a first, second or third place in all runs in all 10 tests show, that FGP is the most efficient in most runs. These results are however, excluding the warm-start test, shown in Figure 5.26 and

Position	1.	2.	3.
GPL	3	14	17
GPBB	11	16	8
FGP	18	9	4

Table 6.3: Number of first, second and third places counted in number of iterations in all runs in all 10 tests.

the SNR test, shown in Figure 5.15. In these tests, GPL was clearly superior and adding the places in those to the table, GPL and FGP attain a comparable number of first, second, and third positions.

6.1 Future work

Before the concluding remarks, some recommendations on future work are in order. As mentioned in the discussion above, the limiting factors of deconvolution is shift-variant point-spread functions and boundary conditions. The referenced sectioning method [41], should be relatively easy to implement and compare with the embedded method [8] and coordinate transformation [40]. It would be interesting to see how the resolution and efficiency changes with the number of sections used and what the computational cost will be. The lower frequency limit is dictated by the imposed boundary conditions, and improving this limit requires some work on the physical interpretation of such. The procedure in [39] could be a good starting point.

Warm-starting was seen to produce computationally efficient and high resolution reconstructions by looping through subsequent frequency bands. More work could be put into an investigation of optimal stopping criteria and frequency bandwidth. It is difficult to see that warm-starting could become a general purpose method, however for certain specialized conditions it could become beneficial in terms of efficiency and resolution.

In [37] it is shown that there is a connection between the singular value decomposition of the point-spread function and its apparent width. Specifically it is shown that the rate of decay of singular values is fast if the point-spread function is wide, and slow if it is narrow. This could indicate that the SVD of the point-spread function can provide information about the ill-posedness of the deconvolution problem and possibly the resolution limit of the reconstructed maps. This perspective is intriguing because it could give a diagnosis of the ill-posed problem before it is solved and possibly assist in gaining a better solution.

From the theoretical viewpoint in Section 3.4, efficient methods for the NNLS problem are confined to the class of gradient methods. Only little has been mentioned about alternatives to this class of methods, however some methods that might still be relevant to look into is the interior point methods. Much work has

been done to optimize this class of methods and many commercial and open-source software packages are available. The most challenging part and the key point in getting a computational efficient solver, is to make the algorithm matrix-free. Some work has been done on this [34], although the implementation is far from trivial, why a first step towards developing interior point methods for beamforming deconvolution could be to try out mature software packages such as MOSEK [42] or CVX [43].

7 Conclusion

The basic properties and limitations of delay-and-sum beamforming was outlined in chapter 2 along with the cross-spectrum formulation. This led to the formulation of the deconvolution problem, posed in chapter 3, that due to its ill-posed nature was formulated as a nonnegative least squares (NNLS) problem. The essential principles of optimization was shown to provide a solid framework for the analysis of the deconvolution problem and identification of efficient algorithms. Based on a review of possible algorithms it was concluded that an efficient solver of the NNLS problem should be found within the class of first-order gradient methods. Three of such gradient methods were proposed, where one of them, dubbed GPL, has previously been shown to provide good results in the field of beamforming deconvolution, while the two others, GPBB and FGP, have not previously been applied in this field of research.

Elaborate studies have been carried out, with simulated and experimental data, to test the proposed deconvolution algorithms in different scenarios. The results showed, that the primary cause of slow convergence is due to a model–data mismatch caused by the assumption, that the point-spread function to a good approximation is shift-invariant. In tests, where the model–data mismatch was small, the improvement in resolution, by means of deconvolution, was seen to be about a factor of two, compared to that of the beamformer. This means, for instance, that the lower frequency limit dictated by the beamformer's resolution is halved and that the maximum measurement distance is doubled.

It is well-known that applying diagonal removal to the cross-spectrum matrix improves the spatial resolution of both beamformer and deconvolved maps. This was shown to be true for both simulated and experimental data. Although, low-level noise contributions was seen to be reduced in level in maps with a high dynamic range.

The main contributions were the proposal of deconvolution algorithms GPBB and FGP and the warm-start sequence method. The two deconvolution methods was shown to perform at a level comparable to GPL, although only FGP could provide efficiency improvements for a wide range of test cases. The proposed method for efficient calculation of a range of frequency bands, by means for a warm-start, was shown to reduce computational time and increase the spatial resolution. This was asserted with simulated and experimental data.

Appendices

A MATLAB code

The beamforming framework and deconvolution algorithms have been implemented in MATLAB. In the following, the basic functionality of the code is presented after which the functions are listed for reference.

The construction of beamformer map and point-spread function is given by the function `psf`, with the call

```
[b,PSF,X,Y,x,y] = psf(N,z0,f,phi,rn,source),
```

that takes the measurement geometry, frequency and source positions as input and gives the beamformer map, point-spread function and grid as output.

The deconvolution algorithms are called via the function `soldeconv`, with the call

```
[x,info] = soldeconv(@nnlsqfun,PSF,b,x0,opt),
```

where `@nnlsqfun` is a function handle to the objective function, `PSF` is the point-spread function, `b` is the beamformer map, `x0` is the starting guess and `opt` is a struct with options for the deconvolution algorithms. The function `deconv_opt` holds the default settings, while the chosen deconvolution algorithm is chosen by issuing e.g. `opt.algo = 'FGP'` before `soldeconv` is called to choose deconvolution algorithm FGP.

In the post-processing of measured data, the function `csm` is used to construct cross-spectral matrices. The function call is given by

```
[CSM,freqs,n] = csm(signal,window,noverlap,nfft,fs),
```

where the usual spectral settings are given as input.

The MATLAB code is printed in the following pages.

psf.m

```
function [b,PSF,X,Y,x,y] = psf(N,z0,f,phi,rn,source)

% Number of microphones in array
M = size(rn,1);

% Constants
c = 343;                % Speed of sound
omega = 2*pi*f;
Np = round(N*1.3);      % PSF grid size

% Allocate memory
dj = zeros(Np,Np,M);
ej = zeros(Np,Np,M);
ejhat = zeros(Np,Np,M);
PSF = zeros(Np,Np);
b = zeros(N,N);

% Setup
z0_true = 1.1*z0;           % True distance, z0 ≠ z0_true to ensure that inverse crime is not commited.

L = 2*z0*tand(phi);                % Area covered at distance z0

% Region of interest
x = [-L/2 L/2];     % x grid
y = x;              % y grid

% Beamformer grid
rx = linspace(x(1),x(2),N);
[X,Y] = meshgrid(rx);

% PSF grid
rx_psf = linspace(x(1),x(2),Np);
[Xp,Yp] = meshgrid(rx_psf);

% Following [Koop et al.]:
d0 = sqrt(Xp.^2 + Yp.^2 + z0^2);    % |d0|: Distance from (0,0,0) to all mesh points

% Distance dj from each microphone to each grid point
% Steering vectors ej, ejhat
for n = 1:M
    dj(:,:,n) = sqrt((Xp-rn(n,1)).^2+(Yp-rn(n,2)).^2 + z0^2);
    ej(:,:,n) = (dj(:,:,n)./d0).*exp(1j*omega.*dj(:,:,n)./c);
    ejhat(:,:,n) = (d0./dj(:,:,n)).*exp(1j*omega.*dj(:,:,n)./c);
end

% POINT-SPREAD FUNCTION
% Is constructed by the DAS beamformer response of a unit strength point
% source in the middel of the grid
ind = round(Np/2);
e_unit = reshape(ejhat(ind,ind,:),M,1);

for ii = 1:length(Xp)
    for jj = 1:length(Yp)
        e = reshape(ej(ii,jj,:),M,1);
```

```
        PSF(ii,jj) = e'*e_unit;
    end
end

PSF = rot90(abs(PSF).^2/M^2);      % Normalize PSF
if N ≠ Np
    PSF = interp2(Xp,Yp,PSF,X,Y);
end

% CROSS-SPECTRAL MATRIX
% Construct by placing simulated point sources with value 1
qsource = zeros(N,N);

% Source indices:
ai = sub2ind(size(X),source(:,2),source(:,1));

strength = ones(length(ai),1);

for i = 1:length(ai)
qsource(ai(i)) = strength(i);
end

dj = zeros(N,N,M);
ej = zeros(N,N,M);
ejhat = zeros(N,N,M);
% Distance and steering vectors (using the true distance)
d0 = sqrt(X.^2 + Y.^2 + z0_true^2);    % |d0|: Distance from (0,0,0) to all mesh points
for n = 1:M
    dj(:,:,n) = sqrt((X-rn(n,1)).^2+(Y-rn(n,2)).^2 + z0_true^2);
    ej(:,:,n) = (dj(:,:,n)./d0).*exp(1j*omega.*dj(:,:,n)./c);
    ejhat(:,:,n) = (d0./dj(:,:,n)).*exp(1j*omega.*dj(:,:,n)./c);
end

ejhat = reshape(ejhat,N*N,M)';

Rmn = zeros(M,M);
for ii = 1:length(ai)
        Rmn = Rmn + qsource(ai(ii))*(ejhat(:,ai(ii))*ejhat(:,ai(ii))').';
end

% Diagonal removal
Rmn(logical(eye(size(Rmn)))) = 0;

% DELAY AND SUM BEAMFORMER OUTPUT
for ii = 1:length(X)
    for jj = 1:length(Y)
        e = reshape(ej(ii,jj,:),M,1);
        b(ii,jj) = (e'*Rmn*e)/(M^2-M);
    end
end;
end
```

soldeconv.m

```
function [x,info] = soldeconv(fun,PSF,b,x0,opt)
%SOLDECONV Solve deconvolution problem b = q * psf
%
% Usage:  [x,info] = soldeconv(@fun,PSF,b,x0,opt)
%
% Solve deconvolution problem by following methods:
%  GPL: Gradient projection with exact line search
%  GPBB: Gradient projection with bb-steps
%  FGP: Fast gradient projections
%
% Input:
%   @fun: Function handle with objective function and gradient
%   PSF: Point-spread function
%   b : Beamformer map
%   x0: Starting vector
%   opt: Struct with options - see deconv_opt.m for default settings
%
%
% Output:
%   x: Source distribution for all iterations
%   info: Struct with info about
%       - info.obj
%       - info.grad_norm
%       - info.time
%       - info.iter
%       - info.L
%
% Author: Oliver Lylloff
% Date: 5/2/14
% Latest revision: 5/2/14
%
% http://github.com/1oly/mscthesis

if opt.zeropad
    N = size(b,1);
    npad = 2*N;
    b = zeropad(b,npad,npad,'pad');
    PSF = zeropad(PSF,npad,npad,'pad');
    x0 = zeropad(x0,npad,npad,'pad');
end

switch opt.algo

    case 'GPL'
        [x,info] = gpl(fun, PSF, b, x0, opt);
        if opt.zeropad
            x = zeropad(x,N,N,'unpad');
        end

    case 'FGP'
        [x,info] = fgp(fun, PSF, b, x0, opt);
        if opt.zeropad
            x = zeropad(x,N,N,'unpad');
        end
```

```
    case 'GPBB'
        [x,info] = gpbb(fun, PSF, b, x0, opt);
        if opt.zeropad
            x = zeropad(x,N,N,'unpad');
        end

    otherwise
        error('Unknown method')
end

end
```

deconv_opt.m

```
function options = deconv_opt
% DEFAULT OPTIONS STRUCT FOR DECONVOLUTION ALGORITHMS

% General settings:
options.maxit = 300;        % Max number of iterations
options.tol = 1e-4;         % Projected gradient norm
options.verbose = 1;        % Prints out progress to Command Window
options.zeropad = 1;        % Zero-padding

% FGP:
options.stop_cond = 1;      % 1: Use tol as stopping criterion else 0: just maxit
options.gradn = 1;         % Compute gradient of x every k iterations default = 20
options.backtrack = 0;      % Toggle backtracking on/off
options.descent = 0;        % Descent condition

% GPBB
options.sigma = 0.298;      % Diminishing scalars
options.eta = 0.99;
options.Mds = options.maxit/10;

end
```

nnlsqfun.m

```
function [f,g,r] = nnlsqfun(PSF,b,x,varargin)
% NONNEGATIVE LEAST SQUARES OBJECTIVE
%
% Compute objective, gradient and residual of
% Nonnegative least squares problem (NNLS):
%   minimize_x      0.5||Ax - b||^2
%   subject to         x ≥ 0
%
% Usage:
%       f = nnlsqfun(PSF,b,x)
%       [f,g] = nnlsqfun(PSF,b,x)
%       [f,g,r] = nnlsqfun(PSF,b,x)
%
% Input:
%   PSF: Point-spread function (matrix)
%   b : Beamformer map (matrix)
%   x: Source distribution (matrix)
%   varargin: Can take precomputed PSF
%
% Output:
%   f: Objective function value (scalar)
%   g: Gradient (matrix)
%   r: Residual (matrix)
%
%
% Author: Oliver Lylloff
% Date: 7/2/14
% Latest revision: 5/2/14
%
% http://github.com/1oly/mscthesis


if nargin==5
    Fps = varargin{1};
    FpsT = varargin{2};
else
    Fps = fft2(PSF);
    FpsT = fft2(rot90(PSF,2));
end

r = fftshift(ifft2(fft2(x).*Fps)) - b;
f = 0.5*norm(r,'fro')^2;
if (nargout > 1)
    g = fftshift(ifft2(fft2(r).*FpsT));
end

end
```

gpl.m

```
function [x,info] = gpl(fun, PSF, b, x0, opt)
% GPL GRADIENT PROJECTION ALGORITHM WITH EXACT LINE SEARCH
%
% Usage:  [x,info] = gpbb(@fun, PSF, b, x0, opt)
%
% Input:
%   @fun: Function handle with objective function and gradient
%   PSF: Point-spread function
%   b : Beamformer map
%   x0: Starting vector
%   opt: Struct with options - see deconv_opt.m for default settings
%
%
% Output:
%   x: Source distribution for all iterations
%   info: Struct with info about
%       - info.obj
%       - info.grad_norm
%       - info.time
%       - info.iter
%
% Author: Oliver Lylloff
% Date: 5/2/14
% Latest revision: 5/2/14
%
% http://github.com/1oly/mscthesis
%
% Reference:
% Ehrenfried, Klaus, and Lars Koop. 2008.
% 'A Comparison of Iterative Deconvolution Algorithms for the Mapping of Acoustic Sources.'
% American Institute of Aeronautics and Astronautics.

start_time = tic;

if opt.verbose
   fprintf('GPL ')
end

x = x0;

% Precompute fft of PSF
Fps = fft2(PSF);
FpsT = fft2(rot90(PSF,2));

fgx = @(x) fun(PSF,b,x,Fps,FpsT);
[¬,grad] = fgx(x);
w = grad;
w(x == 0 & w > 0) = 0;
ngrad = norm(w,'fro');
ngrad1 = ngrad;
info.grad_norm(1) = ngrad1;
info.sumx(1) = sum(x(:));
n = 0;
while (n < opt.maxit) && (opt.tol < ngrad/ngrad1)
```

```
    if opt.verbose
        if (mod(n, 30) == 0)
            fprintf('.');
        end
    end

    n = n+1;
    [f,grad,r] = fgx(x);
    w = grad;
    w(x == 0 & w > 0) = 0;

    g = fftshift(ifft2(fft2(w).*Fps));
    alpha = dot(g(:),r(:))/dot(g(:),g(:));

    x = x - alpha*w;
    x(x < 0) = 0;

    info.obj(n) = f;
    ngrad = norm(w,'fro');
    info.grad_norm(n+1) = ngrad;
    info.time(n) = toc(start_time);
end
if opt.verbose
    fprintf('\n\t'); disp([num2str(n) ' iterations in ' num2str(info.time(n)) ' seconds'])
end
info.iter = n;
end
```

gpbb.m

```
function [x,info] = gpbb(fun,PSF,b,x0,opt)
% GPBB GRADIENT PROJECTION ALGORITHM WITH BB-STEPS
%
% Usage:  [x,info] = gpbb(@fun, PSF, b, x0, opt)
%
% Input:
%   @fun: Function handle with objective function and gradient
%   PSF: Point-spread function
%   b : Beamformer map
%   x0: Starting vector
%   opt: Struct with options - see deconv_opt.m for default settings
%
%
% Output:
%   x: Source distribution for all iterations
%   info: Struct with info about
%       - info.obj
%       - info.grad_norm
%       - info.time
%       - info.iter
%
% Author: Oliver Lylloff
% Date: 5/2/14
% Latest revision: 5/2/14
%
% http://github.com/1oly/mscthesis
%
% Reference:
%   Kim, Dongmin, Suvrit Sra, and Inderjit S Dhillon. 2013.
%   'A Non-Monotonic Method for Large-Scale Non-Negative Least Squares.'
%   Optimization Methods and Software 28 (5) (October): 1012-1039.
%

start_time = tic;

if opt.verbose
   fprintf('GPBB ')
end

x = x0;

% Precompute fft of PSF
Fps = fft2(PSF);
FpsT = fft2(rot90(PSF,2));

fgx = @(x) fun(PSF,b,x,Fps,FpsT);
[f,grad] = fgx(x);

% Diminishing scalars
xref = x;
fref = f;
gradref = grad;
beta = 1;           % no scaling at the beginning,
```

```
% Projected gradient
grad(x == 0 & grad > 0) = 0;
gradold = grad;
ngrad = norm(grad,'fro');
ngrad1 = ngrad;
info.grad_norm(1) = ngrad1;
info.obj(1) = f;

n = 0;
while (n < opt.maxit) && (opt.tol < ngrad/ngrad1)

   if opt.verbose
       if (mod(n, 30) == 0)
           fprintf('.');
       end
   end

   n = n+1;

   if mod(n,opt.Mds) == 0
       [f,grad] = fgx(x);
       if (fref - f) < opt.sigma*(dot(gradref(:),xref(:)-x(:)))
           beta = beta*opt.eta;
           disp('Diminishing scalars...')
       else
           xref = x;
           fref = f;
           gradref = grad;
       end
   end

   x(x == 0 & grad > 0) = 0;
   grad(x == 0 & grad > 0) = 0;
   gradold(x == 0 & grad > 0) = 0;

   % Compute BB-steps
   g = fftshift(ifft2(fft2(gradold).*Fps));  % Auxillary vector

   if (mod(n, 2) == 0)
       alpha = dot(gradold(:),gradold(:))/dot(g(:),g(:));
   else
       numer = dot(g(:),g(:));
       g = fftshift(ifft2(fft2(g).*FpsT));
       g(x == 0 & grad > 0) = 0;
       alpha = numer/dot(g(:),g(:));
   end

    x = x - beta*alpha * grad;      % Step along search path
    x(x < 0) = 0;

    [f,grad] = fgx(x);

    grad(x == 0 & grad > 0) = 0;

    gradold = grad;

    info.obj(n+1) = f;
```

```
    ngrad = norm(grad,'fro');;
    info.grad_norm(n+1) = ngrad;
    info.time(n) = toc(start_time);
end
if opt.verbose
    fprintf('\n\t'); disp([num2str(n) ' iterations in ' num2str(info.time(n)) ' seconds'])
end
info.iter = n;
end
```

fgp.m

```
function [x,info] = fgp(fun, PSF, b, x0, opt)
% FGP FAST GRADIENT PROJECTION ALGORITHM
%
% Usage:  [x,info] = fgp(@fun, PSF, b, x0, opt)
%
% Input:
%   @fun: Function handle with objective function and gradient
%   PSF: Point-spread function
%   b : Beamformer map
%   x0: Starting vector
%   opt: Struct with options - see deconv_opt.m for default settings
%
%
% Output:
%   x: Source distribution for all iterations
%   info: Struct with info about
%       - info.obj
%       - info.grad_norm
%       - info.time
%       - info.iter
%       - info.L
%
% Author: Oliver Lylloff
% Date: 5/2/14
% Latest revision: 5/2/14
%
% http://github.com/1oly/mscthesis
%
% Reference:
% Beck, Amir, and Marc Teboulle. 2009.
% 'A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.'
% SIAM Journal on Imaging Sciences 2 (1) (January): 183-202.
%
start_time = tic;

if opt.verbose
   fprintf('FGP ')
end

% Initialize variables
x = x0;
xold = x;
y = x;
t = 1;

% Precompute fft of PSF
Fps = fft2(PSF);
FpsT = fft2(rot90(PSF,2));

% Evaluate f(x0)
fgx = @(x) fun(PSF,b,x,Fps,FpsT);

if opt.backtrack
    L = 1;
```

```
    beta = 1.5;
else
    L = lipschitz(PSF,Fps);
end

% For n = 1
[fx,gradx] = fgx(x);
grady = gradx;
fy = fx;
gradx(x == 0 & gradx > 0) = 0;
ngradx = norm(gradx,'fro');

% Get info
info.grad_norm(1) = ngradx;
ngrad1 = ngradx;
info.obj(1) = fx;
info.sumx(1) = sum(x(:));

if opt.stop_cond
    cond = ngradx/ngrad1;
else
    cond = inf;
end

% Start iteration
n = 0;
while (n < opt.maxit) && (opt.tol < cond)

    if opt.verbose
        if (mod(n, 30) == 0)
            fprintf('.');
        end
    end

    n = n+1;

    x = max(0,y - (1/L)*grady);

    % Backtracking
    if opt.backtrack
        fx = fgx(x);
        m = 1;
        while fx > fy + dot(x(:)-y(:),grady(:))+(L/2)*norm(x-y,'fro')^2;
            L = m^(beta)*L;
            x = max(0,y - (1/L)*grady);
            fx = fgx(x);
            m = m + 1;
        end
    end

    if opt.descent
        if fgx(x)>fgx(xold)
            x = xold;
        end
    end

    tnew = (1+sqrt(1+4*t*t))/2;
```

```
    y = x + ((t-1)/tnew)*(x-xold);

    [fy,grady] = fgx(y);
    xold = x;
    t = tnew;

    if opt.stop_cond && (mod(n, opt.gradn) == 0)
        [fx,gradx] = fgx(x);
        gradx(x == 0 & gradx > 0) = 0;
        ngradx = norm(gradx,'fro');
        cond = ngradx/ngrad1;

        info.obj(n+1) = fx;
        info.grad_norm(n+1) = ngradx;
    elseif opt.stop_cond==0
        cond = inf;
    else
        info.obj(n+1) = fx;
        info.grad_norm(n+1) = ngradx;
    end

    if opt.backtrack
        info.L(n) = L;
    else
        info.L = L;
    end

    info.time(n) = toc(start_time);
end

if opt.verbose
    fprintf('\n\t'); disp([num2str(n) ' iterations in ' num2str(info.time(n)) ' seconds'])
end

info.iter = n;

end


function L = lipschitz(PSF,Fps)
% Estimate Lipschitz constant by power iteration
x = rand(size(PSF));
for k = 1:10
    x = fftshift(ifft2(fft2(x).*Fps))/norm(x,'fro');
end
    L = norm(x,'fro')^2;    % lambda(A'A) Assuming a symmetric matric A
end
```

csm.m

```
function [CSM,freqs,n] = csm(signal,window,noverlap,nfft,fs)

% CSM Compute cross-spectral matrix
%
%function [CSM,freqs] = csm(signal,window,noverlap,nfft,fs)
%
% Input:
%   signal: 2D array, size [N,M] with time data of N samples from M microphones
%   window: Window, e.g. Hann, Hamming, etc. with length = nfft
%   noverlap : Overlap in samples of time segments
%   nfft: Number of samples of each time-segment
%   fs: Sampling frequency
%
% Output:
%   CSM: Cross-spectral matrix
%   freqs: Frequency vector
%
% Author: Oliver Lylloff
% Date: 8/12/13
% Latest revision: 8/12/13
%
% http://github.com/1oly/mscthesis
%
% Inspired by FFTANALYSER TOOLBOX by Quentin Leclere
%

[N,M] = size(signal);
n = fix((N-noverlap)/(length(window)-noverlap));    % Number of frames

if round(nfft/2) == nfft/2
    n_spec = nfft/2 + 1;
else
    n_spec = (nfft+1)/2;
end

df = round(fs/nfft);  % Spectral resolution
freqs = (0:n_spec-1)*df;    % Frequency vector

% Initialize
frame = zeros(nfft,M);
spectrum = zeros(n_spec,M);
CSM = zeros(M,M,n_spec);
comp = sum(abs(window).^2)/nfft;
window = window/sqrt(comp);     % Power correction, such that window power = 1
pos = 1;

% First frame
for i = 1:M
    frame(:,i) = signal(pos:pos+nfft-1,i).*window;   % Current frame of x
end

h = waitbar(0,'Computing CSM');
for i = 1:n
    waitbar(i/n)
```

```
        % Compute spectrum:
        s = fft(frame)./nfft;
        spectrum(1,:) = s(1,:);     % DC component
        switch fix(nfft/2) == nfft/2
            case 1
                spectrum(2:nfft/2,:) = 2*abs(s(2:nfft/2,:)).*exp(1j*angle(s(2:nfft/2,:)));
                spectrum(nfft/2+1,:) = s(nfft/2+1,:);
            case 0
                spectrum(2:(nfft+1)/2,:) = 2*abs(s(2:(nfft+1)/2,:)).*exp(1j*angle(s(2:(nfft+1)/2,:)));
        end

        pos = pos + (nfft - noverlap);  % Update position

        % Assemble CSM:
        for j = 1:M
            % Update diagonal of cross-spectral matrix
            CSM(j,j,:) = CSM(j,j,:) + permute(abs(spectrum(:,j)).^2,[2 3 1]);
            % Update lower triangular Rmn
            for k = j+1:M
                temp = CSM(j,k,:) + permute(conj(spectrum(:,j)).*spectrum(:,k),[2 3 1]);
                CSM(j,k,:) = temp;
            end
            % Update new frame
            if i<n
                frame(:,j) = signal(pos:pos+nfft-1,j).*window;
            end
        end
    end
    close(h);

    % Average CSM
    CSM = CSM/n/2;       % rms value

    % Assemble upper triangular matrix
    for j = 1:M
       for k = j+1:M
           CSM(k,j,:) = conj(CSM(j,k,:));
       end
    end

    end
```

B Measurement Equipment

Type	Manufacturer	Model/Name	Description
Front-end	B&K	Frame Type 3560D 5×12 Input module Type 3038-B 5/1 I/O Controller module Type 7537	–
Computer	Dell	Latitude E6400 ATG Intel Core 2 Duo, 2.4 GHz 3.45 GB RAM	Used for measurements
Measurement software	B&K	PULSE LabShop v. 17.1.0	–
Computer	Apple	MacBook Intel Core 2 Duo, 2.4 GHz 8 GB RAM	Used for post-processing
Post-processing software	The Mathworks	MATLAB v.8.1.0.604 (R2013a) 64-bit	–
Noise generator	B&K	2× Noise Generator Type 1405	Mode: 20 kHz white noise
Amplifier	Custom made (DTU)		Used with *Source 1*
Amplifier	B&K	Power Amplifier Type 2706	Used with *Source 2*
Sound sources	B&K	2× Omnisource Type 4295 Denoted *Source 1* & *Source 2*	Nearly omni-directional response to approximate point sources
External noise source	B&K	Sound Source Reference HP 1001	–
Noise generator and power amplifier	B&K	Sound Power Source Type 4205	–
Microphone array	B&K	P60 D100 60 × 1/4" microphones Type 4935	Calibrated using TEDS database
Calibrator	B&K	Sound Level Calibrator Type 4301	94 dB at 1 kHz.
Fan	–	–	1330 RPM

Table B.1

Bibliography

[1] J. D. Maynard, E. G. Williams, and Y. Lee. Nearfield acoustic holography: I. Theory of generalized holography and the development of NAH. *J. Acoust. Soc. Am.*, 78(4):1395, 1985.

[2] J. Hald and J. J. Christensen. *Technical Review: Beamforming, No. 1*. Brüel & Kjær, April 2004.

[3] A. Torras-Rosell, S. Barrera-Figueroa, and F. Jacobsen. Sound field reconstruction using acousto-optic tomography. *J. Acoust. Soc. Am.*, 131(5):3786, 2012.

[4] J. Hald. Beamforming and Wavenumber Processing. In D. Havelock, S. Kuwano, and M. Vorländer, editors, *Handbook of Signal Processing in Acoustics*, pages 131–144. Springer New York, October 2008.

[5] D. H. Johnson and D. E. Dudgeon. *Array signal processing: Concepts and techniques*. Prentice Hall, 1993.

[6] F. Jacobsen and M. Juhl, P. *Fundamentals of general linear acoustics*. Wiley, 2013.

[7] T. F. Brooks and W. M. Humphreys. A deconvolution approach for the mapping of acoustic sources (DAMAS) determined from phased microphone arrays. *Journal of Sound and Vibration*, 294:856–879, July 2006.

[8] K. Ehrenfried and L. Koop. A comparison of iterative deconvolution algorithms for the mapping of acoustic sources. *American Institute of Aeronautics and Astronautics*, February 2008.

[9] J. L. Starck, E. Pantin, and F. Murtagh. Deconvolution in Astronomy: A Review. July 2002.

[10] W. Wallace, L. H. Schaefer, and J. R. Swedlow. A workingperson's guide to deconvolution in light microscopy. *BioTechniques*, 31(5):1076–1097, 2001.

[11] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. *IEEE J. Sel. Top. Signal Process.*, 1 (4):586–597.

[12] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. Institute of Physics Pub, 1998.

[13] R. P. Dougherty and R. W. Stoker. Sidelobe Suppression for Phased Array Aeroacoustic Measurements. *American Institute of Aeronautics and Astronautics*, 1998.

[14] P. C. Hansen. *Discrete inverse problems: Insight and Algorithms*. SIAM, 2010.

[15] P. C. Hansen. Deconvolution and Regularization with Toeplitz Matrices. *Numerical Algorithms*, 29(4): 323–378, 2002.

[16] J. Nocedal and S. J. Wright. *Numerical optimization*. Springerverlang, 2006.

[17] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[18] Y. Nesterov. Introductory lectures on convex optimization: A basic course. *Applied optimization*, 87, 2004.

[19] L. Eldén, L. Wittmeyer-Koch, and H. Bruun Nielsen. *Introduction to numerical computation: Analysis and MATLAB® illustrations*. Studentlitteratur, 2004.

[20] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[21] D. Kim, S. Sra, and I. S. Dhillon. A non-monotonic method for large-scale non-negative least squares. *Optimization Methods and Software*, 28(5):1012–1039, October 2013.

[22] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[23] Y-H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.*, 100(1):21–47, February 2005.

[24] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.*, 2(1):183–202, January 2009.

[25] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[26] L. Vandenberghe. Lecture Notes on EE236C - Optimization Methods for Large-Scale Systems. URL `http://www.seas.ucla.edu/~vandenbe/ee236c.html`.

[27] G. H. Golub and C. F. Van Loan. Matrix Computations. 1996.

[28] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.

[29] W. W. Hager and H. Zhang. Algorithm 851: CG DESCENT, a conjugate gradient method with guaranteed descent. *ACM Transactions on Mathematical Software*, 32(1):113–137, 2006.

[30] W. W. Hager and H. Zhang. A New Active Set Algorithm for Box Constrained Optimization. *SIAM J. Optim.*, 17(2), 2007.

[31] C. Li. A Conjugate Gradient Type Method for the Nonnegative Constraints Optimization Problems. *Journal of Applied Mathematics*, (4):1–6, 2013.

[32] D. Kim, S. Sra, and I. S. Dhillon. Tackling box-constrained optimization via a new projected quasi-newton approach. *Siam Journal on Scientific Computing*, 32(6):3548–3563, 2010.

[33] S-J Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale L1-regularized logistic regression. 2013.

[34] J. Gondzio. Matrix-free interior point method. *Comput Optim Appl*, 51(2):457–480, October 2010.

[35] J. L. Mueller and S. Siltanen. *Linear and nonlinear inverse problems with practical applications*. SIAM, 2012.

[36] D. L. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*. Springer, 2013.

[37] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring images: Matrices, spectra, and filtering*. SIAM, 2006.

[38] X. Zhou, F. Zhou, X. Bai, and B. Xue. A boundary condition based deconvolution framework for image deblurring. *Journal of Computational and Applied Mathematics*, 261:14–29, October 2013.

[39] W. A. Veronesi and J. D. Maynard. Nearfield acoustic holography (NAH) II. Holographic reconstruction algorithms and computer implementation. *J. Acoust. Soc. Am.*, 81:1307, 1987.

[40] A. Xenaki, F. Jacobsen, and E. Fernandez-Grande. Improving the resolution of three-dimensional acoustic imaging with planar phased arrays. *Journal of Sound and Vibration*, 331(8):1939–1950, April 2012.

[41] J. G. Nagy and D. P. O'Leary. Fast iterative image restoration with a spatially-varying PSF. *Advanced signal processing: algorithms, architectures, and implementations VII*, 3162:388–399, 1997.

[42] E. D. Andersen and K. D. Andersen. MOSEK. URL `http://www.mosek.com`.

[43] M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming, 2014. URL `http://cvxr.com/cvx`.