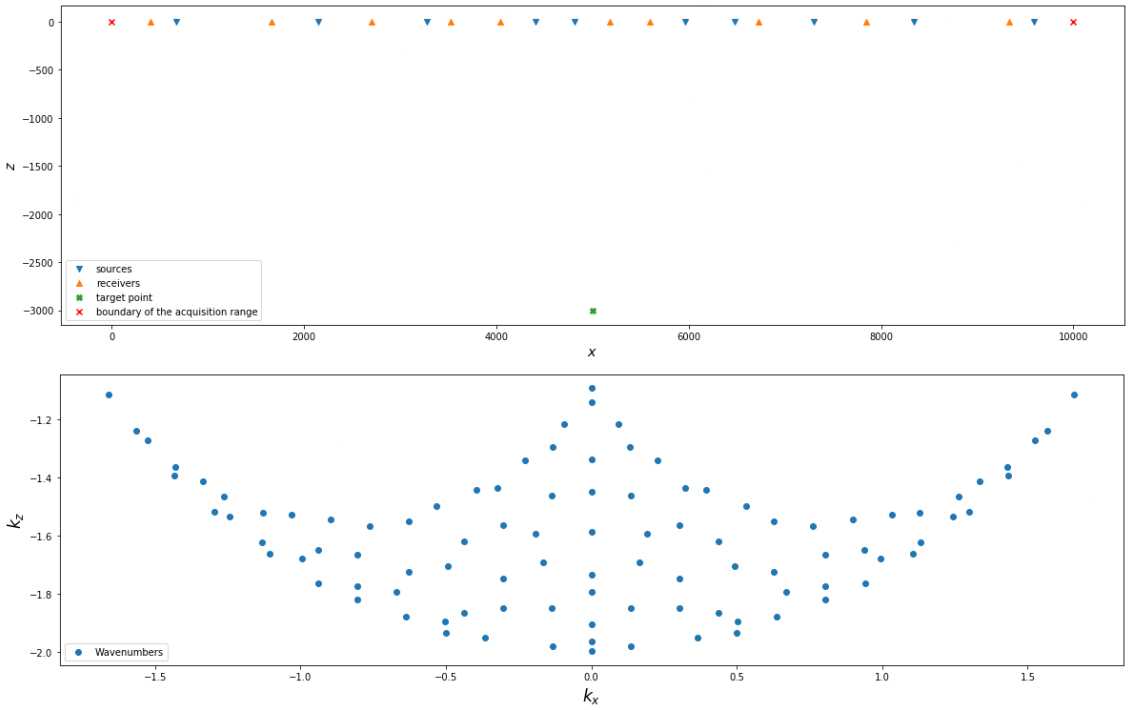
Optimal design for full waveform inversion: optimization of the wavenumber space coverage

Mohamed Amine Abdellaziz



MSIAM M2
MSc thesis
June 14, 2021

Supervisors
Ludovic Métivier
Edouard Oudet

Abstract

In this report we propose a new approach for the optimization of the **wavenumber coverage** in the plane wave approximation of the gradient in the context of the **Full Waveform Inversion**. This approach is based on the **optimal positioning** of sources and receivers on the surface. For a fixed number of devices and a prescribed acquisition range, the optimality criterion in our case is the regularity of the coverage inside the wavenumber cloud, the boundary of which is fixed by the extremities of the range.

We propose to express the problem as an instance of stable **Centroidal Voronoi Tessellation** and use a Newton-based method to minimize an energy function that takes as argument angles instead of positions on the surface. In our implementation we used a **BFGS** method on a simple instance for a homogeneous model of the subsurface. We noticed an improvement of the regularity of the coverage, but so far no optimal pattern seems to emerge from the solutions.

Acknowledgements

After two years in Grenoble as a student I would like to express my gratitude to the professors and staff members of the MSIAM program for their professionalism and dedication and who made it an experience worth coming for. I am in particular thankful to my supervisors Ludovic Métivier and Edouard Oudet for their patience and trust during this internship.

I wish also to thank my family who has always been there for me, no matter the time nor the distance, and especially my parents to whom I will forever be indebted.

A word also for my comrades Carlos, Dušan and Joel who have shared with me the time I spent at the LJK and who have been of great support. Thanks.

Contents

Chapter 1

Introduction

The Full Waveform Inversion is a geophysical imaging technique used to measure the physical properties of the Earth. It works by comparing observed data obtained by probing the subsurface using sources and receivers placed at the surface, and syntethic data obtained from a computer program. Using an optimization framework the method tries to find the model that approximates best the mechanical characteristics of the subsurface via the minimization of the difference between the observed and the calculated data. Started during the 1980's, the method has since been under continuous development. Taking advantage of the positive evolution of the computing power of the recent decades it is today widespread in industry and academia and continues to fuel the interest of the research community.

The FWI uses a gradient descent method to solve the minimization part. Under the right assumptions, the gradient of the cost function can be approximated by a sum of plane waves. Sirgue and Pratt proposed a method for the optimization of the wavenumber content of the plane waves using an optimal selection of temporal frequencies. It has also been shown that the positioning of the sources and receivers at the surface has an impact on the wavenumber content of the approximation. In this project we present a new approach to optimize this wavenumber coverage through the optimal design of the positioning of sources and receivers on the surface.

In this report, we will introduce the FWI method and the problematic in chapter 2. Next, we will study the wavenumber cloud and then present useful results from geometry, related to the Centroidal Voronoi Tesselation, that will provide us with the necessary tools to model and solve the problem. In chapter 4 we will talk about the implementation of the method and in chapter 5 we will present some results from a simple instance of the problem. We will conclude with a short summary of what has been done and propose some perspectives.

Chapter 2

Background

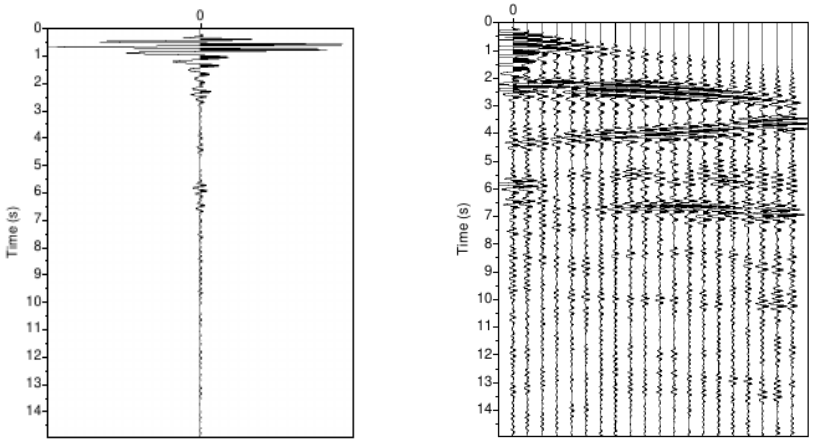
In this chapter we will present the problem in the frame of the Geophysical Imaging and its relation to the Full Waveform Inversion method.

2.1 Full Waveform Inversion

Given an area on the Earth's surface we want to get an estimate of the mechanical properties of its subsurface.

Suppose we arrange on the surface N_s sources and N_r receivers. The sources perturb the ground on the surface thus creating a wavefield propagating in the subsurface. The receivers register local variations in their neighbourhoods and will pick up the propagating wavefield. By processing the data registered by a single receiver we get a seismic trace recording the arrival times of the waves at the location of the receiver, as in 2.1(a). The traces produced by the receivers registering perturbations generated by a single source can be grouped into a seismogram 2.1(b) where they are laterally stacked according to the *offset*, or distance to the source, of their receivers.

Let us denote by $s = 1, \ldots, N_s$ the indices of the sources and by $r = 1, \ldots, N_r$ those of the receivers. The result of our configuration will be N_s seismograms, each associated to a source, recording the arrival times of the waves at the receivers during a period of time $t \in [0, T]$, where T represents the total recording time. Those seismograms represent the observed data $d_{obs} = (d_{obs,s})_{s=1,\ldots,N_s}$ obtained from the acquisition and are functions of time and space.



(a) A seismic trace (b) Seismogram

Figure 2.1: A seismic trace and a seismogram recording the arrival times (on the ordinate) of the waves emitted by a source at respectively one receiver and several receivers. In the seismogram, the abcissa corresponds to the distance of a receiver to the associated source, or offset, and thus is a function of space and time.

The mechanical properties of the subsurface can be expressed as a function $m(x)$ of space that

associates points x of the domain $\Omega \subset \mathbb{R}^n$ (representing the space of the subsurface, $n = 2$ or 3 denotes the dimension of the model) to one or several values representing the parameters we want to quantify, for instance the velocity of the medium, the density etc. Given a model $m \in \mathcal{M}$ of the subsurface we can compute the wavefield $u_s[m](x,t)$, generated by a perturbation made by the source s, by first solving an acoustic-wave equation [11, p. 232]

$$A(m)u_s(x,t) = \varphi_s(x,t)\,, \tag{2.1}$$

with $\varphi_s(x,t)$ the source term of the PDE and $A(m) = \left(\frac{1}{m^2}\frac{\partial^2}{\partial t^2} - \Delta\right)$ is a general wave propagation operator, modeling the propagation of mechanical waves within the subsurface.

We get the data returned by the receivers via a restriction operator $R[u](x,t) = \int_\Omega \delta(y - x)u(y,t)dy$. We apply it on the resulting wavefield $u_s[m]$ to retrieve the values at the locations of the receivers as follows

$$d_s[m](x_r,t) = R\left(u_s[m]\right)(x_r,t), \text{ for } r = 1,\dots,N\ . \tag{2.2}$$

Thus for parameters m we are then able by means of the forward equation (2.2) to compute synthetic seismograms $d_{cal}(m) = (d_{cal,s}(m))_{s=1,\dots,N_s}$ of the subsurface. What interests us is the inverse problem where given observed data d_{obs} we can deduce the model m^* that would yield $d_{cal}(m^*) \simeq d_{obs}$, in other terms the argument that reduces best the difference between d_{obs} and the calculed data $d_{cal}(m)$. We can express this optimization problem via the following PDE-constrained minimization problem [9, p. 66]

$$\begin{aligned} \min_{m\in\mathcal{M}} f(m) = \quad & \frac{1}{2}\sum_{s=1}^{N_s}\sum_{r=1}^{N_r}\int_0^T |d_{cal,s}(x_r,t) - d_{obs,s}(x_r,t)|^2 dt \\ \text{subject to:} \quad & A(m)u_s(x,t) = \varphi_s(x,t), \text{ for } s = 1,\dots,N_s\ , \\ & d_{cal,s}(x,t) = R[u_s](x,t), \text{ for } s = 1,\dots,N_s\ . \end{aligned} \tag{2.3}$$

The **Full Waveform Inversion** is a seismic imaging technique that uses a gradient descent method to solve 2.3 and match the calculated data through a computer model of 2.2 and observations. A fundamental component of the FWI is the gradient $\nabla f(m)$ of the misfit function which gives the descent direction.

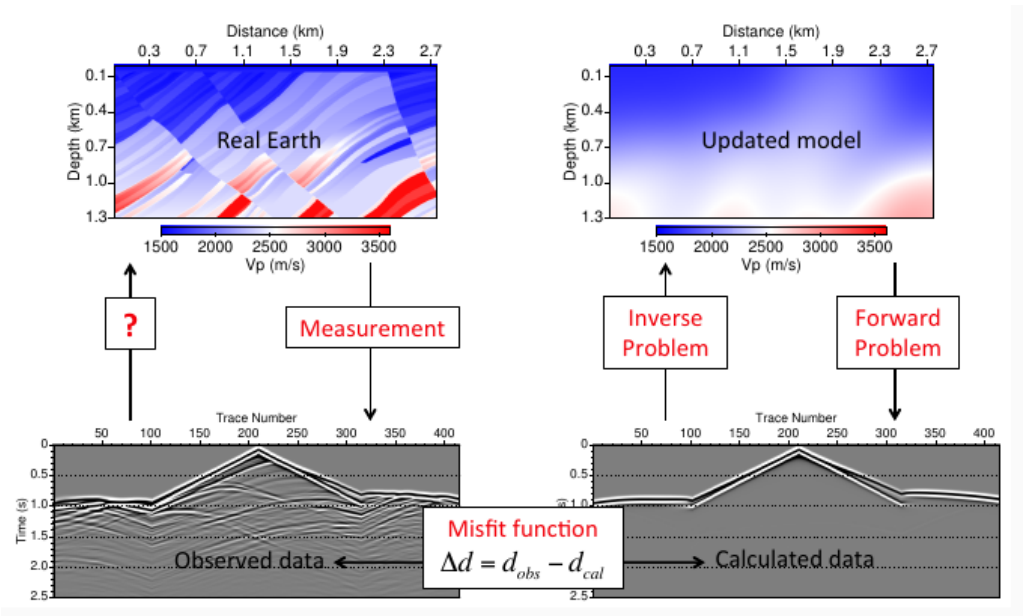


Figure 2.2: Illustration of the iterative process used in the FWI [9].

A way to compute $\nabla f(m)$ would be to use finite difference approximations that requires for each source s solving at least as many wave propagation problems (2.1) as there are points in the

discretized grid of the domain Ω which is impractical from the computational point of view. In comparison the **adjoint state method** [1] provides a way to compute the gradient which requires to solve only two wave equations per source s, the incident equation (2.1) and an adjoint equation.

The gradient ∇f can be expressed as the sum over the sources of the zero-lag time correlation[1] between the incident and adjoint wavefields u_s and λ_s

$$\nabla f(x) = \sum_{s=1}^{N_s} \left(\left[\frac{\partial A(m)}{\partial m} u_s(x,t) \right] \star \lambda_s(x,t) \right)(0) \,. \tag{2.4}$$

The adjoint wavefield λ_s is solution of the same equation 2.1 as u_s except that it is computed backward in time and the source term is the backprojection of the residuals from the receivers positions [2, p. 499]. For more details about the adjoint state method see appendix B.

2.2 Wavenumber content of the Gradient

To advance further let us simplify our model. We assume the following:

- The medium is homogeneous, with constant velocity c_0 everywhere in the subsurface,
- The sources and receivers are far away from the diffraction or target point x.

Also we will focus on a unique source-receiver pair (s, r) so we will drop the indices. Assuming these we can approximate the wavefields u and λ by plane waves. By neglecting amplitude and considering only null phases we would have

$$\begin{aligned} u(x,t) &= e^{ik_0(p_s.x+t)} \,, \\ \lambda(x,t) &= e^{ik_0(p_r.x+t)} \,, \end{aligned} \tag{2.5}$$

where k_0 is given by

$$k_0 = \frac{2\pi f_0}{c_0} \,. \tag{2.6}$$

The vectors p_s (resp. p_r) are the unit-length directions of the rays connecting the sources (resp. receivers) to x, for $s = 1, \ldots, N_s$ (resp. $r = 1, \ldots, N_r$). Recall from (2.1) that $A(m)$ is a wave propagation operator, so $\frac{\partial A(m)}{\partial m}$ is a second order time partial derivative and neglecting any constant we would have

$$\frac{\partial A(m)}{\partial m} u_s(x,t) \approx u_s(x,t) \,.$$

So we get

$$\begin{aligned} \nabla f(x) &\approx (u(x,t) \star \lambda(x,t))(0) \\ &= \left(\overline{u(x,-t)} * \lambda(x,t) \right)(0) \\ &\approx e^{ik_0 p_s \cdot x} \cdot e^{ik_0 p_r \cdot x} \,. \end{aligned}$$

We finally find that the gradient can be approximated by a plane wave of expression

$$\nabla f(x) \approx e^{ik_0(p_s+p_r)\cdot x} \,. \tag{2.7}$$

If we generalize this result to the pairs composed by all the sources and receivers we get an approximation of the gradient as a sum of plane waves of wavenumbers

$$k(s,r) = (k_x, k_z) = k_0(p_s + p_r) \,, \tag{2.8}$$

where k_x and k_z denote respectively the horizontal and vertical wavenumber components in the 2D approximation.

To improve the computation of the gradient we need to diversify the wavenumber content of its approximation 2.7. To that end, there are two ways suggested by Sirgue and Pratt:

[1] crosscorelation or autocorrelation of zero time shift.

- We can use a wider range of frequencies, which means shooting each source several times, each time with a different frequency f_0 in (2.6).
- We can use a range of different source-receiver pairs that sample the same diffraction point from different directions p_s and p_r.

The two methods can be used in concert [11, p. 234] and the authors proposed a strategy for selecting temporal frequencies. What we propose is to explore the second strategy.

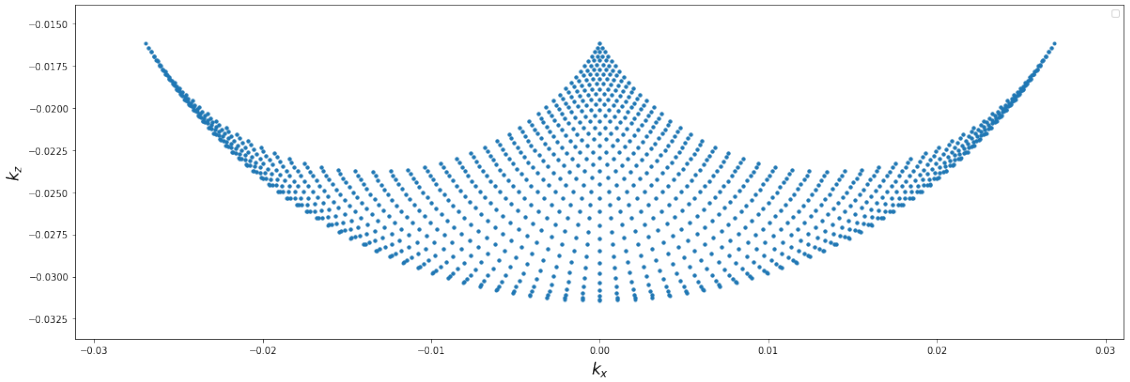


Figure 2.3: Each point in the cloud corresponds to a couple of sources and receivers. Here the mean frequency f_0 is equal to 5 Hz and the velocity c_0 is equal to 2000 m/s. An ensemble of $N_s = 51$ sources (resp. $N_r = 51$ receivers) have been regularly placed on the surface spanning a range from $start = 0$ km to $finish = 10$ km. The diffraction point $\mathbf{x} = (5000, -3000)$ is situated 5 kms from *start* on the horizontal axis and 3 kms deep in the subsurface.

In figure 2.3 is an example of wavenumber content obtained via a specific positioning of sources and receivers with a fixed frequency f_0 for all sources. The objective is to find the optimal disposition of the sources and receivers on the surface that will yield a cloud of points that covers **densely** and **regularly** an area as **large** as possible of the wavenumbers space.

Remark 1. In the remainder of the report we will take $f_0 = \frac{c_0}{2\pi}$ to normalize $k_0 = 1$. The results, including the optimized positioning of the acquisition devices, will still be valid for any frequency f_0 because k_0 is just a constant multiplier in (2.8).

Chapter 3

Optimization of the wavenumber space coverage

After having presented the problem in the previous chapter, we will discuss here the framework of its resolution. At first we will study the problem further and isolate the key elements that define it, then we will discuss results from geometry that will provide us with the necessary tools for the method we propose and which we will present at the end.

3.1 Study of the domain of the wavenumber space coverage

In this section we will investigate the cloud of wavenumber points generated by an **acquisition layout** (the positioning of the sources and receivers on the surface) and study its characteristics before talking about the optimality criteria.

To simplify the handling of wavenumbers we will express them in terms of angles and take advantage of trigonometric identities. As shown in figure 3.1 the direction vectors p_s and p_r can be expressed using the angles ϕ_s and ϕ_r they make with the vertical axis of direction $\vec{n} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ [6, Fig. 7] . We find the coordinates of those vectors to be

$$p_s = \begin{bmatrix} \sin\phi_s \\ -\cos\phi_s \end{bmatrix}, \qquad p_r = \begin{bmatrix} \sin\phi_r \\ -\cos\phi_r \end{bmatrix}. \tag{3.1}$$

From (2.8) and (3.1) we derive an angular formulation of the wavenumber $k(s,r)$ using the **incident** and **adjoint** angles ϕ_s and ϕ_r as follows

$$\begin{aligned} k(s,r) &= k_0(p_s + p_r) \\ &= k_0 \begin{bmatrix} \sin\phi_s + \sin\phi_r \\ -(\cos\phi_s + \cos\phi_r) \end{bmatrix} \\ &= k_0 \begin{bmatrix} 2\cos(\frac{\phi_s-\phi_r}{2})\sin(\frac{\phi_s+\phi_r}{2}) \\ -2\cos(\frac{\phi_s-\phi_r}{2})\cos(\frac{\phi_s+\phi_r}{2}) \end{bmatrix} \end{aligned}$$

and get the identity

$$k = 2k_0\cos(\frac{\theta}{2}) \begin{bmatrix} \sin(\phi) \\ -\cos(\phi) \end{bmatrix}, \tag{3.2}$$

where $\theta = \phi_s - \phi_r$ is the illumination angle and $\phi = \frac{\phi_s+\phi_r}{2}$ is the average of the two angles.

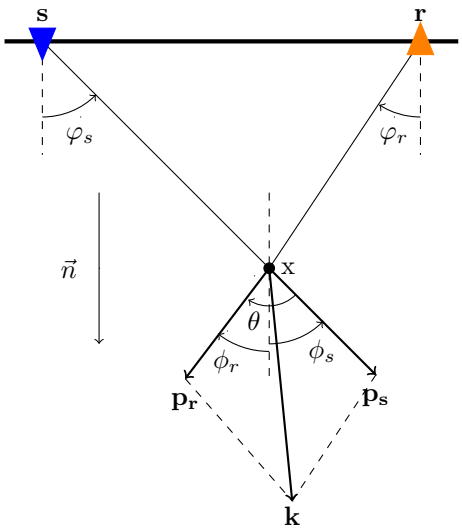


Figure 3.1: A source s and a receiver r are positioned at the surface. We denote by p_s and p_r the unit-length directions of the rays connecting respectively the source and the receiver to the diffraction point $\mathbf{x}$ in the middle along with the angles ϕ_s and ϕ_r that the rays make with the vertical axis. In the case of a non-homogeneous medium the angles φ's do not correspond to the angles ϕ's.

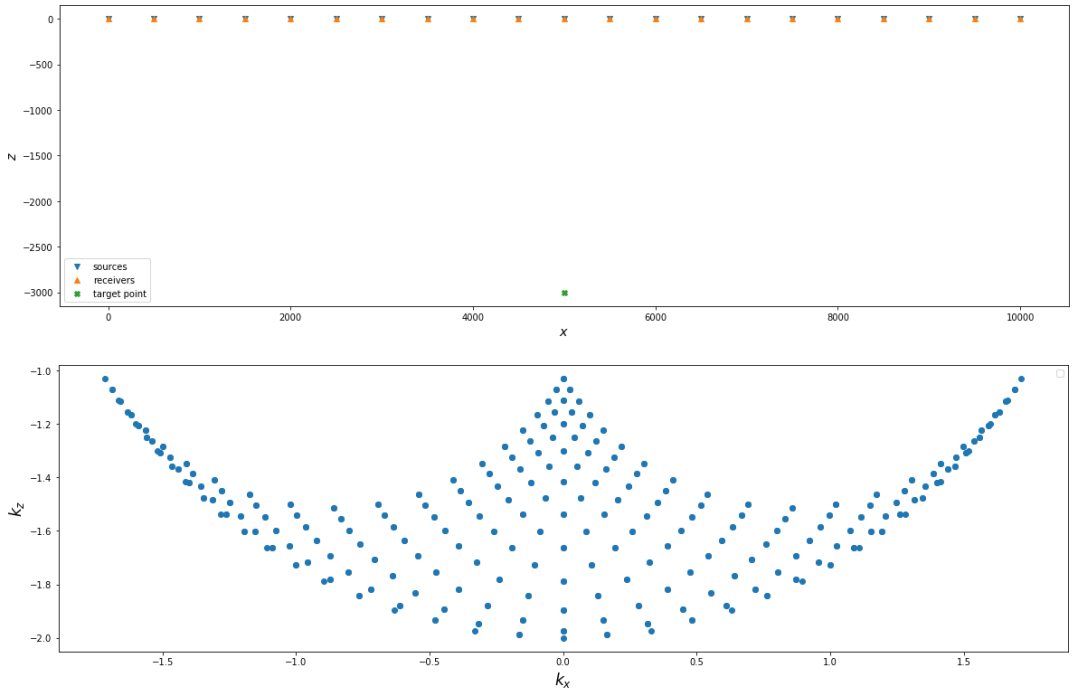


Figure 3.2: Reference acquisition layout and its wavenumber cloud.

Let us consider the same acquisition layout as in figure 2.3: a regular discretization of the range $[0, 10000]$ on the surface. However this time we will have $N = 21$ sources and receivers instead of 51 for the sake of visibility of the wavenumber points in the figures. In the remainder of the document this will be regarded as the reference acquisition layout and the associated wavenumber cloud will be the reference cloud, both shown in 3.2.

3.1.1 Size of the Area

We want to find the trajectory made by all the wavenumbers associated to a single source. To that end we will fix the location of the source and let the location of the receiver vary. Let us say that the source has a fixed angle $\overline{\phi_s}$, while the other angle ϕ_r varies between $-\phi_{max}$ and ϕ_{max}. Here ϕ_{max} denotes the largest angle the directional vectors p_s or p_r can make with the vertical and is related to the farthest source or receiver from the diffraction point. For any ϕ_r we would have

$$k = k_0 \begin{bmatrix} \sin\overline{\phi_s} + \sin\phi_r \\ -\cos\overline{\phi_s} - \cos\phi_r \end{bmatrix} = k_0 \begin{bmatrix} \sin\overline{\phi_s} \\ -\cos\overline{\phi_s} \end{bmatrix} + \begin{bmatrix} k_0\sin\phi_r \\ -k_0\cos\phi_r \end{bmatrix} = k_0 \begin{bmatrix} \sin\overline{\phi_s} \\ -\cos\overline{\phi_s} \end{bmatrix} + \begin{bmatrix} k_0\cos(\frac{\pi}{2} - \phi_r) \\ -k_0\sin(\frac{\pi}{2} - \phi_r) \end{bmatrix},$$

so

$$k = k_0 \begin{bmatrix} \sin\overline{\phi_s} \\ -\cos\overline{\phi_s} \end{bmatrix} + \begin{bmatrix} k_0\cos(\phi_r - \frac{\pi}{2}) \\ k_0\sin(\phi_r - \frac{\pi}{2}) \end{bmatrix}. \tag{3.3}$$

Consequently the wavenumbers associated with s form a circular arc of radius $r = k_0$, center $c = k_0\left(\sin\overline{\phi_s}, -\cos\overline{\phi_s}\right)$ and length $l = 2\phi_{max}k_0$. The same goes for a receiver r, with its angle replacing $\overline{\phi_s}$. In the figure 3.3 we see the circular arcs hidden in a wavenumbers cloud, along with their centers.

The domain $\mathcal{W}$ of all potential wavenumbers, considering a range on the surface that produces a maximum angle ϕ_{max}, can be defined as

$$\mathcal{W} = \bigcup_{\phi_s=-\phi_{max}}^{\phi_{max}} \left\{ k_0 \begin{bmatrix} \sin\phi_s \\ -\cos\phi_s \end{bmatrix} + \begin{bmatrix} k_0\cos(\phi_r - \frac{\pi}{2}) \\ k_0\sin(\phi_r - \frac{\pi}{2}) \end{bmatrix} : -\phi_{max} \le \phi_r \le \phi_{max} \right\} = \bigcup_{\phi_s=-\phi_{max}}^{\phi_{max}} \mathrm{Arc}_{\phi_s}, \tag{3.4}$$

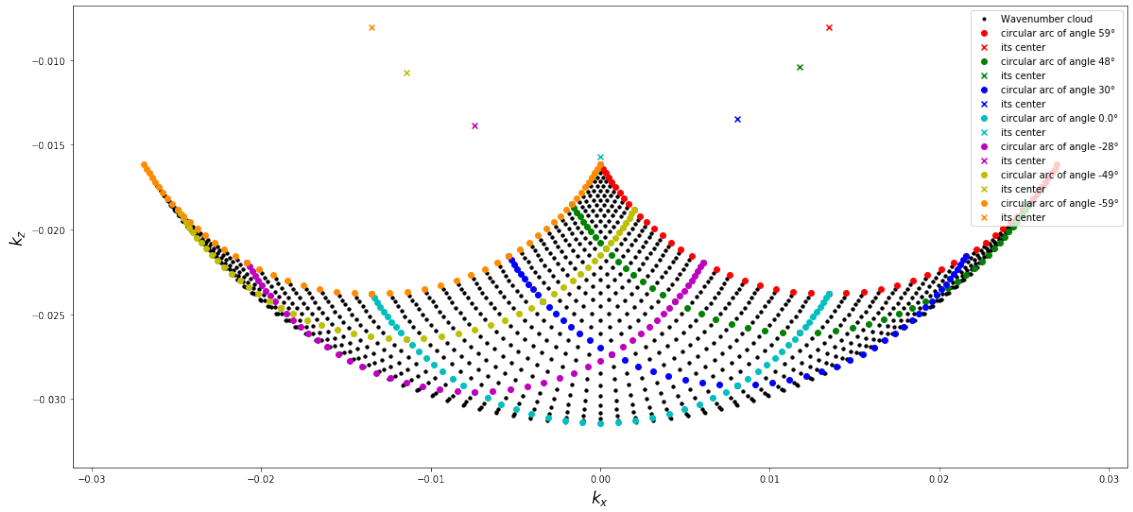


Figure 3.3: The circular arcs in the wavenumbers cloud of figure 2.3 associated to some of the sources in the acquisition layout. Notice how the centers of the circular arcs also form a circular arc above the wavenumber cloud.

where Arc_{ϕ_s} denotes the circular arc of radius $r = k_0$ and center $c = (k_0 \sin\phi_s, -k_0 \cos\phi_s)$, generated by angles $b = \phi_r - \frac{\pi}{2}$ spanning $[-\phi_{max} - \frac{\pi}{2}, \phi_{max} - \frac{\pi}{2}]$.

The domain $\mathcal{W}$ has an enveloppe constituted by the union of the side circular arcs $Arc_{-\phi_{max}}$ and $Arc_{\phi_{max}}$ with the bottom arc of equation

$$k(s,s) = 2k_0 \begin{bmatrix} \sin\phi_s \\ -\cos\phi_s \end{bmatrix} \text{ for } \phi_s \in [-\phi_{max}, \phi_{max}],$$

which is the circular arc of center the origin, radius $r = 2k_0$ and angle ϕ_s. The area covered by the wavenumber domain is limited by the maximum angle ϕ_{max} and thus its extension is conditionned by the farthest emplacement on the surface from the diffraction point: the largest the range on the surface covered by the acquisition is, the largest the area of the wavenumber domain will be.

For a fixed acquisition range on the surface it is possible to enlarge the maximum angle ϕ_{max} by considering the placement of sources and receivers inside wells at both ends of the range in the aim of adding degrees to ϕ_{max}. Figure 3.4 shows the enveloppe of the reference wavenumber cloud and compares it with the theoretical maxed out domain if we had $\phi_{max} = \frac{\pi}{2}$, which corresponds to an impossible infinite acquisition range on the surface.

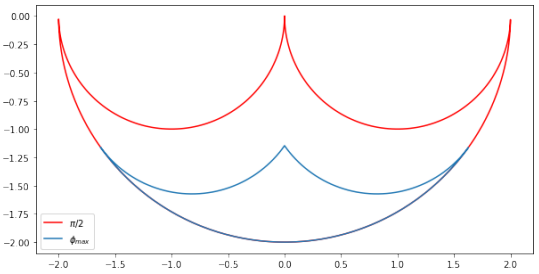


Figure 3.4: Enveloppe of the domain of 3.2 (in blue) and the maximum theoretical expansion possible of the domain (in red) if $\phi_{max} = \frac{\pi}{2}$.

3.1.2 Number of points

Each wavenumber is associated to a pair of source/receiver so for an acquisition layout of N_s sources and N_r receivers we should have $N_w = N_s \times N_r$ wavenumbers. So the density of the wavenumber cloud is linked to the density of the coverage of the acquisition range on the surface.

Notice in (2.8) and (3.2) that the function $k(s,r)$ is symmetric which implies that if a source/receiver pair (s,r) swapped their positions with each other they would produce the same wavenumber, i.e.

$$k(s,r) = k(r,s), \qquad \forall (s,r).$$

In the reference layout each source shares its emplacement with a receiver and the symmetry makes it that each wavenumber in the reference cloud is duplicated. For this reason, from the geometric point of view, it is better to avoid symmetric source/receiver pairs. Remark that in

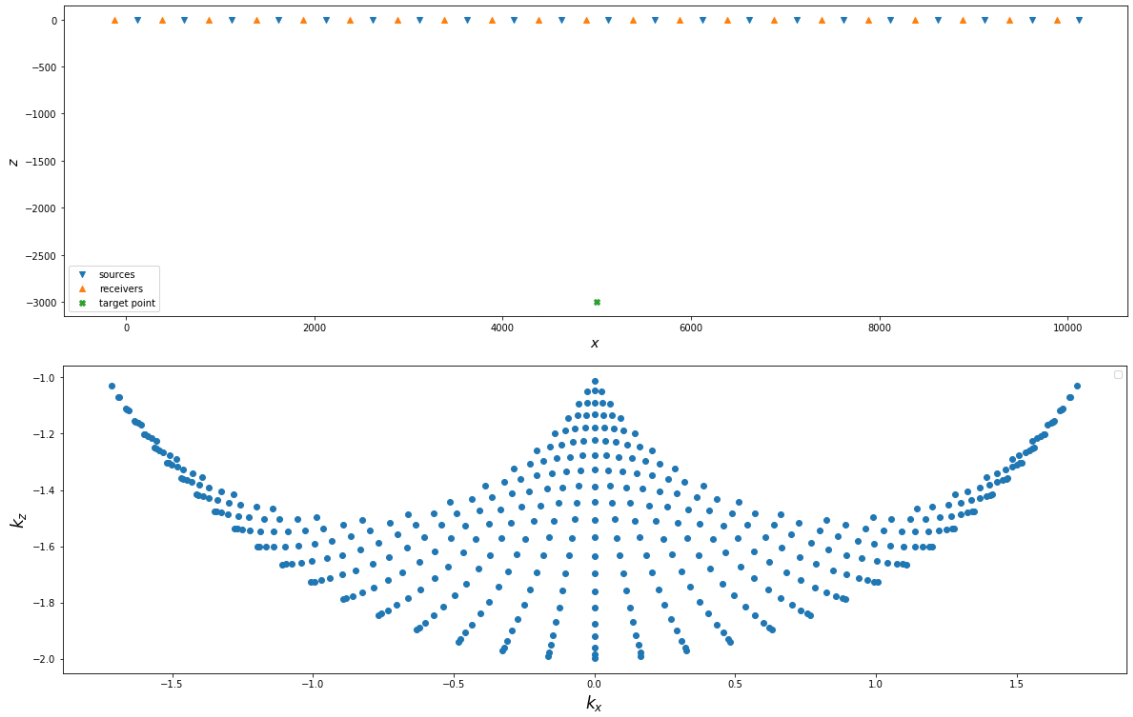


Figure 3.5: The sources and receivers of the reference layout have been shifted on opposite directions, both with a distance of 119 m.

figure 3.6 we acquired double the number of wavenumbers[1] with the same number of sources and receivers compared to 3.2, by using a *staggered* acquisition where the receivers are in the middle of the sources. By this we avoid any symmetric pair.

3.1.3 Regularity

As we have seen so far, (1) the size of the wavenumber domain is constrained by the range of the acquisition on the surface and (2) the number of points inside the domain is equal to $N_s \times N_r$, provided there are no duplicates. We conclude that the larger the range and the more acquisition devices the better it is. The last optimality criterion to consider is the regularity of the wavenumber space coverage.

We find that the distance between two wavenumbers $k = k(s, r)$ and $\hat{k} = k(\hat{s}, \hat{r})$ of angular increments $a_s = \phi_s - \phi_{\hat{s}}$ and $a_r = \phi_r - \phi_{\hat{r}}$ is given by the formula

$$\|k - \hat{k}\| = 2k_0 \sqrt{\sin^2(\frac{|a_s|}{2}) + \sin^2(\frac{|a_r|}{2}) + 2\sin(\frac{|a_s|}{2})\sin(\frac{|a_r|}{2})\cos\left[\phi_s - \phi_r + \frac{1}{2}(a_s - a_r)\right]}. \quad (3.5)$$

The wavenumbers that share the same source or receiver will form a circular arc as demonstrated in (3.3) so the distance between k and $\hat{k} = k(s+1, r)$ (resp. $\hat{k} = k(s, r+1)$) is equal to

$$\|k - \hat{k}\| = 2k_0 \sin\frac{a}{2},$$

where $a = |a_s|$ (resp. $a = |a_r|$). We deduce than a regular discretization of the space of incident and adjoint angles would yield a regular positioning of the wavenumbers along the circular arcs.

The formulae that connect the angles ϕ_s (resp. ϕ_r) and vectors $p_s = \frac{\mathbf{x}-\mathbf{s}}{\|\mathbf{x}-\mathbf{s}\|}$ (resp. $p_r = \frac{\mathbf{x}-\mathbf{r}}{\|\mathbf{x}-\mathbf{r}\|}$) are

$$\begin{aligned} \phi_s &= \arccos(p_s \cdot \vec{n}) \\ \phi_r &= \arccos(p_r \cdot \vec{n}). \end{aligned} \quad (3.6)$$

[1] distinct

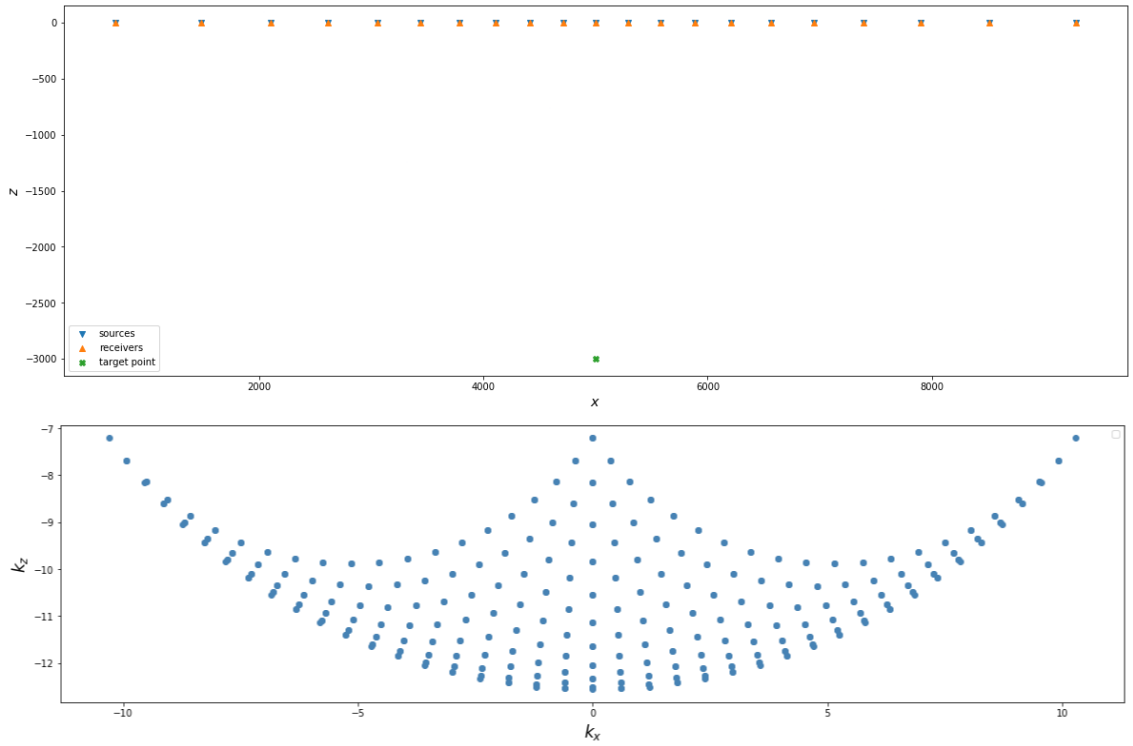


Figure 3.6: Acquisition layout and wavenumber clouds associated to a regular distribution of the angles ϕ_s and ϕ_r.

Because of the nonlinearity of the operations, a layout of the sources and receivers with a regular interval distance on the surface does not translate into a regular discretization in the space of incident and adjacent angles. Figure 3.6 shows the acquisition layout that would yield regular discretizations $(\phi_s)_{s=1,\dots,N}$ and $(\phi_r)_{r=1,\dots,N}$ of $[-\phi_{max}, \phi_{max}]$ along with the resulting wavenumber cloud. If on the top of the cloud the distances between points appear regular it is not the case for those on the bottom. This is due to the fact that the distance in (3.5) is not purely a function of the increments a_s and a_r but depends also on the difference $\phi_s - \phi_r$: the more ϕ_s and ϕ_r grow apart the more distant the wavenumber k will be from its neighbours. This is why in 3.6 the wavenumbers in the bottom, which correspond to close vectors p_s and p_r, seem to converge to an accumulation point.

Should the range boundaries and the numbers N_s and N_r be imposed parameters we are left with the positioning of the sources and receivers (equivalent to the choice of incident and adjoint angles) within the prescribed range to play on the regularity of the coverage. We have seen that simply choosing a regular spacing of the angles is not enough to ensure the regularity of the spacing between adjacent wavenumbers. To solve our problem we are to frame it into an optimization problem and solve it using rigourous and proven techniques of mathematical programming.

The regularity aspect can be seen as an instance of **Centroidal Voronoi Tessellation** (CVT) : we seek a regular distribution of the wavenumbers inside the domain $\mathcal{W}$, with the additional constraint of interdependency through the sources and receivers. In the next section we will talk about the CVT and present some relevant results from [7].

3.2 Centroidal Voronoi Tessellation

Let $\Omega \subset \mathbb{R}^N$ be a bounded domain and let $\|.\|$ denote the Euclidean norm on R^N. Let $X = (x_i)_{i=1,\dots,n} \subset \Omega$ be a set of n distinct points that we will call seeds. The Voronoi cell of a seed x_i is defined as the set of points in Ω to which x_i is the closest of all the seeds,

$$\Omega_i = \{x \in \Omega \quad | \quad \|x - x_i\| \leqslant \|x - x_j\|, \forall j \neq i\} . \tag{3.7}$$

The Voronoi cells $\{\Omega_i\}_{i=1,\dots,n}$ form a Voronoi tessellation[2] of Ω. A tessellation of Ω is a collection of subsets $(V_i)_{i=1,\dots,n}$ of Ω that verify (a) $V_i \cap V_j = \emptyset$ for $i \neq j$ and (b) $\bigcup_{i=1}^{n} \overline{V}_i = \overline{\Omega}$.

Given a density function $\rho(x)$ defined on $\overline{\Omega}$, the mass centroid of a subset $S \subset \Omega$ is given by

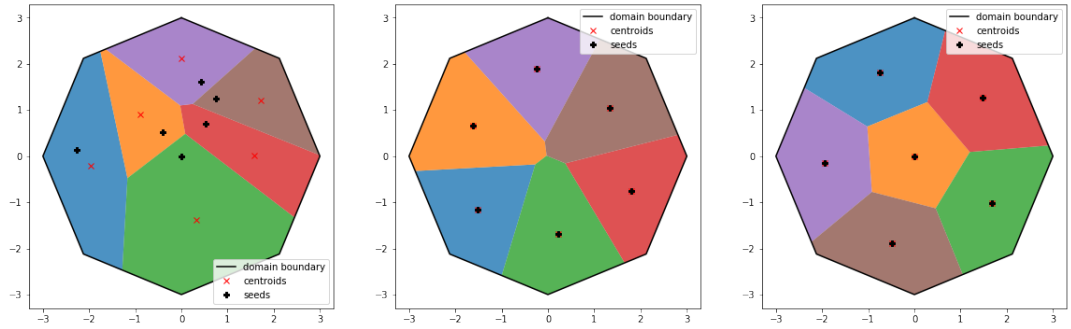
$$c(S) = \frac{\int_S \rho(x) x d\sigma}{\int_S \rho(x) d\sigma}. \tag{3.8}$$

Here we have all the elements to construct a CVT.

Definition 1 ([7, Definition 1])**.** The Voronoi tessellation $\{\Omega_i\}_{i=1,\dots,n}$ is a **Centroidal Voronoi Tessellation** if for all $i = 1, \dots, n$, we have $x_i = c_i$; that is, each seed coincides with the centroid of its Voronoi cell.

Arbitrary seeds in Ω do not correspond to the centroids of their Voronoi cells so a CVT is a special case of Voronoi tessellation. Due to its specificity CVT has a lot of applications in many fields, see [3, section 2] for some examples.

Given a domain Ω, the CVT problem consists in finding the seeds $\{x_i\} \subset \Omega$ that would generate a CVT of the domain. The solution is in general not unique [3, p. 639]. Figure 3.7 illustrates this with two CVT's of the same domain. The CVT can be seen from two perspectives [7, p. 5]: the Geometric characterization and the Variational one.



(a) A Voronoi tessellation (b) A Centroidal Voronoi Tessellation (c) A different CVT

Figure 3.7: Examples of a Voronoi Tessellation (left) and a CVT of an hectagon (center) with a constant density function ρ. Another CVT of the hectagon is also possible as illustrated in the example on the right.

3.2.1 Geometric characterization

A CVT can be understood from definition 1 to be a solution of the system of nonlinear equations of the form

$$x_i = c(\Omega_i)\,, i = 1, \dots, n\,, \tag{3.9}$$

which relies on the concept of centroid. The classic method to produce a CVT is Lloyd's algorithm, an iterative method which starting from an initial Voronoi tessellation of the domain iteratively updates the seeds with the centroids of their Voronoi cells. Lloyd's method is named after Stuart P. Lloyd [8] and algorithm 1 gives an outline of it.

[2] also Voronoi diagram

Algorithm 1: Lloyd's method [3, p. 657]

Input: A domain Ω, integer $n > 0$, density function ρ defined on $\overline{\Omega}$.
Output: A CVT of Ω (with n seeds).

```
Initialize the seeds {x_i}_{i=1,...,n} and construct the associated Voronoi Tessellation
  {Ω_i}_{i=1,...,n};
while Stopping criteria not met do
    Compute the centroids c_i of Ω_i, i = 1, ··· , n;
    Update the seeds with the centroids: x_i ← c_i, i = 1, ..., n;
    Construct the Voronoi Tessellation {Ω_i}_{i=1,...,n} of Ω;
end
return {Ω_i}_{i=1,...,n}
```

The map $T : X \longrightarrow T(X) = \{c(\Omega_i)\}_{i=1,\dots,n}$ that associates the seeds $X = \{x_i\}_{i=1,\dots,n}$ to the centroids of their Voronoi cells is referred to as the **Lloyd map** [3, p. 658]. We easily notice that the seeds X of a CVT are a fixed point of T which makes Lloyd's method a fixed point iteration

$$X^{(n+1)} = T(X^{(n)}). \tag{3.10}$$

Contrary to what its predominent use would suggest, Lloyd's method has a linear convergence and is considered to be of poor efficiency as we will clearly see later. With the growing interest in the CVT and the research done on it, new and better methods have been explored, among them Newton-based optimization methods introduced by Liu et al..

3.2.2 Variational characterization

Finding a CVT can be redefined in the frame of an optimization problem. We can associate an energy function to the CVT via the result:

Theorem 1 ([3, Propositions 3.1 and 3.2])**.** *Given $\Omega \subset \mathbb{R}^N$, a positive integer n and a density function $\rho(.)$ defined on $\overline{\Omega}$, let $X = \{x_i\}_{i=1,\dots,n}$ denote any set of n points belonging to Ω. Let*

$$F(X) = \sum_{i=1}^{N} \int_{y \in \Omega_i} \rho(y) \|y - x_i\|^2 dy\,. \tag{3.11}$$

A necessary condition for F to be minimized is for the seeds in X to generate a CVT of Ω.

Every minimizer of F is a CVT but the reverse is not true: it's not every CVT that minimizes F. However we can still characterizes a CVT by the energy function F. If we denote by $M(X)$ the diagonal matrix of elements $\{m_i\}_{i=1,\dots,n}$, the masses of the Voronoi cells $\{\Omega_i\}_{i=1,\dots,n}$,

$$m_i = \int_{\Omega_i} \rho(y) dy, \qquad i = 1, \cdots, n\,, \tag{3.12}$$

then the gradient of F is given by the formula [3, Proposition 6.2]

$$\nabla F(X) = 2M\left(X\right)\left(X - T\left(X\right)\right). \tag{3.13}$$

Since CVTs are fixed points of T they are also stationary points of the gradient ∇F and from here we obtain a second, more in depth, definition of a what constitutes a CVT.

Definition 2. [7, Definition 2] A **Centroidal Voronoi Tessellation** of a closed domain Ω with n seed points $X = \{x_i\}_{i=1,\dots,n}$ is the Voronoi tessellation given by the seeds X_0 which is a critical point of the CVT energy function $F(X)$. Furthermore, a CVT is called a **stable**

CVT if X_0 is a local minimizer of $F(X)$, and it is called an **optimal CVT** if X_0 is a global minimizer of $F(X)$.

In short optimal, CVT's are the results of optimization programs of the form

$$\min_{X=(x_i)_{i=1,\dots,n}\subset\Omega} F(X)\,. \tag{3.14}$$

Liu et al. argue that the function F has been for a long time wrongly believed to be non-smooth (C^0 and not even C^1) due to its complicated piecewise nature and this impeded impactful progress in finding more computationaly efficient alternatives to Lloyd's method and deterred any research in the direction of robust descent methods. The authors denoted the space of sets of seeds of n distinct elements by

$$\Gamma_C = \left\{ X = \{x_i\}_{i=1,\dots,n} \subset \Omega \mid x_i \neq x_j \text{ for } i \neq j \right\},$$

and provided conditions on the smoothness of the energy function F in the 2D case[3] via the theorem:

Theorem 2. *The 2D CVT function is C^2 in Γ_C if Ω is convex and the density function $\rho(X)$ is C^2.*

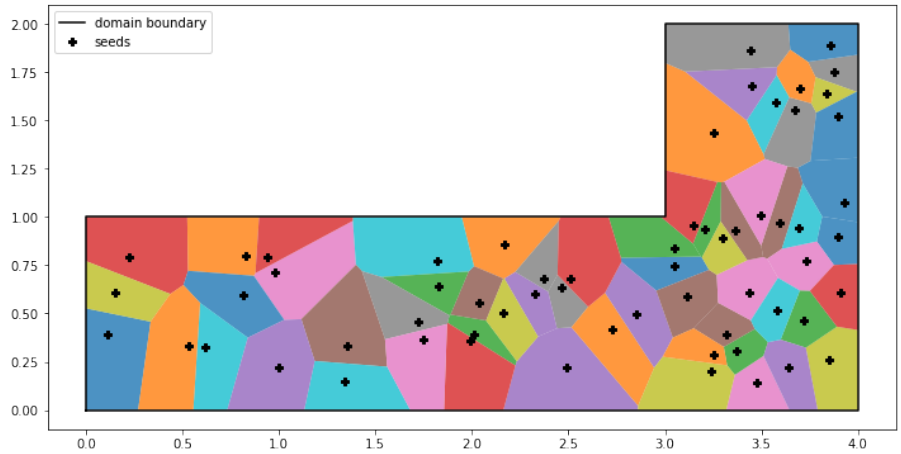


Figure 3.8: Illustration of remark 2: with enough seed points no ridge of the Voronoi cells is parallel to the boundary of the domain.

Remark 2. The authors explained that the C^2 smoothness is lost when a continuous part of $\partial\Omega$ is contained inside a ridge of a Voronoi cell. In this case F is C^1. In practice, with a sufficient number of seeds no ridge of Voronoi cell is parallel to the boundary $\partial\Omega$ in most configurations. Thus during the optimization procedure the energy function is C^2 in the neighbourhood of the visited iterations $X^{(n)}$.

With those elements in mind we can apply a gradient descent of the form

$$X^{(n+1)} = X^{(n)} - \alpha_n B(X^{(n)}) \nabla F(X^{(n)})\,, \tag{3.15}$$

where α_n is the step size to carefully choose at each iteration and $B(X^{(n)})$ is a positive definite matrix.

In light of this new result and formula (3.13) to compute the gradient, Liu et al. propose to take advantage of the C^2 smoothness of F and use a Newton method of quadratic convergence for the CVT in the 2D and 3D case so (3.15) would be

[3]They also found the same result for the 3D case

$$X^{(n+1)} = X^{(n)} - H(X^{(n)})^{-1}\nabla F(X^{(n)})\,, \tag{3.16}$$

with $\alpha_n = 1$ and $B = H^{-1}$, where H denotes the Hessian of F.

Although a method exists for computing the second order partial derivatives of the CVT energy function [7, p. 11], constructing the Hessian matrix and inversing it at every iteration of the Newton method would be costly for large-scale instances of the CVT. So the authors propose to use quasi-Newton methods that approximates the Hessian, like **BFGS**. To circumvent the large memory space requirement of the BFGS to store the dense inverse Hessian matrix they settle for the **L-BFGS** (limited memory BFGS) variant as an alternative.

To take advantage of the availabilty of a formula for the Hessian, they also propose to test the P-L-BFGS method, alongside the L-BFGS, which updates the Hessian $\tilde{H}^0$ in L-BFGS with the current exact Hessian every fixed numer of iterations.

From 3.10 and 3.15 we conclude that Lloyd's method can also be expressed as a gradient descent method with step size $\alpha_n = \frac{1}{2}$ and $B(X^{(n)}) = M(X^{(n)})^{-1}$. We see here the problem with this method: it is of order 1 thus has a linear convergence; coupled with the fixed step size α_n it has no mechanisms to avoid saddle points. This unveils the overall inefficiency of the Lloyd's method and why more robust and efficient methods are needed.

After running tests on simple geometric examples (polygons and polyhedrons), both with constant and varying density ρ [7, section 5], and to instances of mesh surface [7, section 6] using the Newton-based optimization methods and the Lloyd's method they arrived at the conclusion that the L-BFGS and P-L-BFGS methods are significantly faster than Lloyd's method.

Now that we layed down the basics of the CVT, let us apply it and express our coverage problem as a minimization problem.

3.3 Regularity of the wavenumber coverage as an instance of CVT

Let us go back to our wavenumbers. For ϕ_{max} fixed, corresponding to a prescribed range of acquisition, if we consider the associated $\mathcal{W}$ as the domain Ω in $\mathbb{R}^2$ with constant density $\rho(x) = 1$ then the function F applied to the wavenumber points expresses the regularity of the coverage of $\mathcal{W}$. So we aim at finding the positioning of sources and receivers on the surface that would yield the points in the wavenumber space which will produce a tessellation of the domain $\mathcal{W}$ as close to a stable CVT as the interdependency will allow.

Lloyd's method cannot be applied to our problem: the seeds are interdependant via their parametrization by the sources and receivers, so updating one wavenumber point will inevitably impact other poinst as illustrated by figure 3.9. Instead, the main idea is to use the optimization formulation (3.14) and inject the source-receiver parametrization.

As in formula (3.2) we will define a function that links a pair source-receiver (s, r) to a wavenumber via their respective angles ϕ_s and ϕ_r,

$$\begin{array}{rccc} \mathrm{k}: & [-\phi_{max}, \phi_{max}]^2 & \longrightarrow & \mathbb{R}^2 \\ & (\phi_s, \phi_r) & \longmapsto & k(s,r) \end{array}\,.$$

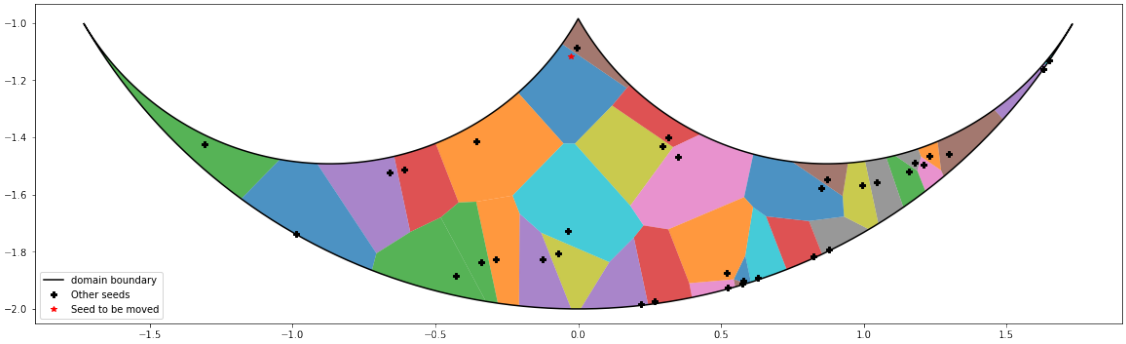
We also define a function that gives the set of wavenumbers generated by incident and adjoint angles $S = (\phi_s)_{s=1,\dots,N_s}$ and $R = (\phi_r)_{r=1,\dots,N_r}$

$$\begin{array}{rccc} \mathrm{K}: & [-\phi_{max}, \phi_{max}]^{N_s+N_r} & \longrightarrow & \left(\mathbb{R}^2\right)^{N_s \times N_r} \\ & (S, R) & \longmapsto & \left(k(s_i, r_j)\right)_{\substack{j=1,\cdots,n \\ i=1,\cdots,m}} \end{array}\,. \tag{3.17}$$

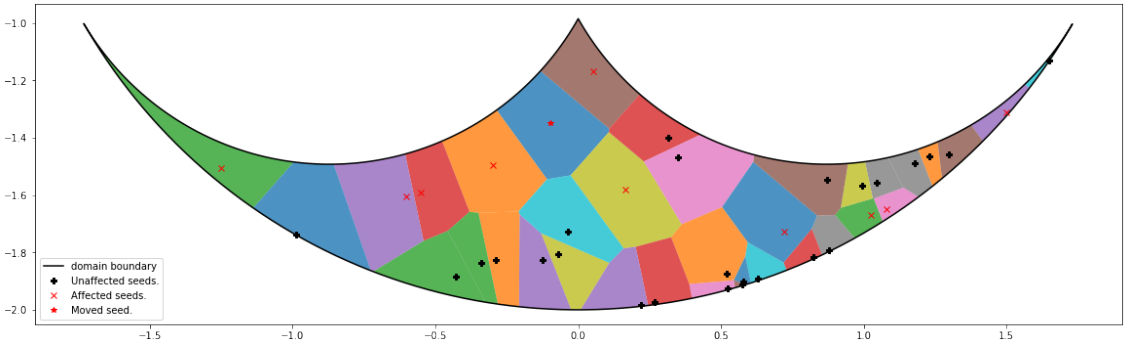
The composition of K and the energy function F yields a new energy function F_ϕ that assesses the fitness of an acquisition layouts by the criterion of regularity of the distribution of the points in the produced wavenumber cloud,

$$F_\phi = F \circ K : [-\phi_{max}, \phi_{max}]^{N_s+N_r} \xrightarrow{K} \left(\mathbb{R}^2\right)^{N_s \times N_r} \xrightarrow{F} \mathbb{R}^+/,\,, \tag{3.18}$$

and from here we obtain a minimization formulation of the regularity problem described in subsection 3.1.3:



(a) A Voronoi tesselation of $\mathcal{W}$.



(b) Result of applying an operation of Lloyd's method.

Figure 3.9: An attempt on applying the update of a seed by the centroid of its Voronoi cell for the problem of regularity of the wavenumber cloud; the result is that the positions of the source and receiver associated to the moved seed changed and this impacted the other seeds depending on those same source and receiver. Thus Lloyd's method is impractical for our problem.

$$\min_{(S,R)} F_\phi(S,R). \tag{3.19}$$

Because of the non-convexity of $\mathcal{W}$ in figure 3.4 the cost function is C^1 but taking into account remark 2 we can still apply Newton-based methods. Using the chain rule we obtain the formula of the gradient

$$\nabla F_\phi(x) = J_K(x)^T \cdot \nabla F\left(K(x)\right) . \tag{3.20}$$

where $J_K(x) \in \mathbb{R}^{2(n\times m)\times(m+n)}$ is the Jacobian of K and ∇F is the gradient defined in 3.13. At the end we will use a gradient-based method to solve the optimization problem.

Remark 3. Because the seeds need to be distinct points in the definition (3.7) of a Voronoi cell to generate a correct Voronoi tessellation, we are to ensure that no wavenumber is duplicated in the wavenumber space. To that end we will avoid symmetric source-receiver pairs in the initialization of (3.19) as discussed in 3.1.2.

In the next chapters we will apply our approach and present some results.

Chapter 4

Implementation

The implementation of the methods seen in this project has been done in **Python**, using `numpy` [5] and `scipy` [13] for numerical computing along some specific librairies and scripts for other aspects of the problem. The project is accessible at this repository.

4.1 Wavenumbers

A function `wavenumbers_angles` has been written that takes as input a set of incident and adjoint angles and generates the points of the wavenumber cloud using formula (3.2). To find the acquisition layout on the surface that would produce the angles a function `layout2angles` converts the positions (x, z) of a source or receiver to an angle ϕ via the formula

$$\phi = \arccos\left(\frac{z_{\mathrm{x}} - z}{\sqrt{(x_{\mathrm{x}} - x)^2 + (y_{\mathrm{x}} - y)^2}}\right),$$

where $(x_{\mathrm{x}}, z_{\mathrm{x}})$ are the coordinates of the diffraction point x. The function is mainly used for convert initial acquisition layouts into angles to feed them to the optimization program. The function *angles*2*layout* does the opposite via the formulae

$$x = x_{\mathrm{x}} + z_{\mathrm{x}} \tan(\phi) \quad \text{and} \quad z = 0\,,$$

and is used to retrieve the positions of the sources and receivers on the surface corresponding to the optimal angles at the end of the execution of the program.

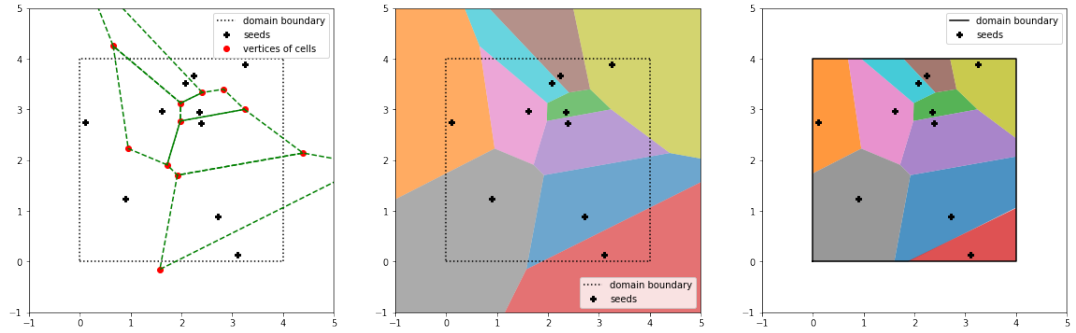
The wavenumber cloud of an acquisition layout is obtained by running `wavenumbers_angles` with the angles returned by `layout2angles` as input. Plot functions produces the wavenumber clouds and acquisition layout plots like the ones seen in figures of section 3.1.

4.2 Voronoi tessellation

The function `scipy.spatial.Voronoi` takes as input an array of points and generates Voronoi tessellations of the space $\mathbb{R}^N$ using the points as seeds. Because the function does not consider a domain it only provides vertices of the Voronoi regions that are intersections of ridges of the regions, the rest are considered stretching to infinity 4.1(a).

To complete the Voronoi diagram a Python script has been used [12], that reconstructs the infinite Voronoi regions outputted by `scipy.spatial.Voronoi` in 2D into a finite diagram 4.1(b) but not inside a desired domain as it constructs the missing vertices by intersecting the ridges stretching to infinity with a circle the radius of which is a parameter to fix.

To get the Voronoi tessellation of a domain we define or approximate the latter by a polygon and we use the `intersection` method made available by Shapely [4], a Python library for manipulating planar geometric object; we create polygonal Voronoi regions using the previous script with a sufficiently large radius in the option and we intersect them with the domain 4.1(c). Concerning the domain $\mathcal{W}$ in the wavenumber space, the circular arcs that compose it have been approximated by a large number of line segments.



(a) Voronoi diagram produced by Scipy

(b) Using Virtanen's script.

(c) Intersection with the domain via Shapely.

Figure 4.1: Creation process of a Voronoi tessellation of a domain.

Lloyd's method has been implemented for the CVT and has been used to check the correct implementation of the variational approach, as discussed in section 4.4.

4.3 Integration over polygons

To compute the area of a Voronoi cell as in (3.12), we use the divergence theorem to get

$$m_i = \int_{\Omega_i} dx\, dy = \frac{1}{2} \int_{\Omega_i} \nabla \cdot \begin{bmatrix} x \\ y \end{bmatrix} dx\, dy = \frac{1}{2} \int_{\partial\Omega_i} \begin{bmatrix} x \\ y \end{bmatrix} \cdot \vec{n}\, ds\,. \tag{4.1}$$

Considering the cell to be a polygon we can parametrize the line segments C made by two adjacent vertices p_1 and p_2 by $C(t) : t \longmapsto t p_1 + (1-t)\, p_2$ for $t \in [0,1]$. Summing over the line segments C forming the polygon we get the area via the line integrals

$$m_i = \frac{1}{2} \sum_C \int_0^1 [(1-t)\, p_1 + t p_2] \cdot \vec{n}\, \|C\|\, dt\,. \tag{4.2}$$

We apply the same technique and the same parametrization to get a way to numerically evaluate $T(X)$ and $F(X)$; we find for the centroid in (3.8)

$$c_i = \frac{1}{m_i} \int_{\Omega_i} \mathrm{x}\, d\mathrm{x} = \frac{1}{m_i} \begin{bmatrix} \int_{\Omega_i} x\, dx\, dy \\ \int_{\Omega_i} y\, dx\, dy \end{bmatrix} = \frac{1}{2m_i} \begin{bmatrix} \int_{\Omega_i} \nabla \cdot \begin{bmatrix} x^2 \\ 0 \end{bmatrix} dx\, dy \\ \int_{\Omega_i} \nabla \cdot \begin{bmatrix} 0 \\ y^2 \end{bmatrix} dx\, dy \end{bmatrix} = \frac{1}{2m_i} \begin{bmatrix} \int_{\partial\Omega_i} \begin{bmatrix} x^2 \\ 0 \end{bmatrix} \cdot \vec{n}\, ds \\ \int_{\partial\Omega_i} \begin{bmatrix} 0 \\ y^2 \end{bmatrix} \cdot \vec{n}\, ds \end{bmatrix}. \tag{4.3}$$

For the energy function (3.11), denoting x_i the seed of the Voronoi cell Ω_i, we would have

$$\int_{\Omega_i} \|\mathrm{x} - \mathrm{x}_i\|^2\, d\mathrm{x} = \int_{\Omega_i} \|\mathrm{x}\|^2 + 2\,\mathrm{x} \cdot \mathrm{x}_i + \|\mathrm{x}_i\|^2\, d\mathrm{x} = \int_{\partial\Omega_i} \left(\frac{1}{3} \begin{bmatrix} x^3 \\ y^3 \end{bmatrix} + \begin{bmatrix} x_i x^2 \\ y_i y^2 \end{bmatrix} + \frac{\|\mathrm{x}_i\|}{2} \begin{bmatrix} x \\ y \end{bmatrix} \right) \cdot \vec{n}\, ds. \tag{4.4}$$

The function `scipy.integrate.quad` has been used to compute the line integrals.

To test the correct implementation of formulae (4.1) and (4.3) we generate a Voronoi tessellation of some polygonal domain using random seeds and we compute the masses and centroids of the Voronoi cells using the line integrals and compare the result with the methods `area` and `centroid` of Shapely. Table 4.1 shows the relative error for different domains.

	Shapes				
	Triangle	Square	Hectagon	L-shaped	$\mathcal{W}$
number of seeds	56	100	95	118	99
c_i	3×10^{-15}	3.3×10^{-15}	1.96×10^{-15}	5.23×10^{-15}	2.37×10^{-14}
m_i	5.36×10^{-15}	2.99×10^{-15}	3.92×10^{-15}	5.67×10^{-15}	2.62×10^{-14}

Table 4.1: Relative errors

4.4 Optimization

Functions F and K have been implemented along with the gradients ∇F and $\nabla\left(F_\phi\right)$. The gradient is a fundamental component of a gradient descent method and it is useful to test its correct implementation. For that we will compare the implementations with approximations given by finite difference methods, using both a first order right-sided scheme

$$\frac{\partial f}{\partial x_i}(\mathrm{x}) \approx \frac{f(\mathrm{x}+h\cdot e_i)-f(\mathrm{x})}{h}, i=1,\ldots,n\,, \tag{4.5}$$

and a second-order centered scheme

$$\frac{\partial f}{\partial x_i}(\mathrm{x}) \approx \frac{f(\mathrm{x}+h\cdot e_i)-f(\mathrm{x}-h\cdot e_i)}{2h}, i=1,\ldots,n\,, \tag{4.6}$$

for f a continuous scalar-valued function of several variables on $\mathbb{R}^N$ and h a small positive increment with e_i denoting the ith standard unit vector of $\mathbb{R}^N$. The CVT energy function F and its gradient will be evaluated at a set X composed of randomly generated points inside some polygon; the polygon will play the role of the domain and the points will be the seeds which will generate its Voronoi tessellation. The cost function F_ϕ will be valued at randomly generated sets of incident and adjoint angles spanning $[-\phi_{max}, \phi_{\max}]$. For the comparison we will compute the relative error of the finite difference methods at some point x

$$\epsilon = \frac{\|\nabla f(\mathrm{x}) - \widetilde{\nabla f}\|}{\|\nabla f(\mathrm{x})\|}\,,$$

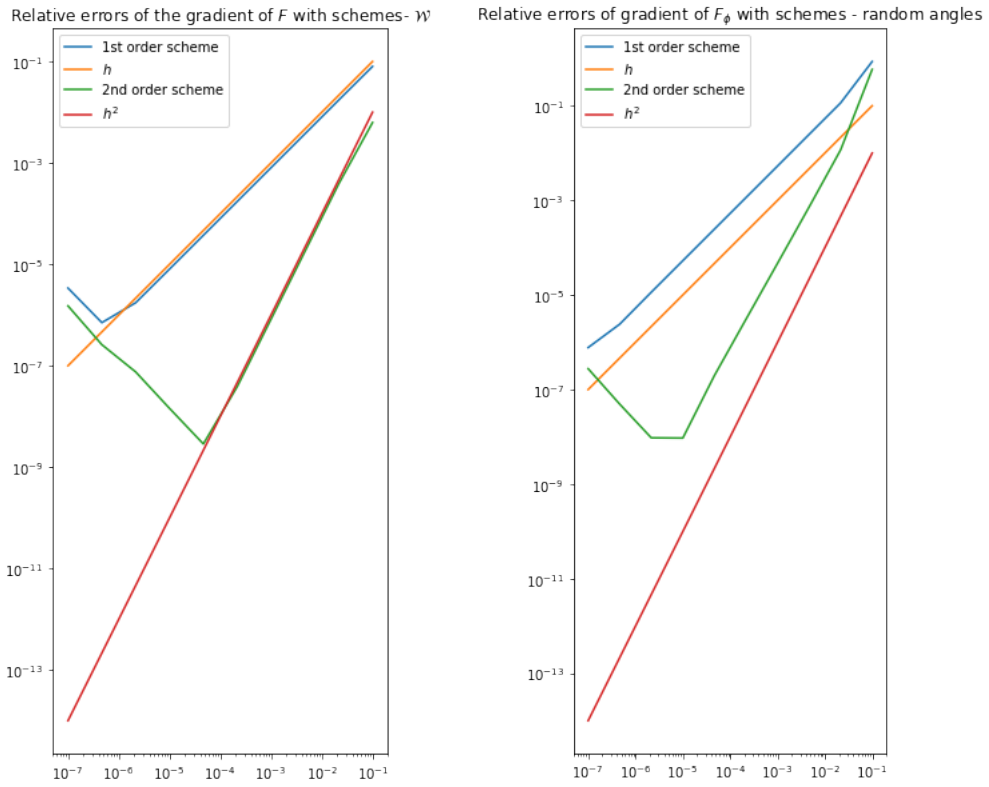
where $\widetilde{\nabla f}$ is the approximated gradient obtained through (4.5) or (4.6).

Figure 4.2(a) shows the relative errors obtained using different values of the increment h with F evaluated at a tessellation of $\mathcal{W}$ and 4.2(b) shows the relative error for F_ϕ at random angles. We confirm that the right-sided scheme is of first order and the centered scheme is of second order which indicates that the implemented gradients are correctly derived from their respective functions.

Being confident about the cost functions and their gradients we can use an optimization method to solve the corresponding problems. For that we will use `scipy.optimize.minimize`, a function for minimizing scalar objective functions of one or mor evariables, proposing several solvers among them `BFGS` for unconstrained problems and `L-BFGS-B` for bounded problems; the former will be used for the CVT as there is no clear way to identify the bounds characterizing the seeds and both solvers for the wavenumber coverage problem.

We test the `BFGS` solver of the `scipy` library on instances of the CVT problem against Lloyd's method. We take some different geometric shapes of domains and generate random seeds inside it as an initialization and apply both Lloyd's and BFGS methods to find a CVT. We then compare the results (values of F and the 2-norm of its gradient) and the performance (total execution time and number of iterations) of each.

Results of some of the tests have been compiled in table 4.2. The stopping criteria for Lloyd's method have been chosen to be the residuals between the seeds and the centroids of their respective Voronoi cells not exceeding a threshold, namely $\|X - T(X)\| < 10^{-6}$, or the number of iterations going beyond the limit 1000; thus the method either yields a CVT or stops short of it because it takes too much time to do so. So far in the examples Lloyd's method always terminate with a CVT. We notice that the two methods provide the same values of F while the difference resides in the magnitude of ∇F, with BFGS always yielding a gradient of order 10^{-5} because of its stopping criterion which sets a threshold for the gradient. This indicates that the two methods are both successful and the optimization approch is correct and well implemented.



(a) Test ∇F on $\mathcal{W}$.

(b) Test ∇F_ϕ on a random layout.

Figure 4.2: Relative errors of the gradients with first and second order finite difference schemes. The function F has been evaluated on $\mathcal{W}$ at 24 random points used as seeds and F_ϕ has been evaluated at an acquisition layout composed of 7 sources and 7 receivers of random angles ϕ_s and ϕ_r.

	Shapes				
	Triangle	Square	Hectagon	L-shaped	$\mathcal{W}$
number of seeds	49	100	94	138	105
Initialization					
Value of F	2.15×10^{-3}	1.03	3.32	0.06	12.22
Norm of ∇F	0.22×10^{-3}	0.59	1.60	0.06	5.06
Lloyd					
Value of F	8.49×10^{-4}	0.42	1.13	0.03	3.53
Norm of ∇F	5.27×10^{-6}	1.35×10^{-3}	1.98×10^{-4}	8.5×10^{-5}	3.77×10^{-3}
Number of iterations	86	163	555	308	312
Execution time	1.24 s	5.21 s	14.43 s	11.56 s	12.06 s
BFGS					
Value of F	8.43×10^{-4}	0.42	1.13	0.03	3.53
Norm of ∇F	3.33×10^{-5}	3.97×10^{-5}	5.35×10^{-5}	5.49×10^{-5}	4.46×10^{-5}
Number of iterations	146	153	164	302	133
Number of F evaluations	148	154	165	304	139
Execution time	99 s	163 s	170 s	466 s	233 s

Table 4.2: Comparison of the results between the Lloyd's and BFGS methods.

Chapter 5

Results

After the discussion about the implementation and the various tests conducted we will present in this chapter the results obtained when applying the method discussed in section 3.3. The instance of the problem consists of 10 sources and 10 receivers to be positionned on an acquisition range of $[0, 10000]$, with the diffraction point situated at x $= (5000, -3000)$ in the subsurface. As discussed in remark 1 we will take $f_0 = \frac{2\pi}{c_0}$ and the result of the optimization will still be relevant for other values of the frequency.

In the following we will present a typical solution of the minimization problem, a comparison of this solution with the ideal stable CVT of $\mathcal{W}$ and then a short study on the impact of the initialization on the quality of the solution.

5.1 A first optimization of the layout

Figure 5.1 presents a layout and the obtained wavenumber cloud corresponding to optimized angles ϕ_s and ϕ_r returned by BFGS, which was initialized with the shifted regular acquisition layout presented in figure 3.5. The value of the energy function reached the optimal value of 3.9×10^{-3} from an initial value of 5.4×10^{-3}, after 55 iterations and 61 function evaluations.

The resulting wavenumber cloud looks relatively satisfying in terms of regularity compared to clouds seen so far. Notice the particular configuration of the acquisition layout on the surface: here (1) each source faces a receiver, and vice-versa, with regards to the centeral vertical axis of the acquisition range, (2) there are noticeable blank spaces at both extremities, separarting the two extremal acquisition devices from the rest of the layout and (3) the positioning in the middle part of the range is in pairs of devices of similar nature, two sources or two receivers, whereas it alternates between a source and a receiver in the rest of the range. Such a configuration is counter-intuitive as in general a regular positioning of the sources and receivers would come to mind when tasked with assembling an acquisition layout.

5.2 CVT of $\mathcal{W}$

To get a lower bound of the energy function F_ϕ we consider a stable CVT of $\mathcal{W}$ that we obtain by minimizing F with $n = N_s \times N_r = 100$ seed points. Figure 5.2 shows a stable CVT alongside the Voronoi tessellation we get from the points of the wavenumber cloud of figure 5.1. The stable CVT represents an ideal regular coverage of the wavenumber cloud that we would get had not for the interdependency constraints of the seeds.

We see that the Voronoi tessellation of $\mathcal{W}$ is close to a CVT, the difference results from the fact tha the interdependency constraints forbid a seed to correspond to the centroid of its Voronoi cell because otherwise it would increase some terms $\|x_i - c_i\|^2$ in (3.11) by moving other seeds; the result 5.2(b) corresponds to a balance. Values of the function F for the CVT

	Stable CVT	Optimal layout
Value of F	2.76×10^{-3}	3.9×10^{-3}

Table 5.1: Comparison of the values of the energy function F at the stable CVT of $\mathcal{W}$ and for the optimized layout.

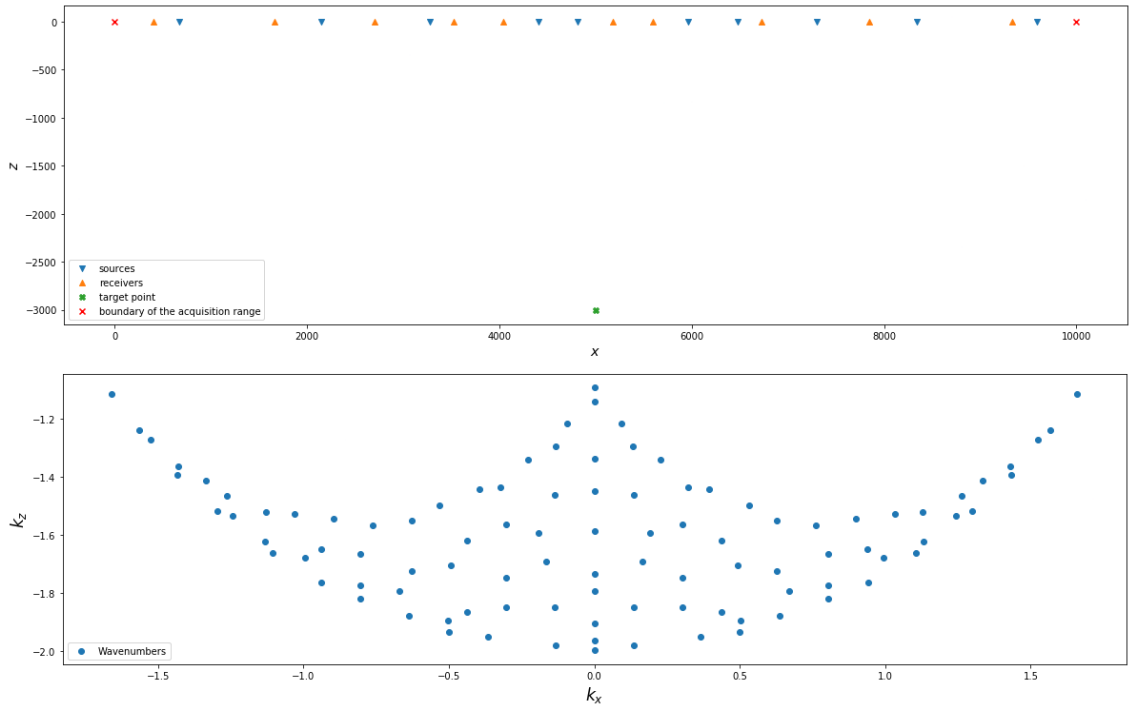


Figure 5.1: The optimal positioning of 10 sources and 10 receivers inside a range $[0, 10000]$ on the surface, resulting from the optimal angles returned by BFGS using the configuration of figure 3.5 as initialization.

and the optimized layout are presented in table 5.1. We can see that the minimizer of F_ϕ is still far from the ideal minimizer of F.

5.3 Different initializations

To investigate the impact of the choice of initialization we have considered four initial configurations:

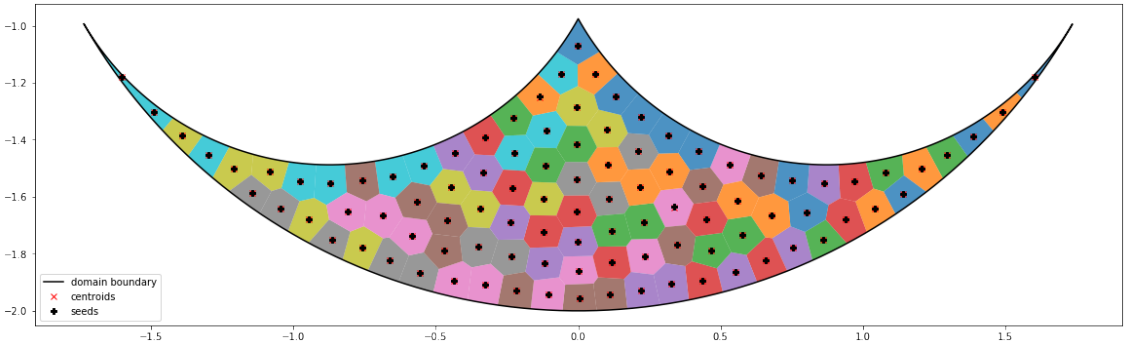
1. A staggered[1] regular positioning of the sources and receivers, the same as in section 5.1.
2. Random positions of the N_s sources and N_r receivers on the surface generated via a continuous random uniform distribution on $[0, 10000]$.
3. An acquisition layout corresponding to angles ϕ_s and ϕ_r forming a regular discretization of $[-\phi_{max}, \phi_{max}]$ as in figure 3.6, but shifted as to avoid duplicate wavenumebrs.
4. An acquisition layout corresponding to $N_s + N_r$ uniformely generated angles ϕ_s and ϕ_r on $[-\phi_{max}, \phi_{max}]$.

We will compare the four initial configurations on the basis of the optimal value of F_ϕ returned by the optimization program. We will also compare their efficiencies by looking at the initial values of F_ϕ and the number of iterations.

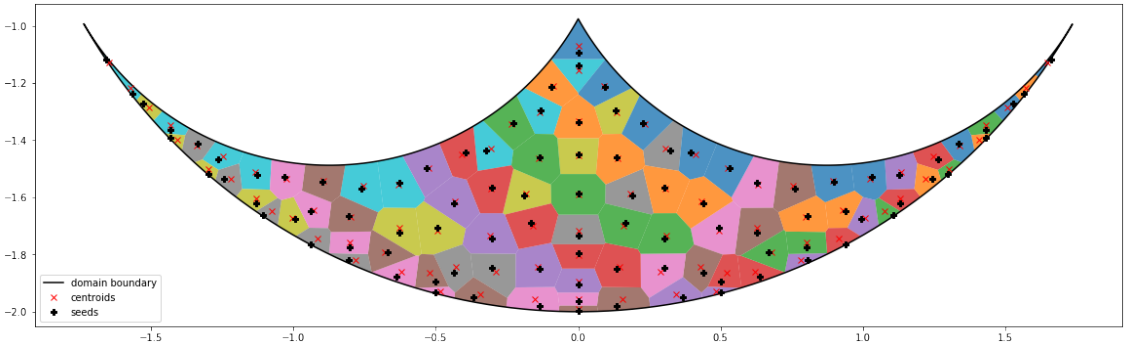
Results for deterministic configurations 1 and 3 are compiled in table 5.2. The optimized layout obtained by using regular initial incident and adjoint angles is similar to the result in figure 5.1, as the optimal values of F_ϕ in the table suggests, the difference being that it takes more iterations (64) to achieve this value for regular angles than initializing with regular positions (which takes 55 iterations) despite the fact that we start with a more promising initial value in the former (4.5×10^{-3}) than the latter (5.4×10^{-3}).

For configurations 3 and 4 subject to randomness we considered 10 runs of the program and the results are compiled in table 5.3. Overall the runs with random initializations yielded better

[1] No source and receiver share the same location.



(a) A stable CVT of $\mathcal{W}$ obtained via BFGS.



(b) The Voronoi tessellation of $\mathcal{W}$ from the previous optimal wavenumbers.

Figure 5.2: Comparison of a stable CVT and the Voronoi tessellation generated by the points of the optimized wavenumber cloud.

	- 1 - Regular positions	- 3 - Regular angles
Initial value of F_ϕ	5.4×10^{-3}	4.5×10^{-3}
Optimal value of F_ϕ	3.9×10^{-3}	3.9×10^{-3}
Number of iterations	55	64
Number of evaluations of F_ϕ	61	81

Table 5.2: Comparison of the results of different determinitic initializations for the optimization program.

results than those of table 5.2, but at the cost of a higher number of iterations due to the fact that the optimization program starts at worst initial values (on average 1.43×10^{-3} for the random positions and 1.35×10^{-3} for the random angles).

The best result found so far came from a random initialization of the angles and achieved a final value of 3.31×10^{-3}, from an initial value of 8.35×10^{-3}, after 75 iterations and 81 evaluations of F_ϕ. Figure 5.3 shows the optimized acquisition layout and the wavenumber cloud obtained from this configuration. Compared to 5.1, we gained a more regular spacing of the points in the bottom region of the wavenumber cloud which translates into a smaller value of F_ϕ. However the symmetry is lost and at first glance no particular pattern seems to emerge from the optimized layout, if we exclude the positionning in a row of 5 sources (resp. receivers) on the right side (resp. the left side) of the acquisition range.

We conclude from this comparison that the proposed optimization method is sensitive to the initialization and an initial configuration subject to randomness may yield better results than a regular one of smaller initial energy. Also the best result we achieved out of all the execution of the program has an assymetric wavenumber cloud and a counterintuitive acquisition layout on the surface.

		- 2 - Random positions	- 4 - Random angles
Initial value of F_ϕ (Average)		1.43×10^{-2}	1.35×10^{-2}
Optimal value of F_ϕ	Average	3.9×10^{-3}	3.63×10^{-3}
	Minimum	3.33×10^{-3}	3.31×10^{-3}
	Maximum	3.93×10^{-3}	3.9×10^{-3}
	$< 3.9 \times 10^{-3}$	9	10
Number of iterations (Average)		87	98
Number of evaluations of F_ϕ (Average)		91	103

Table 5.3: Comparison of the results over 10 runs of different random initializations for the optimization program.

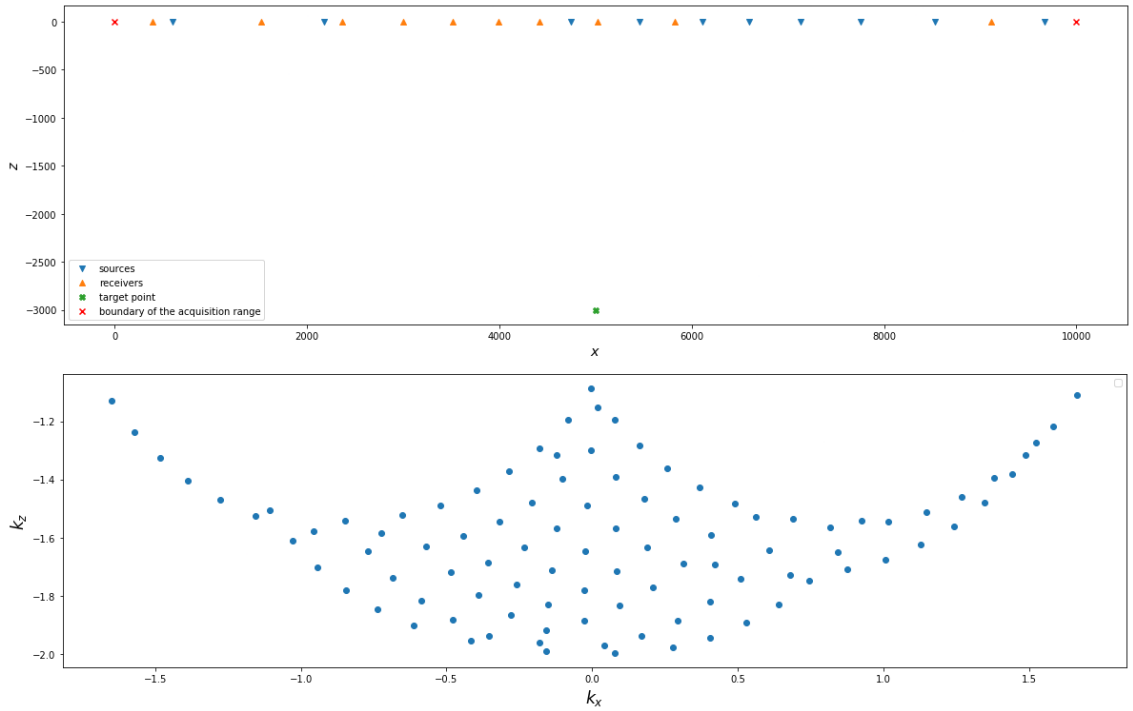


Figure 5.3: The best result returned by BFGS from all the previous runs, using an initial configuration obtained from random angles.

Chapter 6

Conclusion

In this project we proposed a new approach for the FWI to improve the wavenumber coverage of its gradient. To this end we expressed the optimal design of the acquisition layout as a minimization problem using the angular expression of the wavenumbers and we formalized it as an instance of stable CVT. Where Lloyd's method proved inadequate, results from Liu et al. made it possible to solve the problem using Newton-based methods.

The implementation in Python and the various tests conducted on a simple instance led us to a first glimpse on what constitutes an optimal positioning of the sources and receivers on the surface. So far the best acquisition layouts found by the method have been obtained using random initial configurations and seem chaotic, with no clear optimal pattern emerging.

For this reason a thorougher exploratrion of the minimizers of the energy function F_ϕ should be conducted in the aim of infering an optimal pattern that would yield optimal or at least good results for any instance of the homogeneous problem, no matter the number of the acquisition devices.

To demonstrate the usefulness of the approach the FWI algorithm should be tested using an optimized acquisition and compare the result to using a more conventional acquisition.

The model considered in this project being simplistic it is far from reflecting reality and as such the approach should be extended to more complex models:

1. Take into account the topography of the terrain in the acquisition range, which implies finding a formula to get the positions of the sources and receivers from the angles ϕ_s and ϕ_r when the surface is not flat.

2. Explore models with non-homogeneous mediums: for instance 1D models where the velocity depends on the depth z of the subsurface or a more general case where the incident and adjoint vectors p_s and p_r are obtained through the resolution of an eikonal equation. In the former case a formula linking the angles and the positions should be found and in the latter we need to incorporate or implement a numerical method to solve the eikonal equation.

3. Consider a noncentered target point in the acquisition range or even several target points at the same time, in this case we need a redifinition of the energy function F_ϕ.

4. Extend the approach to 3D models. Here the question of the computation cost should be addressed and the performence of the implementation will be central because of the leap in complexity from the 2D case to the 3D one. It might prove useful to move from Python to more low-level programming languages like **C++** or **Fortran**.

Appendix A

Optimization

Two fundamental strategies exist for solving unconstrained optimization problems of smooth function: line search and trust region [10]. A linesearch algorithm is an iterative method where the next iterate m_{k+1} is chosen from the previous one following a descent direction Δm and steplength α_k:

$$m_{k+1} = m_k + \alpha_k \Delta m. \tag{A.1}$$

At each iteration new descent direction and steplenght need to be chosen. Provided a sufficiently small steplength, any direction Δm_k that makes an angle inferior to $\frac{\pi}{2}$ with the opposite of the gradient

$$-\frac{\nabla f(m_k)\Delta m_k}{\|\nabla f(m_k)\|\|\Delta m_k\|} < 0 \tag{A.2}$$

is guaranteed to produce a decrease in the cost function. Different choices of the descent direction exist, the most used ones are the steepest descent which takes the opposite of the gradient $\Delta m_k = -\nabla f(m_k)$ at each iteration and the Newton method which fixes $\Delta m_k = -H(m_k)^{-1}\nabla f(m_k)$ where $H(m_k)^{-1}$ designates the inverse of the Hessian of f, provided f is smooth enough.

Finding the best steplength α_k amounts to solving the optimization problem

$$\min_{\alpha>0} f(m_k + \alpha_k \Delta m_k) \tag{A.3}$$

at each iteration, which besides being unpractical is sometimes unnecessary [10]. Instead of taking the best steplenght we settle for a "good" steplength; two sets of criteria exist to assess the suitability of the steplength: the Wolfe conditions and the Goldstein conditions.

Appendix B

Adjoint state method

Let us rewrite the misfit function in (2.3) as a sum

$$f(m) = \sum_{s=1}^{N_s} f_s(m) = \frac{1}{2} \sum_{s=1}^{N_s} \sum_{r=1}^{N_r} \int_0^T |d_{cal,s}(x_r, t) - d_{obs,s}(x_r, t)|^2 dt \,,$$

so we can express the gradient $\nabla f(m)$ as a sum of the gradients of $f_s(m)$

$$\nabla f(m) = \sum_{s=1}^{N_s} \nabla f_s(m). \tag{B.1}$$

Let us find the expression of ∇f_s for a fixed index s. Consider u_s and $d_{cal,s}$ as independant variables, taking the Lagrangian of f_s

$$L(m, u_s, d_{cal,s}, \lambda_s, \mu_s) = f_s(m) + \int_0^T \int_\Omega \left[A(m)u_s - \varphi_s\right](x,t) \cdot \lambda_s(x,t) dx dt \\ + \sum_{r=1}^{N_r} \int_0^T \left[d_{cal,s} - R[u_s]\right](x_r, t) \cdot \mu_s(x_r, t) dt \,,$$

we find from the expression of its gradient conditions on λ_s and μ_s [9, p. 81] for which ∇f_s has the form

$$\nabla f_s(m)^1 = \int_0^T \frac{\partial A(m)}{\partial m} u(x,t) \cdot \lambda_s(x,t) dt. \tag{B.2}$$

Namely, equation B.2 holds for Lagrangian multipliers λ_s, μ_s that satisfy:

$$\begin{cases} A^T(m)\lambda_s = R^T \mu_s \\ \mu_s = d_{obs} - d_{cal} \end{cases}.$$

Let's remark that μ_s is the opposite of the residual $d_{cal,s} - d_{obs,s}$ and λ_s is the solution of the adjoint[2] wave equation:

$$A^T(m)\lambda_s = R^T(d_{obs,s} - d_{cal,s}) \,. \tag{B.3}$$

From the result (B.2) we deduce that the gradient ∇f_s can be seen as the zero-lag time correlation between the incident and adjoint wavefields u_s and λ_s.

[1] m is depends on x so f_s can also be expressed as a function of x

[2] The name adjoint state method derives from the fact that the conjugate transpose of a matrix is also called the *adjoint matrix*; equation B.2 is called the *adjoint equation* and λ_s is called the *adjoint state vector* [1, p. 2]

Bibliography

[1] Andrew M. Bradley. Pde-constrained optimization and the adjoint method. `https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf`, October 2019. [Online; accessed 22 May 2021].

[2] Guy Chavent. Identification of functional parameters in partial differential equations. 06 1974.

[3] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999. doi: 10.1137/S0036144599352836. URL `https://doi.org/10.1137/S0036144599352836`.

[4] Sean Gillies et al. Shapely: manipulation and analysis of geometric objects. `"https://github.com/Toblerity/Shapely"`, 2007–2021. [Online; accessed 01 June 2021].

[5] Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[6] Y. Huang and G. Schuster. Resolution limits for wave equation imaging. *Journal of Applied Geophysics*, 107:137–148, 2014.

[7] Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. On centroidal voronoi tessellation—energy smoothness and fast computation. *ACM Trans. Graph.*, 28(4), September 2009. ISSN 0730-0301. doi: 10.1145/1559755.1559758. URL `https://doi.org/10.1145/1559755.1559758`.

[8] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982. doi: 10.1109/TIT.1982.1056489.

[9] Ludovic Métivier. Lecture notes: High resolution geophysical imaging using full waveform modeling and inversion. `https://www-ljk.imag.fr/membres/Ludovic.Metivier/LECTURE_NOTES/2020_COURS_MSIAM_FWI.pdf`, January 2021. [Online; accessed 22 May 2021].

[10] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[11] Laurent Sirgue and Robert Pratt. Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies. *Geophysics*, 69:231–248, 01 2004. doi: 10.1190/1.1649391.

[12] Pauli Virtanen. `https://gist.github.com/pv/8036995`. [Online; accessed 01 June 2021].

[13] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.