École polytechnique de Louvain

Wireless indoor source localization as an inverse problem

An optimization approach via ray tracing simulations

Author: **Sophie LEQUEU**
Supervisors: **Prof. Laurent JACQUES, Prof. Claude OESTGES, Jérome EERTMANS**
Reader: **Prof. Benoît LEGAT**
Academic year 2024–2025
Master [120] in Mathematical Engineering

Abstract

Ray tracing has emerged as an increasingly accurate method for modeling the propagation of electromagnetic waves in complex environments. Given a virtual representation of a scene and a specified transmitter position, it enables the estimation of the received power at the receiver locations. This motivates the use of ray tracing to address a corresponding inverse problem, namely, wireless indoor source localization. Wireless indoor source localization refers to the problem of determining the position of a radio transmitter in an indoor environment, based on measurements collected at multiple receivers and knowledge of the environment. In this work, we formulate this task as an optimization problem. We aim to determine the transmitter position that induces, through ray tracing simulations, measurements closest to those observed at the receivers. We first analyze the objective function resulting from the ray tracing model. We then propose resolution strategies based on first-order methods or derivative-free optimization, and evaluate the feasibility and the limitations of this approach.

La méthode de tracé de rayons (ray tracing) s'impose comme un outil de plus en plus précis pour modéliser la propagation des ondes électromagnétiques dans des environnements complexes. A partir d'une représentation virtuelle de la scène et de la position de l'émetteur, elle permet d'estimer la puissance reçue aux emplacements des récepteurs. Cela peut motiver son utilisation pour la résolution d'un problème inverse associé, à savoir la localisation sans fil en intérieur. Ce problème consiste à déterminer la position d'un émetteur électromagnétique dans un environnement encombré, à partir de mesures observées à plusieurs récepteurs et d'une connaissance de la scène. Dans ce travail, nous formulons cette tâche comme un problème d'optimisation. L'objectif est de déterminer la position de l'émetteur qui induit, via la simulation par tracé de rayons, des mesures les plus proches possible de celles observées. Nous commençons par analyser la fonction objectif découlant du modèle par tracé de rayons. Nous proposons ensuite des stratégies de résolution basées sur des méthodes du premier ordre ou des techniques d'optimisation sans dérivées, et évaluons la faisabilité et les limites de cette approche.

Acknowledgements

First, I would like to thank my supervisors Prof. Laurent Jacques, Prof. Claude Oestges and Jérome Eertmans for their advice, their time and their availability throughout the year. I would also like to thank Prof. Benoît Legat for accepting to read this Master's thesis and for dedicating time to evaluate my work.

Then, I would like to sincerely thank all the inspiring people I have had the privilege of meeting throughout my studies. In particular, I am deeply grateful to my close friends for their enthusiasm and unwavering support.

Finally, I thank my family, with special thanks to my grandparents, for their encouragements, their confidence, and so much more.

Use of AI

I used the AI programming tool *GitHub Copilot* for documenting and generating the simplest functions in the code. I also acknowledge the use of *DeepL* for translating words or sentences in this work, and the generative AI chatbot *ChatGPT* for pointing out possible mistakes and rephrasing. I reread the AI contributions carefully afterwards.

Contents

List of Notations **v**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Inverse problems . . . 3
1.3 Structure of the work . . . 3
1.4 Contributions . . . 5

2 Forward model for wave propagation and power measurements **6**
2.1 Wave propagation in free space . . . 6
2.2 Simplified laws for interactions . . . 11
2.3 Path loss and received power . . . 12
2.4 Forward problem formulation . . . 14
2.5 Ray tracing methods . . . 17
2.6 Differentiable ray tracer in telecommunications . . . 19

3 Related work **21**
3.1 Types of information . . . 21
3.2 General approaches . . . 22
3.3 Computer graphics and inverse rendering . . . 24

4 Indoor source localization as an inverse problem **26**
4.1 Inverse problem formulation . . . 26
4.2 Challenges . . . 27
4.3 Proposed solutions . . . 30

5 Resolution strategies for source localization **31**
5.1 Smoothing discontinuities . . . 31
5.1.1 Gaussian smoothing of the loss function . . . 31
5.1.2 Transmitter's position tolerance smoothing . . . 34
5.1.3 Physics-based smoothing . . . 37
5.2 Optimization methods . . . 40
5.2.1 Derivative-free optimization methods . . . 40
5.2.2 First order optimization methods . . . 42
5.2.3 Additional remarks . . . 42
5.3 Initialization . . . 42

6 Results and discussion **44**
6.1 Simulation tool and environments . . . 44
6.2 Parameter tuning and test in free space . . . 46
6.3 Optimization of the piecewise continuous loss function . . . 47
6.3.1 Number of receivers . . . 47
6.3.2 Number of iterations and methods comparison . . . 49

6.3.3 Number of Monte Carlo samples . . . 50
6.3.4 Smoothing factor . . . 51
6.3.5 Stochastic approach . . . 52
6.4 Resolution of the inverse problem . . . 53
6.4.1 Number of receivers . . . 53
6.4.2 Complexity of the scene . . . 53

7 Conclusion and perspectives **55**
7.1 Related inverse problem . . . 55
7.2 Summary . . . 56
7.3 Conclusion . . . 56

Bibliography **58**

Appendix **61**
A Implementation details . . . 61
B Algorithms . . . 61
B.1 Pseudocode for LGS approach (section 5.1.1) . . . 61
B.2 Pseudocode for TTS approach (section 5.1.2) . . . 62

List of Notations

Abbreviations

AD	Automatic differentiation
AoA	Angle of arrival
CG	Computer graphics
CLT	Central limit theorem
`DiffeRT`	Python module for differentiable ray tracing in 3D, developed by supervisors Jérome Eertmans, Prof. Claude Oestges and Prof. Laurent Jacques.
DFO	Derivative-free optimization methods
GM	Gradient method
GNSS	Global navigation satellite system
GO	Geometrical optics
GPS	Global positioning system
GS	Gaussian smoothing
LOS	Line-of-sight
LGS	Gaussian smoothing of the loss function
MC	Monte Carlo
MSE	Mean squared error
NLOS	Non-line-of-sight
PBS	Physics-based smoothing
RFD	Random finite differences
RMS	Root-mean-square
RSS	Received signal strength
RT	Ray tracing
Rx	Receiver
TDoA	Time difference of arrival
ToA	Time of arrival
TTS	Transmitter's position tolerance smoothing
Tx	Transmitter

UTD	Uniform theory of diffraction

General notations

$\|\cdot\|$	Euclidean norm.
$*$	Convolution.
$\cdot$	Dot product between 2 vectors, or matrix-vector product.
∇	Gradient.
$\mathbf{J}f$	Jacobian matrix of function f.
$\hat{\boldsymbol{i}}, \hat{\boldsymbol{r}}, \hat{\boldsymbol{n}}, \hat{\boldsymbol{e}}$	Unit vectors associated to vectors $\boldsymbol{i}, \boldsymbol{r}, \boldsymbol{n}, \boldsymbol{e}$.
$\boldsymbol{E}, \boldsymbol{H}$	Phasor quantities.
$\hat{\mathcal{L}}$	Estimator of $\mathcal{L}$.
I_n	Identity matrix of size $n \times n$.
$X_i \overset{\textbf{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 I_n)$	The variables or vectors $X_i \in \mathbb{R}^n$ for $i = 1, \ldots, N$, are i.i.d. with a Gaussian distribution of mean 0 and covariance matrix $\sigma^2 I_n$.
$\mathcal{G}_\sigma(\cdot)$	2D Gaussian kernel of mean 0 and covariance matrix $\sigma^2 I_2$.
$[A]_{i,:}$	i-th row of matrix A.

Problem-specific notations

$t \in \mathbb{R}^2$	Position (or candidate position) of the transmitter.
$\bar{t} \in \mathbb{R}^2$	True position of the transmitter to recover.
m	Number of receivers.
$r_i \in \mathbb{R}^2$	Position of the i-th receiver.
$\bar{r} \in \mathbb{R}^{m \times 2}$	Positions of the m receivers.
$\mathcal{S}$	Knowledge of the scene: geometry (walls) and electromagnetic properties.
$\Phi(t, r_i) \in \mathbb{R}$	Power received at receiver r_i when the transmitter is at t (dependence on the scene $\mathcal{S}$ is omitted in the notation).
$\mathcal{A}(t, r, \mathcal{S}), \mathcal{A}(t) \in \mathbb{R}^m$	Forward operator giving the received powers at the m receivers located at r in a scene $\mathcal{S}$ with transmitter at t. r and $\mathcal{S}$ may be omitted when only t varies.
$\mathcal{A}^{\mathrm{GS}}(t)$	Smoothed forward operator computed from the convolution of each $\Phi(., r_i)$ with a 2D Gaussian kernel.
$\mathcal{L}(t)$	Loss function evaluated at candidate transmitter position t.
$\mathcal{L}^{\mathrm{GS}}(t)$	Smoothed loss function computed from the convolution of $\mathcal{L}$ with a 2D Gaussian kernel.
$\mathcal{L}^{\mathrm{TTS}}(t)$	Smoothed loss computed from the smoothed forward operator $\mathcal{A}^{GS}(t)$.

Chapter 1

Introduction

Wireless indoor source localization refers to the problem of determining the position of a radio transmitter in an indoor environment, based on measurements collected at multiple receivers and knowledge of the environment.

1.1 Motivation

In an increasingly connected world, location-based services have become ubiquitous, and the demand for accurate and efficient localization techniques has driven significant research in this field [1]. Wireless localization systems have been developed, relying on the propagation of electromagnetic signals from a transmitter to a receiver. While outdoor localization has been extensively studied for over 50 years — leading to the widespread adoption of Global Navigation Satellite Systems (GNSS), such as the GPS [2, 3] —, these techniques are ineffective indoors. Indeed, GNSS rely on the Line-Of-Sight (LOS) propagation between the satellites and the users, which is often obstructed in indoor environments or dense urban areas [4]. New challenges emerge when designing localization techniques for these so-called GNSS-denied environments, where multipath propagation, reflections and diffractions dominate [5].

Indoor localization finds numerous practical applications, ranging from autonomous navigation of unmanned aerial vehicles [6], to tracking soldiers or firefighters during operations [7], or locating people and assets in hospitals, supermarkets, tunnels and other confined areas [8]. Although the method presented in this thesis will only be developed and evaluated on simulations, indoor localization is an active and developing field, where exploring new theoretical approaches is valuable.

In most wireless indoor localization systems, the object to be located carries a transmitting antenna, while multiple receiving antennas are placed at known positions in the environment. This setup contrasts with outdoor localization systems such as GNSS, where satellites act as transmitters and the user's device serves as the receiver. Indoors, we benefit from more control over the environment, including the ability to place infrastructure strategically. It enables having a lightweight transmission device on the object to be detected, while the measurement, processing and computation devices are fixed in the environment.

Various techniques for wireless indoor localization are available in the literature [11]. Most approaches commonly assume a known 2D or 3D environment, including the geometry, the

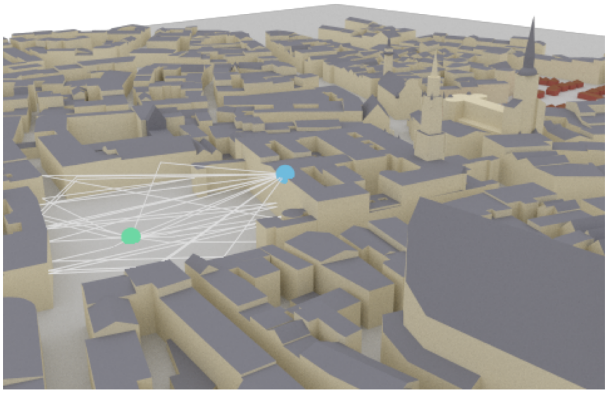


Figure 1.1: Ray tracing (3D) in an urban area with multiple paths and interactions between the antenna (blue) and receiver (green) from [9].

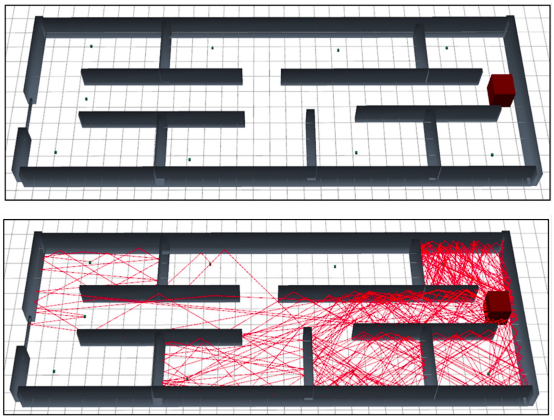


Figure 1.2: Ray tracing (2D) in an indoor environment with multiple paths and interactions between a base station (red box) and 8 receivers (black dots) from [10, Figure 3].

electromagnetic properties of materials, receivers' positions, and known transmitting power of the transmitter. They differ by the type of data collected at the receivers. Common types of measurements include Angle-of-Arrival (AoA), Time-of-Arrival (ToA), or Received Signal Strength (RSS) [11]. In this thesis, we focus on the case where only RSS measurements are available, meaning that the receivers record the power received at their locations. Given these power measurements, our goal is to recover the position of the transmitter.

Traditionally, research has focused on methods inspired by outdoor localization, relying on LOS paths between transmitters and receivers. These techniques treat Non-Line-Of-Sight (NLOS) propagation and multipaths effects as noise, attempting to detect or mitigate them [12, 13]. However, in complex and cluttered environments, such approaches become increasingly difficult and imprecise. On the other hand, indoor environments are typically *information-rich*, and this additional information could be leveraged rather than treated as a limitation.

When the geometry and electromagnetic properties of the environment are known, a *digital twin scene* can be created, a virtual representation of the scene. Wave propagation can be accurately described by *rays* and numerically evaluated in simulation using Ray Tracing (RT). RT is the process of computing the *feasible paths* from the transmitter to the receivers. Examples are shown on Figure 1.1 and Figure 1.2. RT also allows estimation of the received power at the receivers, for any given transmitter position.

In this Master's thesis, we study a novel approach to RSS-based wireless indoor localization, using RT simulations. We formulate the task of finding the position of the transmitter as an optimization problem, that aims to minimize the distance between the acquired measurements and those simulated for a candidate transmitter position in the virtual representation of the scene. If we can iteratively generate these candidate measurements using RT simulations, and differentiate through the simulation [14, 15] with respect to the transmitter's position, this optimization problem could be solved using first order optimization methods. This is a classical approach to solving *inverse problems*.

The main purpose of this work is therefore to propose an indoor source localization method, based on RT simulations and a virtual representation of the scene.

Finally, we mention that similar ideas are considered in the related field of Computer Graphics (CG). Even though the applications and RT methods are different, the subject closely relates to differentiable rendering [16] and inverse rendering [17].

1.2 Inverse problems

Inverse problems generally start with their associated *forward problem.* Forward problems are well-posed problems of the form:

$$\mathcal{A}(t) = b,$$

where $\mathcal{A}$ is called the *forward* operator, $t \in \mathcal{T}$ is the parameter and $b \in \mathcal{B}$ are the measurements. Forward problems consist in computing the measurements b associated to a parameter t, based on the operator $\mathcal{A}$ encoding the underlying physics model of the problem [18].

In real life, we also face the problem in its *inverse* form. Observing data $\tilde{b}$, we would like to recover (or at least estimate) the parameter t that produced these data through the operator $\mathcal{A}$. This is the *inverse problem.* We note that, in practice, measurements $\tilde{b}$ may be obtained through experiments, meaning that they contain additional noise, from modeling errors and measurement errors. We write:

$$\mathcal{A}_0(t) + \eta = \tilde{b} \approx \mathcal{A}(t) = b,$$

where $\mathcal{A}_0$ denotes the real (but unknown) operator — possibly differing from the modeled forward operator $\mathcal{A}$, and accounting for modeling errors — and η is the measurement noise.

The indoor source localization problem, of interest in this work, can be seen as an inverse problem associated with the problem of computing the received power at some receivers, given the position of the transmitter and knowledge of the environment. Due to the complexity and nonlinearity of the forward model, the parameter t — here, the position of the transmitter — cannot be recovered by simply inverting the operator $\mathcal{A}$. As mentioned in the previous section, it is common to solve such an inverse problem through repeated simulations of the forward problem, iteratively improving the candidate parameter. A general illustration of this procedure is shown on Figure 1.3.

1.3 Structure of the work

This Master's thesis is structured into three main parts and seven chapters.

- The first three chapters present the context and theoretical background.
  In this first chapter (**Chapter 1**), we introduce the topic of source localization in NLOS environments, providing motivation for its study and outlining our contributions. In **Chapter 2**, we focus on modeling the forward problem. Since solving the propagation equations exactly is intractable due to the complex geometry, we justify the use of a simplified wave propagation model to approximate the power transmitted from the

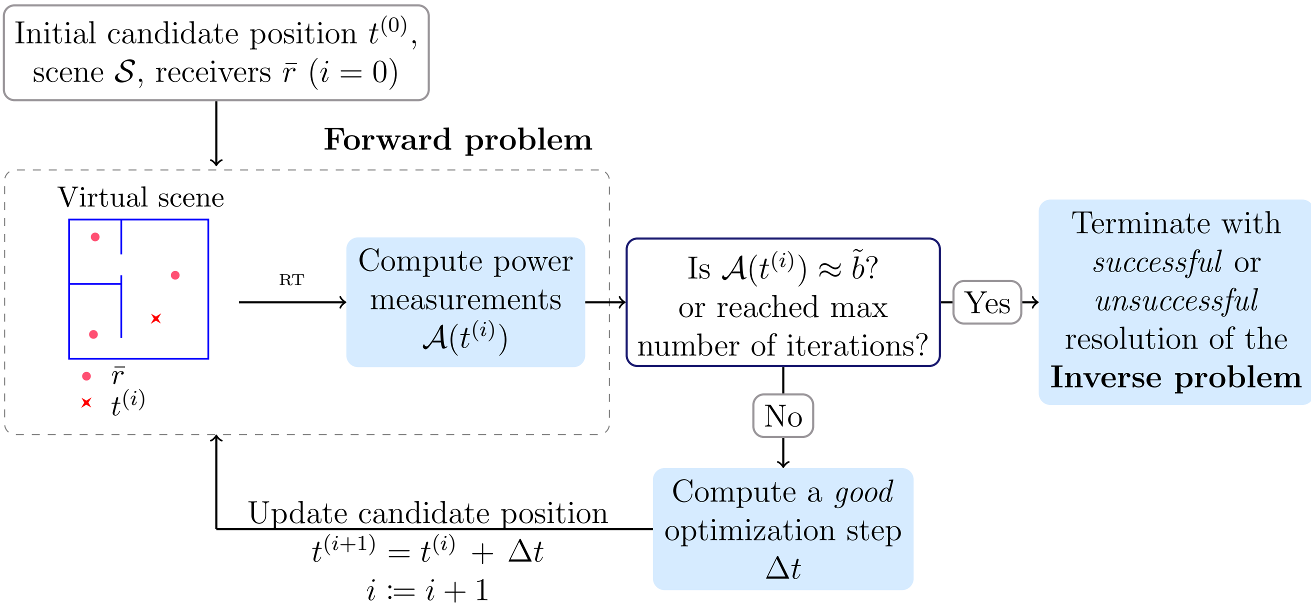


Figure 1.3: Illustration of an iterative inverse problem resolution procedure.

transmitter to the receivers. We detail the concept of *rays* and the simplified laws of reflection and diffraction used in RT. We then discuss various approaches for solving the RT problem, with a focus on exhaustive methods and their (almost everywhere) differentiable implementation. **Chapter 3** reviews the state of the art in indoor localization, discussing the different types of information and methods proposed in the literature.

- The second part presents our approach.
  Chapter 4 introduces our formulation of the inverse problem as the minimization of a loss function. We show that, unfortunately, a naive application of Gradient Method (GM) is inadequate and will often fail to converge to the global minimizer. We highlight the key challenges: the loss function is *i)* only piecewise continuous, and *ii)* nonconvex (even on the subdomains over which it is continuous). **Chapter 5** presents the techniques proposed to address these challenges. It includes *i)* possible smoothing strategies and *ii)* initialization heuristics. We present the optimization methods and the overall proposed approach. Finally, **Chapter 6** assesses the performance of our approach on example scenarios of simplified 2D maps of indoor environments. We evaluate the *success rate* for varying numbers of receivers. We also compare the different methods proposed in Chapter 5 and the impact of their parameters.

- The final chapter, **Chapter 7**, investigates related problems, improvements and perspectives.
  We briefly present the related problem of estimating the reflection coefficient of walls within the scene. Indeed, this is another inverse problem associated with the same forward problem. Finally, we list possible improvements and future research directions, and offer a critical opinion on the proposed method and its limitations.

1.4 Contributions

1. We formulate the wireless indoor source localization task as an optimization problem, minimizing a loss function. To the best of our knowledge, this approach has only recently been explored in the literature in the recent paper [19] (January 2025). We analyze the characteristics of the loss function, arising from the model used in the formulation of the direct problem. Discontinuities, non-convexity and modeling errors in the direct problem are all challenges for the proposed method.

2. We address the above challenges by proposing different smoothing strategies to eliminate discontinuities. The first two relate to an approximation of a Gaussian filtering of the discontinuous loss function. The third one eliminates the sources of discontinuity directly, modifying the model.
   We connect the first two smoothing approaches to derivative-free optimization methods, and we apply classical gradient-based methods (GM) for the third one, leveraging Automatic Differentiation (AD). We also emphasize the importance of initialization in the present nonconvex settings, and propose initialization heuristics.

3. Leveraging the `DiffeRT` Python module, we study the reliability of the proposed wireless indoor source localization method, on simulations. We emphasize the impact of modeling errors, and highlight the limitations of solving the localization task as an inverse problem.

In this work, we restrict ourselves to a 2D description of the environment, but the proposed methods can be generalized to a 3D scene.

Chapter 2

Forward model for wave propagation and power measurements

In *wireless* indoor localization, the term *wireless* refers to electromagnetic waves propagating from a transmitter to one or more receivers without physical connections. Various types of signals can be used for this purpose, such as Wi-Fi, RFID, Bluetooth, or mmWaves [20]. They all satisfy Maxwell's equations of electromagnetic wave propagation. However, solving these equations in complex indoor environments is generally infeasible [21]. A practical alternative is Ray Tracing (RT), which approximates wave propagation using *rays*.

In this chapter, we detail how Maxwell's equations can be approximated by *rays* following straight lines in free space and respecting the laws of reflection and diffraction when interacting with obstacles. We also describe how the power transmitted along these *rays* can be modeled. The first two sections are based on [22] by D. A. McNamara, and aim to be a summary of its second and third chapters. We take this opportunity to underline our assumptions throughout the development, and their consequences. We then formalize the forward problem for received power computation. Finally, we review RT methods in telecommunications and outline their (nearly-everywhere) differentiability.

2.1 Wave propagation in free space

From Maxwell's equations to the Helmholtz equation

Let us denote the spatial coordinate with $r \coloneqq (x, y, z)$ and the angular (radian) frequency with ω. The well-known system of Maxwell's equations describing the behavior of electromagnetic waves — at any point r which is not a source — can be written as follows.

$$\begin{aligned}
\nabla \times \boldsymbol{E}(r,\omega) + j\omega\mu(r)\,\boldsymbol{H}(r,\omega) &= 0,\\
\nabla \times \boldsymbol{H}(r,\omega) - j\omega\varepsilon(r)\,\boldsymbol{E}(r,\omega) &= 0,\\
\nabla \cdot [\varepsilon(r)\boldsymbol{E}(r,\omega)] &= 0,\\
\nabla \cdot [\mu(r)\boldsymbol{H}(r,\omega)] &= 0.
\end{aligned}$$

In the above, the electric and magnetic fields $\boldsymbol{E}$ and $\boldsymbol{H}$ are Root-Mean-Square (RMS) phasor quantities, $\varepsilon(r)$ is the permittivity and $\mu(r)$ is the permeability of the medium. In general, both $\varepsilon(r)$ and $\mu(r)$ must be either constant or smoothly varying in the region of interest. In this work, we assume a homogeneous medium in which both ε and μ are constant. The first two equations are rewritten

$$\nabla \times \boldsymbol{E}(r,\omega) + j\omega\mu\, \boldsymbol{H}(r,\omega) = 0, \tag{2.1}$$

$$\nabla \times \boldsymbol{H}(r,\omega) - j\omega\varepsilon\, \boldsymbol{E}(r,\omega) = 0. \tag{2.2}$$

By isolating $\boldsymbol{H}$ in (2.1) and substituting into (2.2), the vector Helmholtz equation can be obtained:

$$\nabla^2 \boldsymbol{E}(r,\omega) + k^2 \boldsymbol{E}(r,\omega) = 0, \tag{2.3}$$

given that $\nabla \times (\nabla \times \boldsymbol{E}(r,\omega)) = \nabla(\nabla \cdot \boldsymbol{E}(r,\omega)) - \nabla^2 \boldsymbol{E}(r,\omega)$, and $\nabla \cdot \boldsymbol{E}(r,\omega) = 0$ in a homogeneous medium. In the above equation (2.3), we defined $k^2 := \omega^2\mu\epsilon$, k denoting the wavenumber. The same equation holds for the magnetic field $\boldsymbol{H}$.

Luneberg-Kline anticipated solution (Ansatz)

Like Maxwell's equations, the vector Helmholtz equation (2.3) cannot be solved exactly in complex domains, which motivates the use of approximate solutions. Based on the work of Luneberg (among others), Kline established the following asymptotic expansions for a high-frequency electromagnetic field in source-free regions in an isotropic medium

$$\boldsymbol{E}(r,\omega) \sim e^{-jk\Psi(r)} \sum_{n=0}^{\infty} \frac{E_n(r)}{(j\omega)^n}, \tag{2.4}$$

$$\boldsymbol{H}(r,\omega) \sim e^{-jk\Psi(r)} \sum_{n=0}^{\infty} \frac{H_n(r)}{(j\omega)^n}, \tag{2.5}$$

where E_n and H_n are amplitude coefficients and Ψ is known as the wave function. A set of points r sharing the same phase satisfies: $\Psi(r) = \text{const.}$, and is called a wavefront. The symbol $\sim$ refers to asymptotic equality[1].

As the frequency ω tends to infinity, the terms related to $n \geq 1$ become negligible and only the first term of the sum remains ($n = 0$). Equations (2.4) and (2.5) simplify to

$$\boldsymbol{E}(r) \underset{\omega\to\infty}{\sim} E_0(r)e^{-jk\Psi(r)}, \tag{2.6}$$

$$\boldsymbol{H}(r) \underset{\omega\to\infty}{\sim} H_0(r)e^{-jk\Psi(r)}. \tag{2.7}$$

Fields satisfying this approximation are called *Geometrical Optics* (GO) fields.

High-frequency phenomena

The high-frequency hypothesis is at the heart of the GO theory and underlies the use of *rays* in RT. Contrary to what its name might suggest, the validity of this hypothesis is not based on a minimum threshold frequency. Instead, it is considered applicable if the properties of the medium and scatterer size parameters vary slowly over spatial scales on the order of a wavelength [22, Chapter 1]. In the settings of this work, this assumption is justified: typical physical dimensions — such as wall sizes — are on the order of meters, whereas the wavelengths involved are on the order of centimeters or below. Indeed, RSS-based indoor positioning systems typically operate at frequencies between 2.4 GHz – 5 GHz (e.g., Bluetooth, WiFi) and above 100 GHz (mmWaves) [20].

[1] I.e., becoming increasingly accurate as the angular frequency ω tends to infinity.

Ray trajectories

Substituting (2.6) into (2.3) and Maxwell's equations in a homogeneous medium, and carrying out further developments[2], yields a series of results. The first one is commonly known as the Eikonal equation:

$$\|\nabla\Psi\|^2 = 1. \tag{2.8}$$

A second result is the transport equation:

$$2(\nabla\Psi \cdot \nabla)E_0 + (\nabla^2\Psi)E_0 = 0. \tag{2.9}$$

We begin with the Eikonal equation. For GO fields (2.4, 2.5), we can develop the expression of the Poynting vector, which is the directional power flow per unit area. Precisely, it is defined as $\boldsymbol{S} = \boldsymbol{E} \times \boldsymbol{H}^*$, where $\boldsymbol{H}^*$ denotes the complex conjugate of the phasor quantity $\boldsymbol{H}$. Developing, we would derive that the power flows in the direction

$$\hat{\mathbf{s}} = \nabla\Psi,$$

which thus defines the GO *ray* direction. In this sense, a ray is defined by the direction in which the energy is propagated. By changing the coordinate vector variable r (defining the position in space) for the ray coordinate s (measured along the ray), the operator

$$(\nabla\Psi \cdot \nabla) = \hat{\mathbf{s}} \cdot \nabla = \frac{d}{ds},$$

is simply the directional derivative along the ray. From differential geometry, we also have: $\hat{\mathbf{s}} = \frac{dr}{ds}$, the unit vector tangential to the ray curve. With this, we have on the one hand,

$$\frac{d(\nabla\Psi)}{ds} = \frac{d\hat{\mathbf{s}}}{ds} = \frac{d}{ds}\left(\frac{dr}{ds}\right) = \frac{d^2r}{ds^2}. \tag{2.10}$$

On the other hand,

$$\frac{d(\nabla\Psi)}{ds} = (\nabla\Psi \cdot \nabla)\nabla\Psi = \nabla\left(\frac{1}{2}\|\nabla\Psi\|^2\right) = 0. \tag{2.11}$$

Joining the two above equations (2.10, 2.11) leads to:

$$\frac{d^2r}{ds^2} = 0. \tag{2.12}$$

Solving (2.12), the only solution has the form

$$r(s) = As + B, \qquad \text{for some constants } A,\, B \in \mathbb{R}.$$

This leads to a first conclusion:

> **Result 2.1** (Ray trajectories). GO rays in a homogeneous isotropic medium follow straight lines with direction $\hat{\mathbf{s}} = \nabla\Psi$.

The homogeneous isotropic medium hypothesis is important, since it is not the case in general inhomogeneous media.

[2]See [22] for detailed computations.

Phase and amplitude continuation

Knowing that the rays are straight lines normal to the wavefront (Result 2.1), we have

$$d\Psi = ||\nabla\Psi||\, ds = ds.$$

Integrating both sides from a reference point at $s = 0$ to s, we obtain

$$\Psi(s) = \Psi(0) + s.$$

> **Result 2.2** (Phase). The change in phase along a ray path is given by
>
> $$e^{-jk\Psi(s)} = e^{-jk\Psi(0)}\, e^{-jks}.$$

The factor e^{-jks} is the *phase shift* along the path.

A relation between the distance s along the ray from the reference point and the amplitude of the field, can also be derived. We consider an infinitesimally narrow ray tube surrounding the axial ray in the direction of $\hat{\mathbf{s}}$, as depicted in Figure 2.1. The wavefront associated with the reference point $s = 0$ satisfies $\Psi(0) = \text{const.}$, and has principal radii of curvature ρ_1 and ρ_2. Integrating equation (2.9) along the axial ray from 0 to s yields the formula for the attenuation of the amplitude along the ray path. More intuitively, the so-called *spreading factor* can also be derived from the conservation of power within the ray tube between successive wavefronts. Indeed, the power flow occurs solely in the ray tube, with no power transmitted in a transverse direction. Therefore, the spreading factor is the square root of the ratio of the respective areas of the wavefronts in the tube:

$$\left(\frac{\|E_0(s)\|}{\|E_0(0)\|}\right)^2 = \frac{dS_0}{dS} = \frac{\rho_1\rho_2\, d\phi_1 d\phi_2}{(s+\rho_1)(s+\rho_2)\, d\phi_1 d\phi_2} = \frac{\rho_1\rho_2}{(s+\rho_1)(s+\rho_2)}.$$

> **Result 2.3** (Amplitude). The amplitude decreases along the path, following
>
> $$E_0(s) = E_0(0)\sqrt{\frac{\rho_1\rho_2}{(\rho_1+s)(\rho_2+s)}}.$$

The factor $\sqrt{\frac{\rho_1\rho_2}{(\rho_1+s)(\rho_2+s)}}$ is the *spreading factor*, which accounts for the decrease in the amplitude of the field as a function of the distance along the ray path from the reference point $s = 0$.

Geometrical optics terms

From (2.6) and the three results above, we deduce the following expression for the propagation of the GO fields:

$$\boldsymbol{E}(s) = \boldsymbol{E}(0)\sqrt{\frac{\rho_1\rho_2}{(\rho_1+s)(\rho_2+s)}}\, e^{-jks}, \tag{2.13}$$

where $\boldsymbol{E}(0) = E_0(0)\, e^{-jk\Psi(0)}$ denotes the field value at the reference position $s = 0$ (with amplitude $E_0(0)$ and phase $\Psi(0)$). We note that, although the reasoning was done exclusively on the electric field $\boldsymbol{E}$, similar equations apply for (2.7) and the magnetic field $\boldsymbol{H}$.

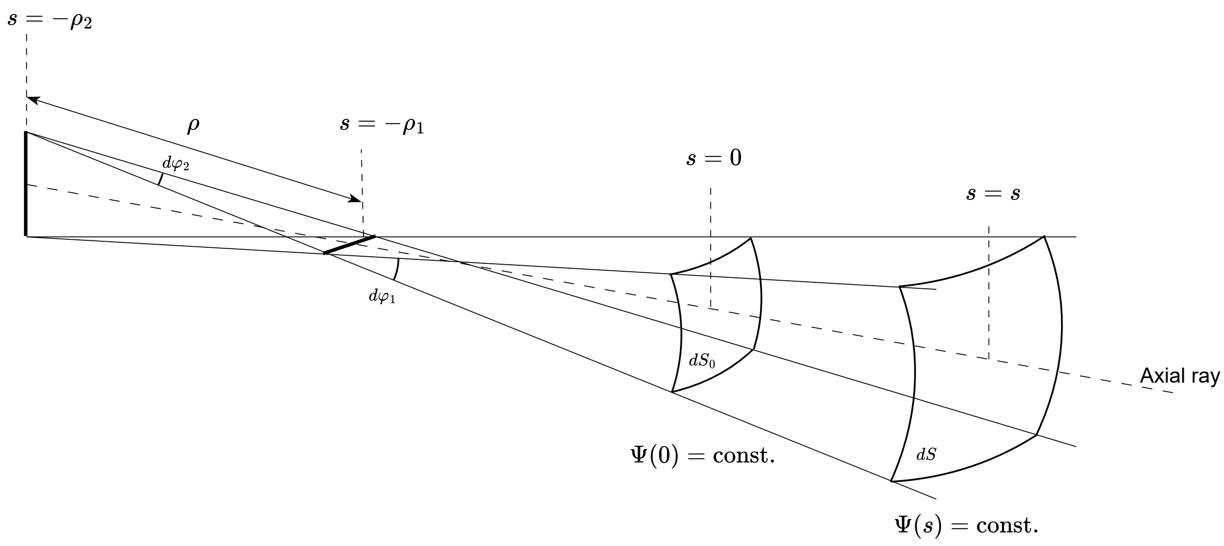


Figure 2.1: Infinitesimally narrow diverging astigmatic ray tube from theoretical development in [22].

Types of waves

Electromagnetic waves can be classified according to the geometry of their wavefronts. The three most common types are *planar waves*, *cylindrical waves* and *spherical waves*. For uniform plane waves, the surfaces of constant phase and the surfaces of constant amplitude are planar everywhere and coincident. For cylindrical and spherical waves, the wavefronts are respectively concentric cylinders and spheres.
In a plane wave, the electric field E_0, magnetic field H_0 and direction of propagation $\hat{\boldsymbol{s}}$ (unit vector) are mutually orthogonal and related through

$$H_0 = Y(\hat{\boldsymbol{s}} \times E_0), \qquad \text{where } Y = \sqrt{\frac{\epsilon}{\mu}}. \tag{2.14}$$

Y is sometimes called the *admittance*, and is the reciprocal of the impedance of the medium: $\eta = \sqrt{\frac{\mu}{\epsilon}}$. At sufficiently large distances from the source, cylindrical and spherical waves can be approximated by uniform plane waves. Indeed, the components of E_0 and H_0 in the direction of propagation usually decay rapidly, leading to the same equation (2.14). The fields are then said to be locally plane.

Special cases of GO

Considering these three special cases, (2.13) can be rewritten as follows.

- Plane waves ($\rho_1 \to \infty, \rho_2 \to \infty$):

$$\boldsymbol{E}(s) = \boldsymbol{E}(0)\, e^{-jks}, \tag{2.15}$$

- Cylindrical waves ($\rho_1 \to \infty, \rho_2 = \rho$):

$$\boldsymbol{E}(s) = \boldsymbol{E}(0) \sqrt{\frac{\rho}{\rho + s}}\, e^{-jks}, \tag{2.16}$$

- Spherical waves ($\rho_1 = \rho_2 = \rho$):

$$\boldsymbol{E}(s) = \boldsymbol{E}(0)\,\frac{\rho}{\rho + s}\,e^{-jks}. \tag{2.17}$$

This final expression (2.17) will be used to model the power transmitted from a transmitter to a receiver.

2.2 Simplified laws for interactions

The previous section described wave propagation in an unbounded medium using the Geometrical Optics (GO) model. Especially for indoor environments considered in this work, we are interested in the interactions of the rays with objects in the scene. The GO framework models interactions at idealized infinite boundaries between media, resulting in two types of interactions: *reflection* and *refraction* (often called *transmission*). Reusing the astigmatic ray tube model and with some further computations, [22] arrives to the law governing these reflected fields, presented hereafter[3]. To account for *diffraction*, GO must be extended through the Uniform Theory of Diffraction (UTD). It enables to model edge-diffracted fields in an accurate way [22]. Other types of interactions exist (e.g. diffuse reflection, vertex diffraction, etc) but are not mentioned here. Most RT algorithms only consider reflection and diffraction, whose laws are cited hereafter.

Law of reflection

When a ray interacts with a conducting surface, part of it is reflected on the surface acting like a mirror. Precisely, the incident and reflected ray satisfy the law of reflection[4]:

> **Result 2.4** (Law of reflection). The incident ray (unit direction $\hat{\boldsymbol{\imath}}$), the reflected ray (unit direction $\hat{\boldsymbol{r}}$) and the normal to the surface (unit direction $\hat{\boldsymbol{n}}$) at the interaction point are coplanar and the angle of incidence is equal to the angle of reflection. Mathematically,
>
> $$\hat{\boldsymbol{r}} = \hat{\boldsymbol{\imath}} - (\hat{\boldsymbol{\imath}} \cdot \hat{\boldsymbol{n}})\,\hat{\boldsymbol{n}}.$$

(a) Law of reflection

(b) Keller's law of diffraction on edge

Figure 2.2: Illustrations of the laws of reflection and diffraction.

[3] In indoor environments, refraction is often negligeable and neglected in the simplified model.

[4] The complete laws of reflection and diffraction also go through the polarization and phase shift, but we will not go deeper in this work. As will be justified in section 2.3, we will not consider phase and polarization.

Law of diffraction

When it interacts with an edge, the incident ray creates a continuum of diffracted rays forming a cone.

> **Result 2.5** (Keller's law of diffraction [23]). The diffracted rays (vectors d) form the lateral surface of a cone whose vertex is located at the interaction point between the incident ray (vector i) and the edge (unit direction $\hat{\boldsymbol{e}}$), and whose half-angle is equal to the angle of incidence. Mathematically,
>
> $$\frac{i \cdot \hat{\boldsymbol{e}}}{\|i\|} = \frac{d \cdot \hat{\boldsymbol{e}}}{\|d\|}.$$

These two simplified laws are illustrated on Figure 2.2. Finally, we note that the reflected and diffracted fields also behave as GO fields.

Ray tracing

Let $t \in \mathbb{R}^2$ denote the position of a transmitter and $r \in \mathbb{R}^2$ the position of a receiver. Based on the GO model and the reflection and diffraction laws discussed above, the electromagnetic field received at r can be approximated as the sum of the contributions from all feasible[5] ray paths connecting t and r. This process of identifying valid propagation paths is known as Ray Tracing (RT). The simplest type of path is the direct, unobstructed straight-line path between t and r, known as the Line-Of-Sight (LOS) path. Defining the *order* of a path as its number of interactions (reflections and diffractions), the LOS corresponds to the unique feasible path of order 0. A visualization of the feasible paths of orders 0, 1 and 2, in a square 2D scene, is shown on Figure 2.3.

Feasible rays are piecewise linear trajectories, completely characterized by the sequence of coordinates of their interaction points (reflections or diffractions). We denote the set of all such paths by $\mathcal{P}(t, r)$, where each element $p \in \mathcal{P}(t, r)$ is a *ray trajectory* defined by a list of the 2D coordinates of the interaction points. Since each sequence of interacting objects and associated types of interactions uniquely determines a corresponding ray trajectory (if feasible), we may equivalently define the set of *interaction paths* $\mathcal{I}(t, r)$ as the set of ordered lists of object indices involved and types of the interactions.

2.3 Path loss and received power

In the forward problem introduced in section 1.2, we aim to model the power transmitted from t to r, given a known scene.

As stated above, the received power can be approximated as the sum of the contributions from all feasible rays connecting t and r.

This approach corresponds to a *noncoherent* summation of multipath effects, as it neglects phase and polarization, and thus does not model interference between rays. This modeling choice is motivated by the high frequencies (i.e. short wavelengths) of the signals considered.

[5] I.e., paths that satisfy the laws of ray propagation and reflection and diffraction within the scene.

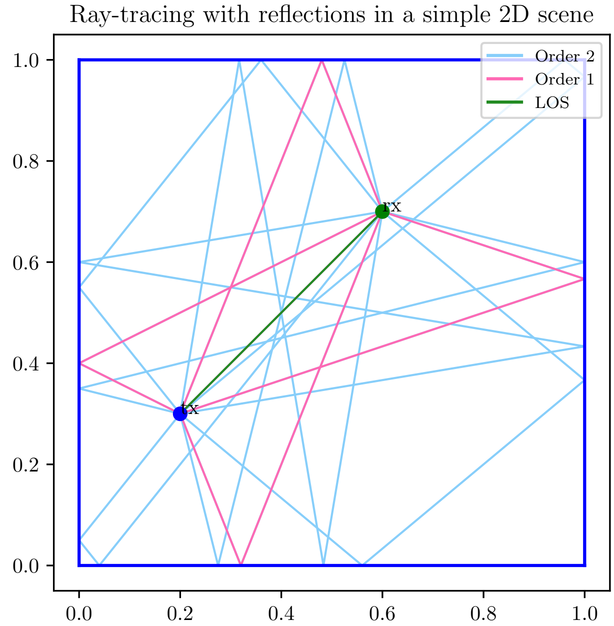


Figure 2.3: Illustration of all feasible ray paths connecting transmitter `tx` and receiver `rx` in a square 2D scene (walls in dark blue), with reflections up to order 0 (LOS, green), 1 (rose), 2 (light blue).

At such scales, inaccuracies in the scene representation — often exceeding the signal wavelength — can make interference modeling unreliable or even misleading. The noncoherent model is a more robust and practical approximation in this context.

Path loss in free space

We assume a transmitting antenna with a spherical radiation pattern, resulting in spherical wavefronts. From the definition of the Poynting vector and the expression of the electric and magnetic fields for a plane wave (2.14), the power density (also called intensity) of the electromagnetic field along a straight ray in free space, denoted by S, is given by [24]:

$$S := |\boldsymbol{S}| = |\boldsymbol{E} \times \boldsymbol{H}^*| = |\boldsymbol{E} \times Y\,(\hat{\boldsymbol{s}} \times \boldsymbol{E}^*)\,| = \sqrt{\frac{\epsilon}{\mu}}\,|\boldsymbol{E}|^2, \tag{2.18}$$

where we used the vector identity: $A \times B \times C = (A \cdot C) \cdot B - (A \cdot B) \cdot C$. Combined with the general form of the GO field (2.17), this expression implies that the power density decreases proportionally to the inverse square of the distance from the transmitter, measured along the ray. This attenuation is also called *path loss*.

Considering reflections and diffractions

When rays interact with objects, the same inverse-square law applies along their piecewise rectilinear trajectories $p \in \mathcal{P}(t, r)$. Let $\ell(p)$ denote the length of a feasible path p, defined as the sum of Euclidean norms of its segments. At each reflection or diffraction, only a fraction of the power is retained, modeled by a reflection or diffraction coefficient whose value depends on the material of the obstacle and on the angle of incidence.

In this work, we consider a simplified model, with only reflections and a unique coefficient, independent from the angle of incidence. For this approximation, typical values can be found in the literature for the reflection attenuation factor [25]. A value of 0.25 has been chosen for the ratio of the amplitudes of the reflected field to the incident one on indoor walls. Since the received power depends on the square of the amplitude of the field (2.18), the chosen attenuation coefficient of the power is 0.125. The attenuation coefficient corresponding to an interaction i (on the associated wall) is denoted $C_p^{(i)}$. In practice, this reflection coefficient could also be estimated as an inverse problem, from known power measurements with a known position of transmitter. This is further detailed in Chapter 7. The product of all such coefficients, along a path p counting Z interactions, is denoted $C_p \coloneqq \prod_{i=1}^{Z} C_p^{(i)}$.

We define $\Phi(t, r)$ as the total power received at r from a source at t in a given scene. Assuming noncoherent summation over feasible paths, we adopt the following model [24]:

Result 2.6 (Path loss and received power).

$$\Phi(t, r) = \bar{P} \sum_{p \in \mathcal{P}(t,r)} \frac{C_p}{\ell(p)^2}. \tag{2.19}$$

In the above, the power received is expressed comparatively to the reference transmitting power $\bar{P}$ of the transmitter, corresponding to the power in Watt received at a unit distance in LOS.

Remark: The set of feasible paths $\mathcal{P}$ theoretically contains infinitly many ray trajectories, with arbitrarily high number of interactions. In practice, we restrict to a finite maximum number of interactions, leading to a finite set $\mathcal{P}$. In general, contributions from higher orders become increasingly negligible, due to the product of coefficients $C_p^{(i)} < 1$ defining C_p in(2.19).

2.4 Forward problem formulation

This model forms the basis for our forward problem formulated hereafter[6].

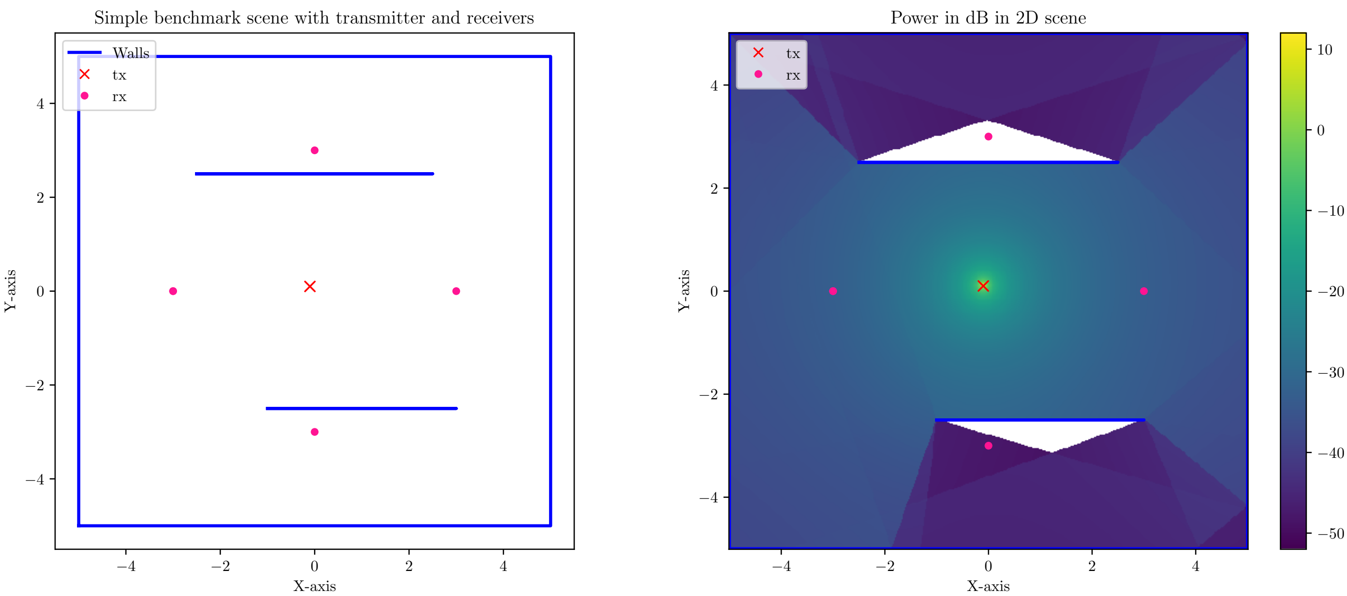
Forward problem (P)

For a transmitter positioned at t with reference transmitting power $\bar{P}$, m receivers with known positions $\bar{r}$, and known geometry and electromagnetic properties of the scene $\mathcal{S}$, the powers measured at the receivers b can be modeled by the following expression:

$$b = \mathcal{A}(t, \bar{r}, \mathcal{S})$$

$$\text{for} \qquad b_i = [\mathcal{A}(t, \bar{r}, \mathcal{S})]_i \coloneqq \Phi(t, r_i) = \bar{P} \sum_{p \in \mathcal{P}(t, r_i)} \frac{C_p}{\ell(p)^2 + \varepsilon}. \qquad \forall i = 1, \dots, m$$

[6]In comparison to (2.19), we added a small $\varepsilon \geq 0$ to the denominator. It should be small compared to $\ell(p)$ and have no impact in most cases; but ensures that the function Φ is bounded as $\ell(p)$ tends to 0 (i.e. when receiver and transmitter coincide).



(a) 2D simple benchmark square scene on $[-5,5]^2$, with 2 additional reflecting walls.

(b) Colormap of the power received in the scene for a transmitter positioned at $t = (-0.1, 0.1)$. White regions correspond to $0\,[W]$ $(-\infty\,[dB])$.

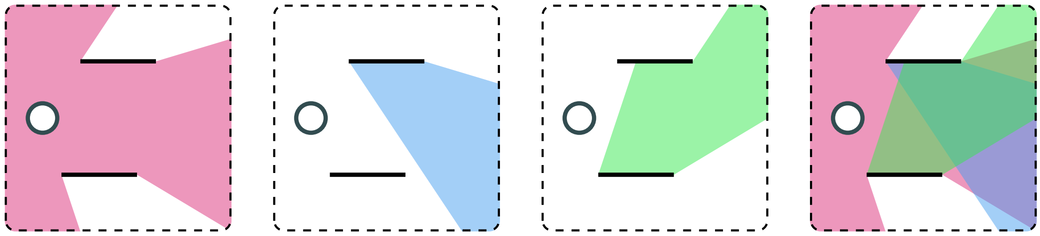
Figure 2.4: Simple benchmark scene and the power received at any point of this scene.

In practice, in the code, simulated measurements b_i are computed in decibels (dB), as commonly used for power measurements.

Figure 2.4 illustrates the received power at each point of a 500×500 discretized grid in a benchmark square scene with two additional reflective walls, expressed in dB and for a transmitter placed at $t = (0.1, -0.1)$. A maximum number of reflections of 2 was chosen here.

This colormap demonstrates that the received power, as modeled by the RT model, is only piecewise continuous across the domain.

According to the model **(P)**, this piecewise structure arises from changes in the set of feasible ray trajectories $\mathcal{P}(t, r)$ as a function of the transmitter and receiver positions. More precisely, as long as the set of feasible interaction paths $\mathcal{I}(t, r)$ remains unchanged, the reflection points slide continuously along the surfaces of the walls [26]. The total path length $\ell(p)$ then varies continuously, preserving the continuity of $\Phi(t, r)$. However, when a new interaction path becomes feasible or an existing one becomes infeasible (e.g., due to ray obstruction or reflection point falling beyond the edge of a wall) discontinuities appear in the power function. Therefore, the scene can be partitioned into continuity regions determined by the position of the transmitter and the obstacles' geometry. An example of these continuity regions on a simple scene for LOS and reflections on one of two walls in shown on Figure 2.5. This partitioning is further studied in [27], showing how to characterize these continuity regions, called *multipath cells*. Here, we simply highlight the key observation that $\Phi(t, r)$ is a piecewise continuous function of r, given a fixed t. By reciprocity, this also implies piecewise continuity in t. Determining the boundaries of the multipath cells being computationally extremely expensive, the positions of the discontinuities are considered unknown.



(a) LOS (b) Upper wall reflection. (c) Lower wall reflection. (d) Three types combined.

Figure 2.5: Example from [27, Figure 2], of multipath cells for individual and combined types of interaction, for a fixed Tx position (circle node) and Rx in the scene. Each color indicates a unique multipath cell, i.e., regions reached by the same type of interaction. When all types are considered simultaneously, more cells are created, each having a different multipath structure. Note that only two walls are considered in this example; the dashed contour is not a wall.

We emphasize that this piecewise structure is an artefact of the model we use, restricting to a finite number of reflections and neglecting diffraction and other types of interactions. In reality, solving Maxwell's equations exactly would lead to a continuous solution on a domain not considering the walls.

Increasing the maximal number of interactions allowed in the model increases its accuracy. It also increases the number of multipaths cells.

Summary

Indoor localization systems rely on high-frequency electromagnetic waves, whose propagation in complex environments is modeled by feasible ray paths connecting the transmitter to the receiver. In the following, only LOS propagation and finite-order reflections are modeled, while diffraction, refraction, scattering, and other effects are neglected. Although a good approximation, it leads to a model with a power function $\Phi(t, r)$ that is only piecewise continuous, whereas the exact solution would not present discontinuities. The assumptions underlying the model in this work are summarized as follows:

- Homogeneous medium;
- Short wavelength compared to the object size (high-frequency hypothesis);
- 2D model, assuming that the walls are vertical, that the transmitter and receiver share the same z-coordinate and neglecting floor and ceiling reflections;
- Diffraction, refraction and scattering effects negligible compared to LOS and reflections;
- Reflection coefficient independent of the incident angle and fixed to a chosen value;
- Sufficiently low reflection coefficient, so that paths with high order have negligible impact;
- Incoherent summation of the powers received from distinct paths.

These assumptions constitute a strong simplification of the underlying physics, resulting in a simplified forward model of wave propagation between the transmitter and receivers. We emphasize that this model could be refined to better reflect realistic propagation conditions and improve agreement with real measurements.

2.5 Ray tracing methods

Multiple methods are available in the literature to solve the RT problem of finding all feasible paths. They can be classified into two categories: exhaustive and non-exhaustive methods.

Exhaustive methods methodically compute all feasible paths between the transmitter and each of the m receivers in the scene, by formulating mathematically the problem of finding all such paths. They are mainly used in fields such as telecommunications, optics or acoustics where accuracy guarantees often outweigh computation speed. Among exhaustive methods, we can cite the Image method [28], the Fermat Path method [29], and the Min-Path-Tracing method [30].

Non-exhaustive methods on the other hand, rely on probability. The most common method is the Shooting-and-Bouncing-Rays (SBR) method, in which N_{rays} rays are launched from the transmitter in various directions. These rays are propagated through the environment, bouncing off obstacles according to the laws of reflection and diffraction. Those that finally reach sufficiently close to a receiver are retained. The main advantage of non-exhaustive methods is their linear dependence on both the maximum number of interactions and the number of walls in the scene, denoted Z and w respectively. In contrast, exhaustive methods typically exhibit polynomial complexity in w and exponential complexity in Z [31]. A brief comparison summary between the two classes is given in Table 2.1.

Criteria	Exhaustive Methods	Non-Exhaustive Methods
Complexity [31]	mw^Z	wZN_{rays}
Speed	Relatively slow, especially for large scenes (due to complexity)	Significantly faster on large scenes
Precision	Exclusively dependent on the precision of the description of the scene. Exhaustively finds all paths given the description.	Not exhaustive, depends on the number of rays N_{rays}. Can be made arbitrarily precise for sufficient number of rays launched.
Main fields	Telecommunications, optics, acoustics	Computer graphics
Typical methods	Image method [28], Min-Path-Tracing method [30], Fermat Path method [29]	SBR method [32]

Table 2.1: Comparison of exhaustive and non-exhaustive ray tracing algorithms.

Among the exhaustive methods, we detail hereafter the Image method, which we will be using in this work.

Image Method

The Image method is an exhaustive RT technique to compute feasible ray paths that involve LOS and reflections, up to a predefined maximum number of interactions. It assumes an exact geometric description of the scene and planar obstacles. It is based on the concept of *image points*.

The method can be decomposed into 4 steps:

1. We begin by listing all possible ordered sequences of objects in the scene and associated types of interaction (reflection on surfaces and diffraction on edges), with lengths ranging from 0 (LOS) up to a maximum order Z. Each such sequence defines a candidate interaction path. We note that this step is common to all exhaustive methods. These methods differ by the computation of the coordinates of the interaction points.

2. We then consider each sequence individually. Starting from the transmitter, we compute the position of its *image point*, reflected symmetrically across the first interaction wall. We denote this point I_1. We proceed by reflecting I_1 across the second object in the sequence to obtain I_2, and so on, until we reach the final image point after reflecting through all walls in the list [28].

3. We then trace the rays backward, starting from the receiver and connecting the image points, determining the interaction points on the walls.

4. If any of these interaction points falls outside of its respective wall segment, or if the ray is blocked by another obstacle on its way, the ray is considered invalid and is discarded.

Applying basic principles of synthetic geometry, we verify that valid paths constructed this way respect the law of reflection at each interaction. The Image method is illustrated in Figure 2.6.

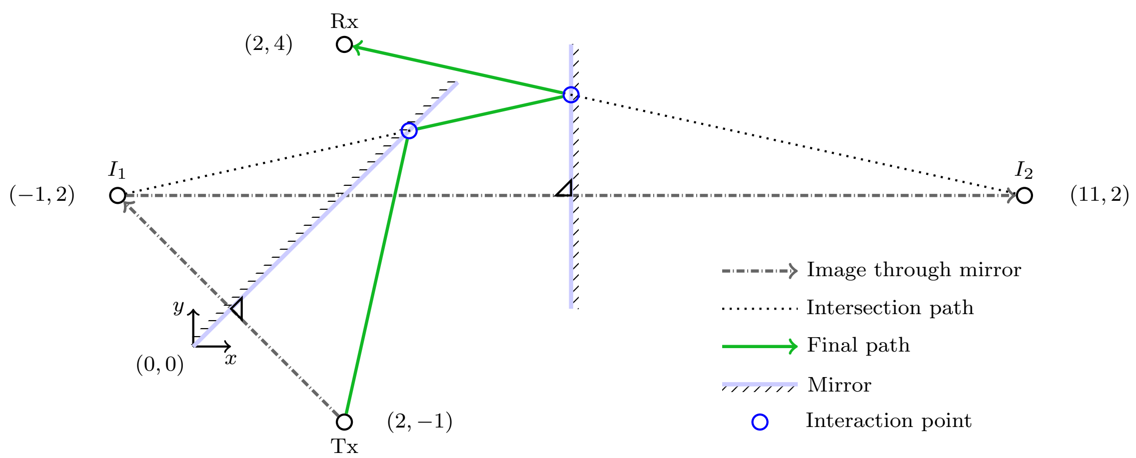


Figure 2.6: Illustration of the Image method for a ray between t (Tx) and r (Rx). The method determines the only valid path that can be taken to join Tx and Rx with, in between, reflection with two mirrors (the interaction order is important). Adapted from [30, Figure 5].

Remark 1: To account for diffraction or for nonplanar surfaces, the Min-Path-Tracing method could be used. It formulates the problem of finding the feasible paths as an optimization problem, given an implicit expression of the geometry of each wall or object in the scene. The complete formulation is detailed in [30].

Remark 2: The Image method only allows for planar surfaces, which can seem restrictive. In practice, it is not, as most representations of scenes consist of meshes, approximating any scene with triangles or quadrilaterals.

2.6 Differentiable ray tracer in telecommunications

> Using the chain rule is like peeling an onion: you have to deal with each layer at a time, and if it is too big you will start crying.
>
> *Anonymous professor*

In the inverse problem formulation discussed in section 1.2, the central challenge is to determine a *good* optimization step. When the objective function is differentiable, a standard approach is to move in the direction of the gradient, for a certain choice of stepsize. It corresponds to the Gradient Method (GM) and subsequent methods in optimization (accelerated gradient method, Adam, etc).

In our case, however, the objective function will include an evaluation of the forward problem, the RT simulation described earlier. The inherent complexity of this forward problem and its piecewise structure (Figure 2.4) make the symbolic calculation of derivatives tedious at best, and impossible to get a closed-form expression in most cases. Fortunately, symbolic differentiation is just one of the three main approaches to evaluating derivatives. The other two are finite differences and Automatic Differentiation (AD). We will return to finite differences in Chapter 5. Hereafter, we focus on the principle of AD, and how it can be useful in differentiable ray tracers for telecommunications.

AD is a technique that computes numerical values of the partial derivatives of any function up to machine-precision accuracy, relying on its implementation. It works by methodically decomposing the function into elementary operations and applying the chain rule step by step. Let us denote by $\mathrm{J}f$ the Jacobian matrix of a function f. Mathematically, the chain rule applied to a function defined by $f = (f_k \circ \cdots \circ f_1)$, and evaluated at x, writes[7]:

$$\mathrm{J}f = \prod_{i=1}^{k} \mathrm{J}f_i(x^{(i-1)}) \qquad \text{where } x^{(i)} := f_i(x^{(i-1)}),\ x^{(0)} := x.$$

The directional derivative v in the direction u is given by:

$$v = Jf \cdot u$$

AD appears in two modes. The *forward mode* is the most straightforward. It represents the whole function as a graph where each node corresponds to a basic operation in the

[7] Notations and example from [33].

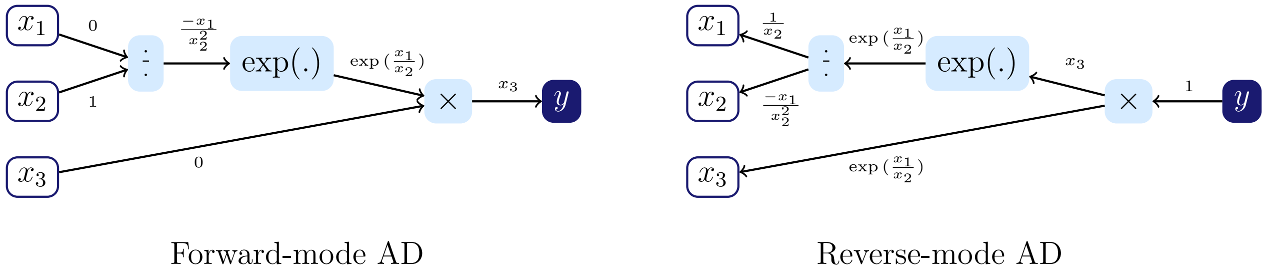


Figure 2.7: Computational graph and derivatives at the nodes in the application of forward and backward automatic differentiation to example function $f(x_1, x_2, x_3) = x_3 \exp(\frac{x_1}{x_2})$ adapted from [33]. The total derivative is computed by multiplying the weights along each edge of the graph, starting from the input (forward-mode) or the output (reverse-mode).

composition. The chain rule is then applied in the forward direction through this graph. Starting from the input nodes, the forward mode computes the desired partial derivative at each node by evaluating Jacobian-vector products, in what is referred to as a *forward-pass*. Mathematically,

$$v^{(i)} = \mathrm{J}f_i(x^{(i-1)})\, v^{(i-1)} \quad \text{for } i = 0, \dots, k, \quad \text{from } v^{(0)} = u \text{ until } v^{(k)} = v.$$

In practice[8], the function values at the nodes are augmented with the derivatives, and computations are done in an efficient way, leveraging the concept of *dual numbers* [34]. Having passed through the whole graph, it results in the partial derivative of the outputs with respect to the input direction u. Since each forward pass computes a single Jacobian-vector product, one pass is required per input variable. Each forward pass has the same memory complexity as a regular function call and requires no intermediate storage. The time complexity depends on the intermediate operations and dimension of the graph. When differentiating with respect to many input variables, it is preferable to turn to the *reverse mode*.

The *reverse mode* is less intuitive. It works in a forward-pass that creates the graph, computes and stores the values and partial derivatives of the elementary functions at each node, then in a backward-pass that propagates the partial derivatives backward, applying the chain rule. Mathematically,

$$v^{(i)} = (v^{(i+1)})^T \mathrm{J}f_i(x^{(i-1)}) \quad \text{where } i = 1, \dots, k \text{ and } v^{(k+1)} = w, v^{(1)} = v,$$

and w is the gradient of the chosen output of the last function f_k of the composition. In reverse mode, AD must be run once per output, making it often a preferable choice when the number of inputs is large compared to the number of outputs. Intermediate results from the forward-pass must be stored, generally implying higher memory cost however.

Examples of both modes applied to function $f(x_1, x_2, x_3) = x_3 \exp(\frac{x_1}{x_2})$ are shown on Figure 2.7. Nowadays, AD in its reverse mode is largely used to train neural networks, and known as *backpropagation* [35]. In this work, we will use it on the implementation of the Image method, to perform gradient-based optimization in Chapters 4 and 5.

[8]In this work, AD is performed using the JAX Python library https://docs.jax.dev/en/latest/autodidax2_part1.html

Chapter 3

Related work

Before presenting our approach in Chapter 4, this chapter provides an overview of existing methods for indoor localization. It situates the chosen approach with its advantages and disadvantages.

In general, Received Signal Strength (RSS) measurements are only one of several signal characteristics that can be exploited for wireless indoor localization. This chapter reviews the different types of information available, along with the corresponding general methods proposed in the literature. We then focus on the formulation as the inverse problem to Ray Tracing (RT), and indicate similar ideas appearing in both telecommunications and Computer Graphics (CG).

3.1 Types of information

Sensors or receivers — commonly referred to as Base Stations (BS) in this context [11] — can record various features from the received signal. The most common ones in wireless indoor source localization include:

- Time-of-Arrival (ToA), time it takes for the signal to propagate from the transmitter to the receiver — assuming time synchronization between the transmitter and receivers;
- Time-Difference-of-Arrival (TDoA), difference in arrival times at two distinct receivers — typically one being designated as the reference, and assuming time synchronization between receivers (avoiding the need of time synchronization with the transmitter);
- Angle (or direction) of Arrival (AoA/DoA), requiring antenna arrays at each BS, which increases infrastructure complexity and power consumption, making it less common;
- and Received Signal Strength (RSS), information equivalent to the received power at the receiver. It is often readily available in most wireless systems, with no hardware modifications needed [11].

For all aforementioned reasons, RSS is currently the preferred choice for indoor localization, supported by the widespread presence of wireless systems in modern environments [11]. While RSS measurements are less precise under LOS conditions, they are also less affected by the absence of LOS.

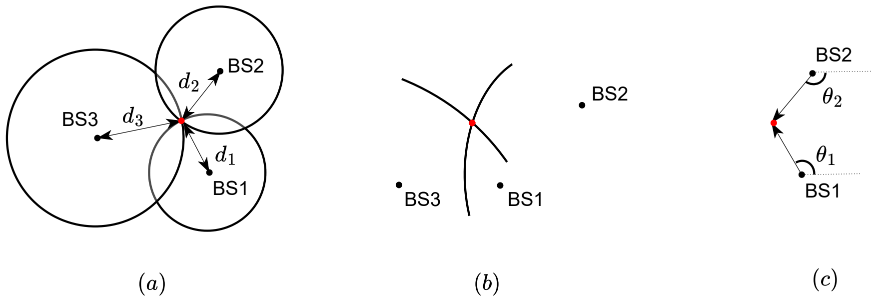


Figure 3.1: Illustration of simple LOS localization techniques using triangulation with information: (a) ToA or RSS, (b) TDoA, (c) AoA, adapted from [11].

3.2 General approaches

Two main directions have emerged for estimating the position of the emitting antenna from these measurements: triangulation and scene-analysis. A third approach is also mentioned in [36], relying on motion sensors on the object to be located. It is typically combined with one of the former two methods in robotics applications such as SLAM.

3.2.1 Triangulation

In an ideal obstacle-free environment, the locus of transmitter positions t that yield a given ToA or RSS measurement at a position r is a circle. Therefore, only three receivers are needed to determine the position of the transmitter, at the intersection of the corresponding circles. This simple technique is called triangulation [37, 11], illustrated on Figure 3.1. For TDoA and AoA measurements, the corresponding loci are a hyperbola and a line, respectively. In these cases, three and two receivers respectively are required, under ideal conditions.

In a typical indoor situation however, walls give rise to multipath effects and possible absence of LOS. Various estimators have been proposed [38], often assuming low impact from reflections, or aiming to mitigate or counteract their effect [11, 12, 13]. Triangulation, however, has initially been designed for outdoor localization, where LOS is available. In indoor settings, it treats NLOS effects as noise.

3.2.2 Scene-analysis

In contrast, the scene-analysis approach treats the complexity of the scene as a valuable source of information and uses the knowledge about particular characteristics or values in the scene. It typically relies on the concept of *fingerprint*, a signal signature of the scene. The simplest form of fingerprint consists of a vector of RSS measurements recorded for various transmitter positions on a predefined grid [11]. Scene-analysis typically involves two stages: an offline training phase, during which the fingerprints are collected, and an online localization phase, which relies on pattern-matching techniques [39]. The fingerprints being usually collected in the real environment, this technique does not require any virtual representation of the scene, nor forward propagation model.

This technique presents both advantages and disadvantages. As opposed to most triangulation techniques, NLOS effects are taken into account. However, it requires a sufficiently dense dataset of RSS measurements collected in the real environment, and the accuracy depends heavily on the number and spatial resolution of these measurements. Additionally, it shows low adaptability to environmental changes. The entire dataset has to be recomputed if the scene changes.

3.2.3 Virtual-scene-analysis

In this work, we consider a similar approach, but assuming that we cannot take a priori measurements on the real scene. Instead, we operate on a digital representation of the scene. This idea of a virtual model of the scene has already been studied in the literature. In [40], the authors use this virtual map and RT simulations to constitute a dataset, similarly to the classical scene-analysis method. On the contrary, we will use the RT simulations in the online phase, to produce estimated measurements for any desired candidate transmitter position, throughout the iterations. These candidate transmitter positions will be chosen strategically, aiming to estimate the true position through an iterative refinement process starting from an initial guess (Figure 1.3).

A clear advantage of this method is that it does not require acquisition of measurements in the real scene. Furthermore, it does not rely on a precomputed dictionary over a fixed grid of transmitter positions, making it highly adaptive to known changes in the environment.

The negative points include the dependence of the method on the availability and precision of a virtual twin scene. We assume to have access to a sufficiently precise description of the indoor environment, including the positions of the walls and their electromagnetic properties. In the chosen simplified forward model (section 2.4), electromagnetic properties resume to the reflection coefficient of the walls, but it could be much more complicated in a more precise model, accounting for polarization for example. As stated in section 2.3, estimating these electromagnetic properties can be challenging.

The approach followed in this work shares similarities with the one presented in [19], published in January 2025. The authors assumed to have access to two types of information about the signals: the power measurements (similar to RSS), and the delays. A delay is the minimum time it takes for a signal to travel from the transmitter to the receiver. Their forward model is therefore different, simulating both RSS and ToA measurements, and is based on the SBR RT method. As mentioned in Chapter 1 and further detailed in Chapter 4, the RT model introduces discontinuities in the loss function, from paths suddenly becoming feasible or infeasible. In [19], the authors proposed to use a smoothing strategy, to transform the discontinuous loss function into a continuous surrogate, and as a tentative to avoid numerous local non-global minima. This smoothing strategy is equivalent to the first one proposed in Chapter 5.

As a second contribution, the authors of [19] also proposed a way to improve their representation of the environment, by iteratively estimating the scene parameters as well. Indeed, the reflection coefficients of the walls, the exact positions of objects or obstacles in the scene, etc, might not be known precisely, and they can be precised using gradient-based methods for fine-tuning. A formulation of this scene parameter estimation problem, based on our forward model, is detailed in Chapter 7.

The inverse problem to RT is also studied in other fields, and differentiable ray tracers are increasingly used to solve inverse problems [41, 42].

3.3 Computer graphics and inverse rendering

RT was initially developed in the field of Computer Graphics (CG), to simulate the light paths between light sources and a camera, and render 3D scenes as 2D images. In this context, RT simulations are based on the SBR method (with sophisticated improvements [32]), but the objective is similar. Just as in the field of telecommunications, the inverse problem to RT can be considered. It is often called *inverse rendering* and consists in recovering parameters of the 3D scene, based on 2D rendered images. The parameters to recover can include the position of the camera, material parameters, lighting parameters [43], or even the positions of the vertices of a mesh representing an object [44]. Similarly again, differentiable ray tracers are developed in CG, so as to enable the use of gradient-based optimization methods on this inverse problem [45, 46]. This problem presents both similarities and differences with our inverse problem.

First, we mention that the forward model is slightly different in ray tracing for CG or for telecommunications. As explained in section 2.5, in telecommunications, the received power at the receiver is evaluated as the sum of the contributions of all feasible paths between the transmitter and receiver, from point to point. The implementation of the corresponding RT algorithms — such as the Image method (section 2.6) — are nearly everywhere differentiable and derivatives can be computed relying on AD. In CG, the measured power at a pixel can be defined as an integral of the pixel filter and radiance, over all rays between light sources and the camera, in the pixel area. This integral is approximated with a Monte Carlo estimator. The integrand being discontinuous, simply applying AD to compute the derivatives of the RT simulation would lead to biased gradient estimates (from the discontinuity at the edges and the Monte Carlo sampling[1]). Researchers have proposed many methods to compute gradients correctly [47, 48, 33].

The first difference between inverse rendering and our inverse problem is the ease with which the ray tracing simulation can be differentiated, and therefore the direct operator.

The forward model used in CG however, results in a continuous forward model, as opposed to the piecewise continuous forward model detailed in section 2.4.

The second main difference is the size of the problem. In our inverse problem, we recall that we have as many measurements as receivers (ideally, as few as possible of them), and only two unknowns: the $x-$ and $y-$coordinates of the transmitter position. In inverse rendering, the pixels of the given 2D image are all measurements that can participate in the loss function. Similarly to what we will formulate in Chapter 4, the inverse rendering

[1] Applying AD on the Monte Carlo estimate amounts to estimating the gradient of the true integral by an approximation of the integral of the gradient. However, in the case of a discontinuous integrand, the gradient and integral operators cannot be interchanged. AD in this case only produces a biased estimate.

problem is seen as an optimization problem, aiming to minimize the discrepancy between the measurements and the results of a forward simulation. On the other hand, the unknowns can be numerous as well, especially if we want to recover the position of vertices of a mesh [43, 49].

Moreover, in CG it is reasonable to assume that these measurements — the 2D image—, have little to no noise, if it comes itself from a rendering simulation. Precisely, if the image was generated with the same model as the forward model used in the inverse resolution procedure, there is no modeling noise. In wireless indoor source localization, the forward model for wave propagation introduces a first error, the modeling error. If the method is applied to noisy measurements, these will also add measurement errors (section 1.2).

A second difference is the number of unknowns and number and quality of available measurements in the inverse formulation.

Finally, inverse rendering often takes advantage from prior information about the parameters to recover (such as the smoothness of the mesh for example) [44]. It is a common practice in inverse problems, to have access to prior information about the parameters and include a regularization term to account for it in the objective function. In our formulation of the indoor source localization however, we have no prior information since the only variables are the 2D coordinates of the transmitter position.

A third difference is the absence of prior information in our inverse problem formulation.

In conclusion, our approach of indoor source localization as an inverse problem, and inverse rendering, share the same main principle and ideas. Namely, they aim to formulate the minimization of a loss function and try to solve it with iterative methods. There are, however, several critical differences, preventing the direct translation of CG methods (and good performance) to the wireless indoor source localization problem. We will see in the next chapter that the characteristics of the loss function will be different due to our choice of forward model **(P)** (section 2.4). We will try to adapt the resolution strategies accordingly.

Chapter 4

Indoor source localization as an inverse problem

This chapter formulates the indoor source localization task as an inverse problem to the forward model **(P)** introduced in section 2.4. We outline the mathematical formulation and highlight the specific challenges induced by the chosen forward model. In particular, we illustrate why direct application of standard Gradient Method (GM) to minimize the loss function may fail to converge to the minimizer.

4.1 Inverse problem formulation

The wireless indoor source localization problem is summarized in Table 4.1. Given received power measurements $\tilde{b} \in \mathbb{R}^m$ at m known receiver positions, the reference transmitting power $\bar{P}$ of the transmitter and a known scene $\mathcal{S}$ (geometry and electromagnetic properties), the objective is to estimate the position $\bar{t} \in \mathbb{R}^2$ of the transmitter in the scene.

We formulate this inverse problem mathematically as an optimization task, aiming to minimize the discrepancy between the acquired measurements $\tilde{b}$ at the receivers and those simulated by the forward operator $\mathcal{A}(t, \bar{r}, \mathcal{S})$ for a candidate transmitter position t. A common approach is to minimize a least-squares objective, which corresponds to using the mean squared error (MSE) as the loss function. We define:

Unknown	**Notation**
Transmitter position	$\bar{t} \in \mathbb{R}^2$

Given data	**Notation**
Power measurements at the receivers	$\tilde{b} \in \mathbb{R}^m$
Vector of the positions of the receivers	$\bar{r} \in \mathbb{R}^{m \times 2}$
Transmitter reference power	$\bar{P} \in \mathbb{R}_+$
Scene (geometry and electromagnetic properties)	$\mathcal{S}$

Table 4.1: Summary of unknowns and given data with their notations.

$$\mathcal{L}(t, \bar{r}, \mathcal{S}, \tilde{b}) = \mathrm{MSE}(\mathcal{A}(t, \bar{r}, \mathcal{S}), \tilde{b}) = \frac{1}{m}\|\mathcal{A}(t, \bar{r}, \mathcal{S}) - \tilde{b}\|_2^2 = \frac{1}{m}\sum_{i=1}^{m}\left([\mathcal{A}(t, \bar{r}, \mathcal{S})]_i - \tilde{b}_i\right)^2 \quad (4.1)$$

where $t \in \mathbb{R}^2$ is the candidate position of the transmitter and $\mathcal{A}$ is the forward operator described in section 2.5.

The MSE function itself is convex and differentiable. Furthermore, it implicitly assumes that the measurements $\tilde{b}$ are corrupted by an additional Gaussian noise. Minimizing the MSE corresponds to computing the maximum likelihood estimate of the parameter $\bar{t}$ under this assumption. Finally, its affine gradient will be used in section 5.1.2.

The mathematical formulation of the inverse problem is:

Inverse problem (I)

$$\min_{t \in \mathbb{R}^2} \mathcal{L}(t, \bar{r}, \mathcal{S}, \tilde{b}) = \min_{t \in \mathbb{R}^2} \frac{1}{m}\sum_{i=1}^{m}\left([\mathcal{A}(t, \bar{r}, \mathcal{S})]_i - \tilde{b}_i\right)^2 \quad (4.2)$$

In the later sections, we will consider fixed receivers' positions $\bar{r}$, a fixed scene $\mathcal{S}$ and fixed measurements $\tilde{b}$. When it is clear, we will denote $\mathcal{L}(t, \bar{r}, \mathcal{S}, \tilde{b})$ by $\mathcal{L}(t)$, emphasizing that the only input varying throughout the iterations is the transmitter's position t.

4.2 Challenges

Unfortunately, problem **(I)** is far from ideal and the objective function presents several significant challenges. In general, the function $\mathcal{L}$ is nonconvex, nonsmooth, and even discontinuous. More precisely, it is piecewise continuous, which renders most classical first-order optimization methods and their convergence guarantees inapplicable as it is. In the following, we demonstrate the piecewise continuity and nearly-everywhere differentiability of the loss function, and illustrate the risks when attempting to solve **(I)** simply using GM. We also emphasize a second and third additional challenges, inherent to most inverse problems: the potential ill-posedness of the problem and the impact of modeling noise.

Piecewise continuity

The piecewise continuity of the loss function with respect to the transmitter's position comes from the discontinuities of each entry of the forward operator $\mathcal{A}(., \bar{r}, \mathcal{S})$. From section 2.3, we know that $[\mathcal{A}(t, \bar{r}, \mathcal{S})]_i = \Phi(t, r_i)$ is a piecewise continuous function in t, with continuity cells depending on $\mathcal{S}$ and r_i. The continuity cells are thus different for each index $i = 1, \ldots, m$. Since the loss function is a sum of compositions of continuous functions (MSE) with these entries of the forward operator, the loss function is piecewise continuous with a cell pattern corresponding to the intersection of the cell patterns of the individual entries above.

Therefore $\mathcal{L}$ is piecewise continuous. Moreover, the number of continuity cells increases with the number of receivers, with the maximum number of reflections considered and with the number of walls in the scene. Logically their average size decreases correspondingly. Consequently, the loss landscape becomes increasingly fragmented, with numerous discontinuities, when increasing any of these three parameters.

From this structure, we deduce that $\mathcal{L}$ is nearly everywhere differentiable — differentiable everywhere except on the boundaries between continuity cells. Where it exists, the gradient can be evaluated via Automatic Differentiation (AD).

However, this piecewise continuity is a serious obstacle for first-order methods such as Gradient Method (GM). Figure 4.1 illustrates the poor performance of a basic and naive application of GM, applied to minimize the loss function $\mathcal{L}$ defined above (4.2), in the benchmark scene presented in Figure 2.4a. Given a maximum number of iterations T and a stepsize schedule $\{h_i\}_{i=1}^{T}$, GM writes:

$$\mathcal{L}(t^{(i+1)}) = \mathcal{L}(t^{(i)}) - h_i \, \nabla\mathcal{L}(t^{(i)}), \quad \text{for } i = 1, \ldots, T$$

where $\nabla\mathcal{L}(t^{(i)})$ is evaluated with AD, since the function $\mathcal{L}$ encompasses the computation of the ray paths with the Image method. Even for an initial point close to the minimizer, the discontinuities in the loss function lead the iterates away from the minimizer. The iterates frequently jump across discontinuities, changing between distinct continuity cells and causing the loss value to spike (Figure 4.1b).

GM and similar first-order methods are designed for continuous, differentiable, L-smooth functions, and will often fail to converge to the minimizer in our case.

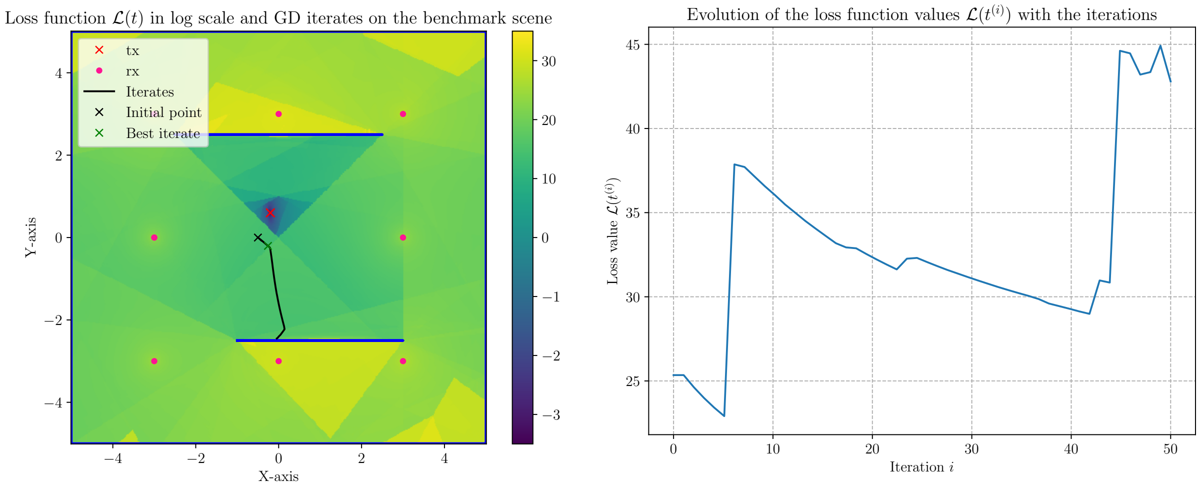
Zero-gradient regions

Another phenomenon that can hinder convergence of first-order methods is the presence of zero-gradient regions — plateaus in the objective function where the gradient vanishes. These regions correspond to areas of the domain where none of the receivers record any power received from the transmitter. While rare in practice, initializing the optimization procedure in such regions prevents any progress toward a minimizer, as gradient-based updates become ineffective.

Well-posedness of the inverse problem

Regardless of the optimization method used, difficulties inherent to inverse problems can also prevent successful recovery of the transmitter position. A fundamental issue is the possible ill-posedness of the problem — in particular, the non-uniqueness of the solution. Indeed, different transmitter positions may lead to very similar power measurements at the receivers, especially in environments with geometric symmetries or significant obstructions. This introduces ambiguities that reduce the reliability of any estimation method.

In general, we expect such ambiguities to become less likely as the number of measurement points m increases. It is therefore crucial to deploy a sufficient number of receivers in the



(a) Loss function and iterates on the benchmark scene.

(b) Loss function values at each iterates of the GM represented in (a).

Figure 4.1: Illustration of the bad behavior of a gradient descent on the discontinuous loss function in the benchmark scene with 8 receivers. (a) illustrates the loss landscape and the GM iterates, (b) shows the corresponding evolution of the loss function with the iterations. This example was implemented leveraging the `DiffeRT` Python module.

scene. However, there exists no theoretical criterion to determine the ideal number or placement of receivers, nor to guarantee well-posedness for a given configuration. The increase of the *success rate* with the number of receivers will be empirically verified in Chapter 6.

Modeling errors

Lastly, a major source of difficulty arises from modeling errors. These occur when the forward model $\mathcal{A}(., \bar{r}, \mathcal{S})$ does not exactly represent the true physical model, denoted $\mathcal{A}_0(., \bar{r}, \mathcal{S})$. Such discrepancies between the simulated and the actual measurements stem from simplifications in the simulation model: limiting the number of reflections, neglecting diffraction and other interactions, or approximating the electromagnetic properties of the materials in the scene $\mathcal{S}$ for instance.

On Figure 4.1 for example, we neglected all sorts of modeling errors. The supposedly real measures b where generated with the same model as the one of the forward problem **(P)** used in the definition of the loss function. We will observe in Chapter 6 the impact of one part of the modeling error, by generating the real *ground-truth* measurements with a maximum number of reflections of 5, while limiting to 2 in the forward model. Figure 4.2 quantifies the impact of neglecting higher-order reflections by showing their average contribution to received power across different receivers positions and 100 random transmitter positions in the benchmark scene, for various values of the reflection coefficient of the walls.

When modeling errors are present, the vector of simulated power measurements $\mathcal{A}(\bar{t}, \bar{r}, \mathcal{S})$ deviates from the true one $\mathcal{A}_0(\bar{t}, \bar{r}, \mathcal{S})$, even at the true transmitter position $\bar{t}$. Consequently, the value of the loss function evaluated at the true transmitter position $\bar{t}$ may be strictly positive. On the other hand, the global minimum of the loss function may no longer lie at $\bar{t}$.

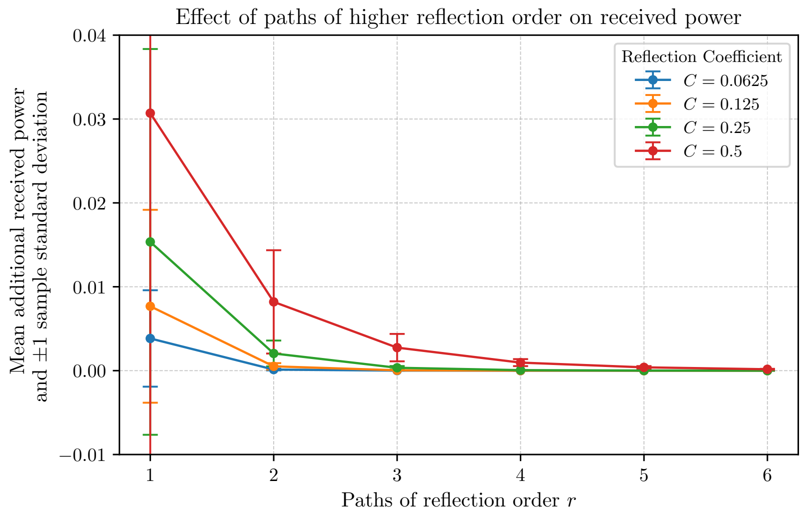


Figure 4.2: Power contributions of additional reflection of order r to the received power at 9 receivers, averaged over the receivers and for 100 random transmitter positions in the benchmark scene, for $\bar{P} = 1$, in Watt.

At the true position $\bar{t}$, we write

$$\mathcal{L}(\bar{t}, \bar{r}, \mathcal{S}, \tilde{b}) = \frac{1}{m}\|\mathcal{A}(\bar{t}, \bar{r}, \mathcal{S}) - \tilde{b}\|^2 = \frac{1}{m}\|\mathcal{A}(\bar{t}, \bar{r}, \mathcal{S}) - \mathcal{A}_0(\bar{t}, \bar{r}, \mathcal{S})\|^2.$$

In practice, this quantity is unknown and cannot be directly computed. For the theoretical part, we will assume that the modeling error is negligible — consequence of a sufficiently good forward model —, so that the inverse resolution on a virtual scene and with the forward model $\mathcal{A}$ has hope to indeed localize the transmitter. The modeling errors constitute however a major issue for our approach.
In fact, one can estimate the magnitude of this error of the forward model by comparing model predictions to real measurements. This estimation can serve as a reference threshold: below this level, further minimizing the loss function risks only overfitting the model discrepancies, rather than improving localization accuracy.

4.3 Proposed solutions

In the next chapter, we address some of the challenges described above. First, we eliminate discontinuities by approximating the discontinuous loss function $\mathcal{L}$ with a smooth surrogate, continuous and everywhere differentiable. In the same time, this smoothing may also reduce the influence of zero-gradient regions, and lead to a substitute objective function that has less local non-global minima than the original discontinuous loss function [50, 51].

Nevertheless, the smoothed loss remains nonconvex in general. Therefore, we run the optimization several times starting from different initial points (also called multistart), and we rely on a cheap initialization strategy to get these points.

Unfortunately, there is not much that we can do against well-posedness of the inverse problem and modeling errors, keeping the forward model defined above. These will be further evaluated in Chapter 6 and discussed in Chapter 7.

Chapter 5

Resolution strategies for source localization

In this chapter, we present resolution strategies addressing the challenges outlined in the preceding section. We begin by introducing various smoothing techniques aimed at handling discontinuities. In this work, the term *smoothing* will refer to the process of replacing a discontinuous function with a continuous surrogate.

5.1 Smoothing discontinuities

We consider three distinct smoothing strategies, each of which has been formulated and implemented. Throughout this section, let $\mathcal{G}_\sigma$ denote the two-dimensional Gaussian function with mean $\mu = (0,0) \in \mathbb{R}^2$ and covariance matrix $\Sigma = \sigma^2 I_2 \in \mathbb{R}^{2\times 2}$. For $t \in \mathbb{R}^2$, this function is given by

$$\mathcal{G}_\sigma(t) = \frac{1}{2\pi\sigma^2} e^{\frac{-\|t\|^2}{2\sigma^2}}.$$

It has been normalized so as to be a distribution. Let us recall the gradient of this Gaussian function. For $t \in \mathbb{R}^2$, it is expressed as

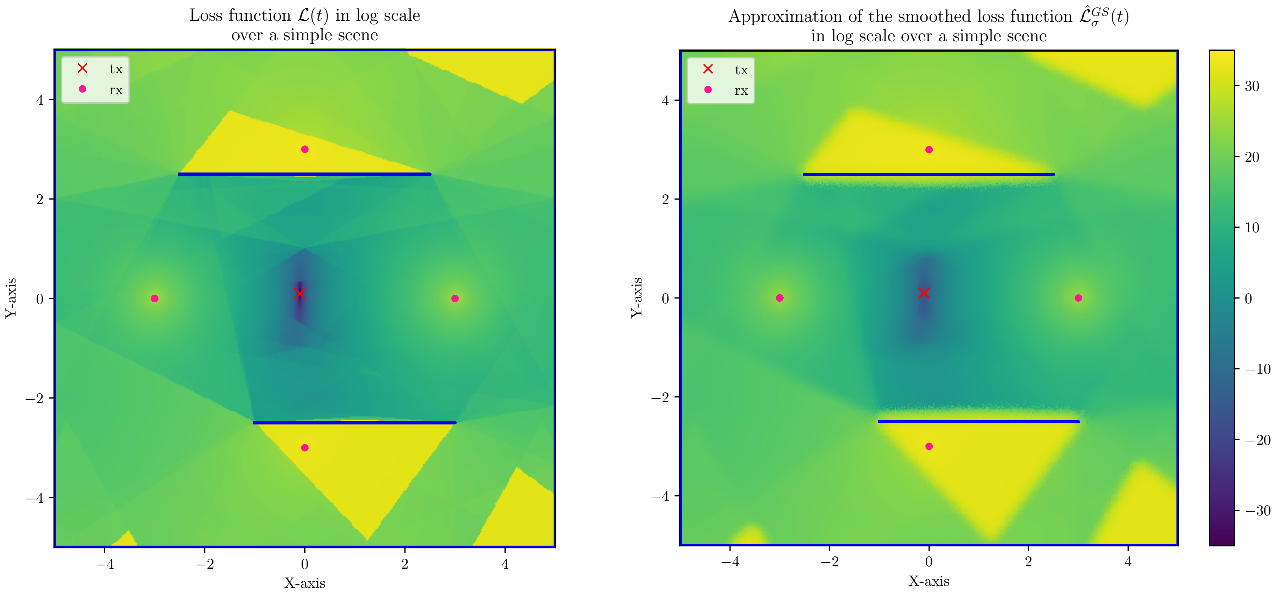
$$\nabla \mathcal{G}_\sigma(t) = \frac{-t}{\sigma^2} \frac{1}{2\pi\sigma^2} e^{\frac{-\|t\|^2}{2\sigma^2}} = \frac{-t}{\sigma^2}\, \mathcal{G}_\sigma(t). \tag{5.1}$$

5.1.1 Gaussian smoothing of the loss function

The first strategy for transforming the discontinuous loss function into a continuous one is to convolve it with a Gaussian kernel. Analogous to Gaussian blurring in image processing, this can be interpreted as smoothing the two-dimensional landscape of the loss function $\mathcal{L}$ over the scene. It is also called Gaussian Smoothing (GS) in the literature. The resulting smoothed loss function is defined as

$$\begin{aligned}
\mathcal{L}_\sigma^{\mathrm{GS}}(t) &\coloneqq (\mathcal{L} \,*\, \mathcal{G}_\sigma)(t) &(5.2)\\
&= \int_{u\in\mathbb{R}^2} \mathcal{L}(t-u)\, \mathcal{G}_\sigma(u)\, du &(5.3)\\
&= \mathbb{E}_{u\sim\mathcal{N}(0,\, \sigma^2 I_2)} \left[\mathcal{L}(t-u)\right]. &(5.4)
\end{aligned}$$

Since we do not have access to a closed-form expression for the discontinuous loss function $\mathcal{L}$,



(a) Colormap of the values of $\mathcal{L}(t)$ for $t \in [-5, 5]^2$.

(b) Colormap of the values of $\hat{\mathcal{L}}^{GS}_{\sigma}(t)$ for $t \in [-5, 5]^2$, $\sigma = 0.1$ and $N_\sigma = 100$.

Figure 5.1: Comparison of the original and smoothed loss landscapes, with colormaps of $\mathcal{L}$ and $\mathcal{L}^{LGS}$ respectively, in logarithmic scale over the simple scene.

the convolution (5.2) cannot be evaluated analytically. However, by interpreting the integral as an expectation over a Gaussian distribution (5.4), we can approximate it using a Monte Carlo (MC) estimator.

$$\mathcal{L}^{\mathrm{GS}}_{\sigma}(t) \approx \hat{\mathcal{L}}^{\mathrm{GS}}_{\sigma}(t) := \frac{1}{N_\sigma} \sum_{i=1}^{N_\sigma} \mathcal{L}(t - u_i), \qquad \text{for } u_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\, \sigma^2 I_2). \tag{5.5}$$

The MC estimator in (5.5) can be interpreted as a non-deterministic quadrature rule for approximating the exact integral (5.3) arising from the convolution. According to the Central Limit Theorem (CLT), it yields an unbiased estimate of $\mathcal{L}^{\mathrm{GS}}_{\sigma}(t)$ and converges with an asymptotic rate of $\mathcal{O}(N_\sigma^{-1/2})$ [52] to the true value of the integral.

Although this rate may seem slow compared to that of classical deterministic quadrature methods, we recall that it holds under the sole assumption that the second moment of the integrand is finite, i.e., $\mathbb{E}_{u \sim \mathcal{N}(0,\, \sigma^2 I_2)} \left[\mathcal{L}^2(t - u)\right] < +\infty$. This condition is significantly weaker than those typically required by deterministic quadrature rules, which rely on the integrand being sufficiently smooth to guarantee their respective (faster) convergence rates [53]. Such regularity assumptions are not satisfied in our context.

Figure 5.1 illustrates the effect of this smoothing, with a comparison of the original and smoothed loss landscapes over the simple benchmark scene.

If wanting to apply first-order methods to the smoothed loss function, the value of the smoothed loss function itself is often unnecessary. Instead, we need to evaluate its gradient at each iteration.

The gradient of the smoothed loss function $\mathcal{L}_\sigma^{GS}$ can be approximated using the following property of convolution:

Property 1: (from Leibniz integral rule). Let $g(.)$ be a continuous, differentiable and integrable function. Let $f(.)$ be an integrable function. Their convolution is differentiable, and

$$\nabla(f \ * \ g)(t) = (f \ * \ \nabla g)(t).$$

Applying this to the smoothed loss function (5.2), we obtain

$$\nabla\mathcal{L}_\sigma^{\text{GS}}(t) = (\mathcal{L} \ * \ \nabla\mathcal{G}_\sigma)(t) \tag{5.6}$$

$$= \int_{u\in\mathbb{R}^2} \mathcal{L}(t-u) \left(\frac{-u}{\sigma^2}\right) \mathcal{G}_\sigma(u)\, du$$

$$= \mathbb{E}_{u\sim\mathcal{N}(0,\sigma^2 I_2)} \left[\mathcal{L}(t-u)\left(\frac{-u}{\sigma^2}\right)\right]. \tag{5.7}$$

As before, we rewrite the convolution (5.6) and approximate the expectation (5.7) using MC sampling:

$$\nabla\mathcal{L}_\sigma^{\text{GS}}(t) \approx \widehat{\nabla\mathcal{L}_\sigma}^{\text{GS}}(t) := \frac{1}{N_\sigma}\sum_{i=1}^{N_\sigma} \mathcal{L}(t-u_i)\left(\frac{-u_i}{\sigma^2}\right), \qquad \text{for } u_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\sigma^2 I_2). \tag{5.8}$$

This estimator for the gradient of the smoothed loss function deserves a few comments.

- With this method, the ray tracer actually does not need to be differentiable. We only evaluate the loss function $\mathcal{L}$, requiring only zero-order oracle of the function, and therefore of the ray tracing simulation.
- Property 1 is often encountered with both functions f and g being continuous and continuously differentiable. In this case, the gradient of the convolution is the convolution of the gradient of any of the two functions, with the other one. Applied to our example, taking the gradient of the loss function $\mathcal{L}$ instead of the Gaussian kernel would lead to alternatively estimate the gradient with:

  $$\frac{1}{N_\sigma}\sum_{i=1}^{N_\sigma} \nabla\mathcal{L}(t-u_i), \qquad \text{for } u_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\, \sigma^2 I_2). \tag{5.9}$$

  This method, however, would lead to biased estimates of the gradient. First, at some points, the gradient of $\mathcal{L}$ may not be well-defined. More importantly, the general property does not hold when one of the two functions is only piecewise continuous. If the integrand is not a continuous function, we cannot exchange the gradient and the integral, to get the estimate (5.9).

- Equation (5.8) provides an unbiased estimate for the gradient of the smoothed loss function. On the other hand, high variance in this estimator could significantly hinder the convergence of the resulting optimization method. According to the Central Limit Theorem (CLT), the width of the confidence interval decreases at a rate of $\mathcal{O}(N_\sigma^{-1/2})$ as the number of MC samples N_σ increases. Without taking into account the function $\mathcal{L}$ — on which we do not have much insight —, we observe that the term $\mathcal{L}(t-u)\left(\frac{-u}{\sigma^2}\right)$ is inversely proportional to σ^2. This suggests that the variance of the estimator grows approximately as $1/\sigma^4$. Intuitively, the smaller the smoothing parameter σ, the more samples we need to get a MC estimate of the gradient with similar confidence interval. This observation should be seen as a general trend rather than a precise statement, since we are neglecting the influence of $\mathcal{L}(t-u)$, which likely varies less for small perturbations u around t (hence for smaller σ).

To reduce the variance of a Monte Carlo estimator, we can use diverse variance reduction techniques. Due to the lack of local information about $\mathcal{L}$ in general, we cannot justify or prove theoretical guarantees. Hereafter, we propose two common variance reduction techniques, which will be useful to make links in section 5.2.1.

- *Antithetic variates* consist in sampling symmetrically, for symmetric distributions. If the function is monotonic, it guarantees variance reduction. Even though we have no guarantee of monotonicity here, it was witnessed to work well in practice for small variance σ. That might be explained by a locally monotonic behavior of the function in a sufficiently small neighborhood.

- A second possible variance reduction technique is *control variate*. Its idea is to subtract, from the Monte Carlo terms, a term with zero-mean but as correlated as possible with the samples. Namely, we estimate the expectation of a random variable X as:

$$\mathbb{E}[X] = \mathbb{E}[X] - \mathbb{E}[Z] = \mathbb{E}[X-Z] \approx \frac{1}{N}\sum_{i=1}^{N} X_i - Z_i, \quad \text{for } \mathbb{E}[Z] = 0.$$

  In our case, we could use $\mathcal{L}(t)\left(\frac{-u}{\sigma^2}\right)$, as control variate. It has zero-mean, however we cannot guarantee correlation with $\mathcal{L}(t-u)\left(\frac{-u}{\sigma^2}\right)$ in all cases.

Algorithm 1 summarizes the minimization procedure implemented for this method and is given in the appendix. Each iteration requires the evaluation of N_σ RT simulations to estimate the gradient, and an additional simulation if we want to follow the evolution of the loss function values $\{\mathcal{L}(t^{(i)})\}_{i=1}^{T}$ with the iterations.

This technique — replacing the loss function with a Gaussian-smoothed surrogate — will hereafter be referred to as the LGS approach (Loss Gaussian Smoothing).

5.1.2 Transmitter's position tolerance smoothing

The motivation behind this second smoothing strategy lies in the observation that the discontinuities arise not from the loss function itself, but from the forward operator directly. Based on this, the objective is to transform the discontinuous received power function $\Phi(., r_i)$ into a continuous one with respect to the transmitter's position. It can be interpreted as a tolerance or uncertainty in the transmitter's location, so the received power is averaged around the estimated position. Alternatively, it can be viewed as treating the transmitter not as a point source, but as having spatial extent, with the transmitted power representing an average over neighboring locations. The resulting smoothed loss function is defined as

$$\mathcal{L}_\sigma^{\mathrm{TTS}}(t) \coloneqq \mathrm{MSE}(\mathcal{A}_\sigma^{\mathrm{GS}}(t), b),$$

where the forward operator with Gaussian-smoothed received power function is defined as

$$\begin{aligned}
\left[\mathcal{A}_\sigma^{\mathrm{GS}}(t)\right]_i &\coloneqq (\Phi(.,r_i) \, * \, \mathcal{G}_\sigma)(t) && (5.10)\\
&= \int_{u\in\mathbb{R}^2} \Phi(t-u, r_i)\, \mathcal{G}_\sigma(u)\, du \\
&= \mathbb{E}_{u\sim\mathcal{N}(0,\sigma^2 I_2)}\left[\Phi(t-u, r_i)\right]. && (5.11)
\end{aligned}$$

To compute the gradient of function $\mathcal{L}_\sigma^{\mathrm{TTS}}$, we apply the chain rule on the composition of functions:

$$\nabla\mathcal{L}_\sigma^{\mathrm{TTS}}(t) = \underbrace{\mathrm{J}\mathcal{A}_\sigma^{\mathrm{GS}}(t)^T}_{\in\mathbb{R}^{2\times m}\ \mathbf{(2)}} \cdot \underbrace{\nabla_x \, \mathrm{MSE}(x,b)|_{x=\mathcal{A}_\sigma^{\mathrm{GS}}(t)}}_{\in\mathbb{R}^m\ \mathbf{(1)}}.$$

Each of these factors can be approximated individually. The resulting estimation of the gradient $\nabla\mathcal{L}_\sigma^{\mathrm{TTS}}(t)$ of the continuous substitute loss function is denoted $\widehat{\nabla\mathcal{L}_\sigma}^{\mathrm{TTS}}(t)$.

1) Gradient of the MSE function

$$\begin{aligned}
\nabla_x \, \mathrm{MSE}(x,b)|_{x=\mathcal{A}_\sigma^{\mathrm{GS}}(t)} &= \nabla_x\left(\frac{1}{m}\sum_{i=1}^{m}(x_i-b_i)^2\right)\Bigg|_{x=\mathcal{A}_\sigma^{\mathrm{GS}}(t)}\\
&= \frac{2}{m}(x \, - \, b)\Big|_{x=\mathcal{A}_\sigma^{\mathrm{GS}}(t)}\\
&= \frac{2}{m}\left(\mathcal{A}_\sigma^{\mathrm{GS}}(t) - b\right). && (5.12)
\end{aligned}$$

In (5.12), it remains to approximate $\mathcal{A}_\sigma^{\mathrm{GS}}(t)$, since we have access only to the evaluations of the operator $\mathcal{A}$ and cannot perform convolution (5.10) exactly. This is done by approximating the expectation (5.11) with a MC estimator:

$$\mathcal{A}_\sigma^{\mathrm{GS}}(t) \approx \hat{\mathcal{A}}_\sigma^{\mathrm{GS}}(t),$$

where

$$\left[\hat{\mathcal{A}}_\sigma^{\mathrm{GS}}(t)\right]_i \coloneqq \frac{1}{N}\sum_{j=1}^{N_\sigma}\Phi(t-u_j, r_i), \qquad \text{for } u_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\sigma^2 I_2). \tag{5.13}$$

We note that, thanks to the choice of the MSE as loss function, its gradient is an affine function and the resulting estimator of this first factor is unbiased. Indeed,

$$\begin{aligned}
\mathbb{E}\left[\frac{2}{m}\left(\hat{\mathcal{A}}_\sigma^{\mathrm{GS}}(t) - b\right)\right]_i &= \frac{2}{m}\left(\mathbb{E}\left[\hat{\mathcal{A}}_\sigma^{\mathrm{GS}}(t)\right]_i - b_i\right)\\
&= \frac{2}{m}(\left[\mathcal{A}_\sigma^{\mathrm{GS}}(t)\right]_i - b_i).
\end{aligned}$$

2) Jacobian of the Gaussian-smoothed power measurements

$$
\begin{aligned}
\left[\mathrm{J}\mathcal{A}_\sigma^{\mathrm{GS}}(t)\right]_{i,:} &= \nabla\left[\mathcal{A}_\sigma^{\mathrm{GS}}(t)\right]_i \\
&= \nabla(\Phi(.,r_i) \, * \, \mathcal{G}_\sigma)(t) \\
&= (\Phi(.,r_i) \, * \, \nabla\mathcal{G}_\sigma)(t) \\
&= \int_{u\in\mathbb{R}^2} \Phi(t-u,r_i)\left(\frac{-u^T}{\sigma^2}\right)\mathcal{G}_\sigma(u)\,du \\
&= \mathbb{E}_{u\sim\mathcal{N}(0,\sigma^2 I_2)}\left[\Phi(t-u,r_i)\left(\frac{-u^T}{\sigma^2}\right)\right].
\end{aligned} \tag{5.14}
$$

Again, we rewrite the gradient of the smoothed power measurement at receiver i as an expectation (5.14), which we estimate using a MC estimator:

$$
\left[\mathrm{J}\mathcal{A}_\sigma^{\mathrm{GS}}(t)\right]_{i,:} \approx \frac{1}{N}\sum_{j=1}^{N_\sigma}\Phi(t-u_j,r_i)\left(\frac{-u_j^T}{\sigma^2}\right), \qquad \text{for } u_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\sigma^2 I_2). \tag{5.15}
$$

It is also an unbiased estimator.

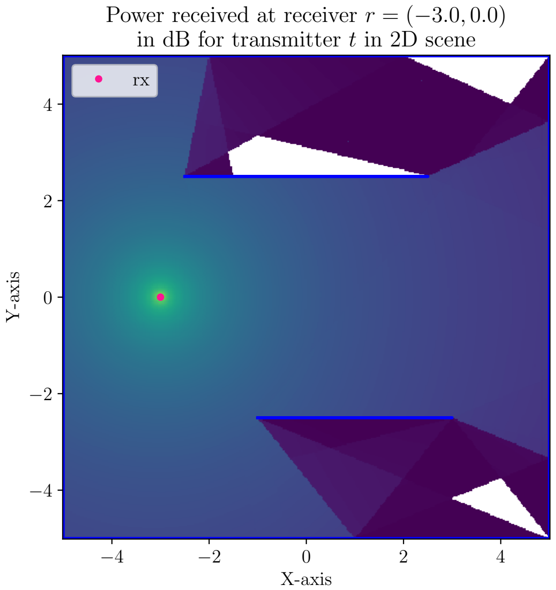
Combining (5.12), (5.13) and (5.15), if the samples from each of the two factors are independent, their product leads to an unbiased estimator of $\nabla\mathcal{L}_\sigma^{\mathrm{TTS}}(t)$ (leveraging $\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]$ if X,Y independent).

Figure 5.2 illustrates the effect of the Gaussian smoothing applied to function $\Phi(.,r_1)$, namely the power received at a receiver positioned at r_1, for a transmitter t anywhere in the scene. Note that we do not show the value of the smoothed loss function, nor did we estimate it. The reason is that, while the linearity of the gradient of the MSE allows for an unbiased estimate of this gradient, it is not the case for the associated loss function. We never estimate the smoothed loss function values during the minimization process. Instead, we follow the evolution of the discontinuous loss function values $\mathcal{L}$.

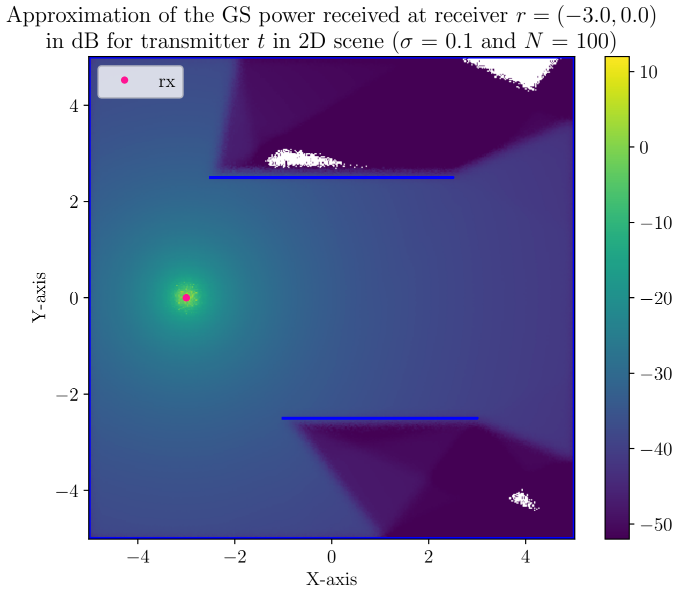
Just as in the first approach (LGS), we emphasize that this approach allows to compute an estimate of the gradient of the substitute loss function with only zero-order oracle of the RT simulation. Therefore, the ray tracer again does not need to be implemented in a framework allowing AD, in this case.

Algorithm 2 summarizes the minimization procedure implemented for this method and is given in the appendix. Each iteration requires the evaluation of $2N$ RT simulations to estimate the gradient, and an additional simulation if we want to follow the evolution of the loss function values $\{\mathcal{L}(t^{(i)})\}_{i=1}^T$. To compare with the LGS method above at equal cost, it suffices to choose $N = N_\sigma/2$.

Remark: In the above two methods, the Gaussian smoothing of either the loss or the power function might not be well-defined close to the boundary of the scene. Indeed, the perturbed transmitter position $t-u$ for $u\sim\mathcal{N}(0,\sigma^2 I_2)$ might fall outside of the domain of $\mathcal{L}$ or $\Phi(.,r)$. A condition must be chosen for these values; in this work, we fix it to the value of the function at the projection of point $t-u$ inside the scene.



(a) Colormap of the values of the power $\Phi(t, r_1)$ as a function of the transmitter's position for $t \in [-5, 5]^2$



(b) Colormap of the values of $\Phi_\sigma^{\mathrm{GS}}(t, r_1)$ for $r = (-3.0, 0.0)$ and $t \in [-5, 5]^2$, $\sigma = 0.1$ and $N_\sigma = 100$

Figure 5.2: Comparison of the original and smoothed received power at a receiver located at $r = (-3, 0)$ for a transmitter at $t \in [-5, 5]^2$ in the simple benchmark scene.

This technique — replacing the power received function with a Gaussian-smoothed surrogate — will hereafter be referred to as the TTS approach (for its interpretation as Tolerance on the Transmitter's position Smoothing).

5.1.3 Physics-based smoothing

This smoothing method was imagined by Jérome Eertmans, published in [14] and initially implemented in the 2D RT module `DiffeRT2D`[1,2]. Similar ideas also appear in CG for differentiable rendering [54].

This smoothing is fundamentally different from the two above. While the two above proposed smoothing methods were isotropic and agnostic of the geometry of the scene and position of the walls, we will see that this method presents a physical interpretation and is entirely based on the environment and obstacles. Indeed, it does not rely on the Gaussian smoothing of a function. Instead, it consists in directly eliminating the sources of discontinuities in the forward model.

In practice, the main sources of discontinuities in functions $\Phi(., r_i)$ for $i = 1, ..., m$ are the decisions of paths' feasibility: whether a path is blocked by a wall (occlusion by obstacle) and

[1] https://differt2d.eertmans.be/latest/quickstart.html. In this work, we will use `DiffeRT` rather than `DiffeRT2D`, for reasons of efficiency and execution time.

[2] For this Master's thesis, this smoothing was implemented and added to the `DiffeRT` module. The version used for the plots is available at https://github.com/SophieL1/DiffeRT. This third smoothing technique was not a personal idea, but I contributed to adding it to the `DiffeRT` module.

whether a reflection is feasible (i.e., the interaction points falls inside the wall's boundaries). It corresponds to the fourth step in the implementation of the Image method discussed in section 2.5. These conditions are checked based on the Möller-Trumbore intersection algorithm [55] applied on rays and triangles or quadrilaterals of a mesh representing the scene.

These binary decisions can be seen as step functions, assigning values 0 or 1:

$$\theta(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

The idea of this Physics-Based Smoothing (PBS) is to replace the discontinuous step function θ with a continuous and everywhere differentiable function s. This substitute function s must be parametrized by some smoothing parameter $\alpha \in \mathbb{R}^+$ [14], and must converge to the discontinuous step function θ for large values of the parameter α. Namely, $s : (x, \alpha) \in \mathbb{R} \times \mathbb{R}^+ \mapsto s(x, \alpha) \in [0, 1]$ satifies:

$$\lim_{\alpha \to +\infty} s(x, \alpha) = \theta(x).$$

We also impose the following characteristics:

1. Same values as function θ at the limits for $x \to -\infty$ and $x \to +\infty$:

$$\lim_{x \to -\infty} s(x, \alpha) = 0 \quad \text{and} \quad \lim_{x \to +\infty} s(x, \alpha) = 1,$$

2. Monotonically increasing function s,

3. Symmetry:

$$s(0, \alpha) = \frac{1}{2} \quad \text{and} \quad s(x, \alpha) - s(0, \alpha) = s(0, \alpha) - s(-x, \alpha).$$

The function chosen in this work is the sigmoid with parameter α:

$$\text{sigmoid}(x, \alpha) = \frac{1}{1 + e^{-\alpha x}}.$$

Its graph is shown on Figure 5.3 for various values of parameter α. While showing everywhere differentiability, the inconvenient of this choice is its theoretically infinite support. We note that a hard sigmoid would be another possible choice, leading to a finite support — and therefore a lot less rays —, but not everywhere differentiable anymore.

Figure 5.4 illustrates the smoothed received power function, denoted by $\Phi^{\text{PBS}}(., r_1)$ and obtained using the PBS method, for various values of the parameter α. This smoothing is not isotropic as the Gaussian one, but has a physics interpretation, as follows. The smoothing is involved in:

1. Depth-wise attenuation through walls: In the absence of smoothing, rays are completely blocked when encountering obstacles. With PBS smoothing, rays can partially go through walls, but are attenuated progressively as we move further beyond the wall, depending on the value of α.

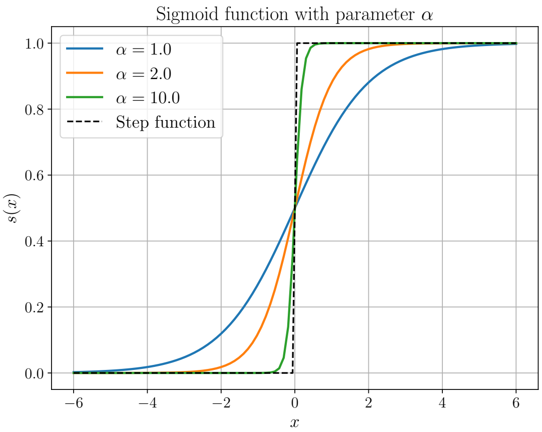
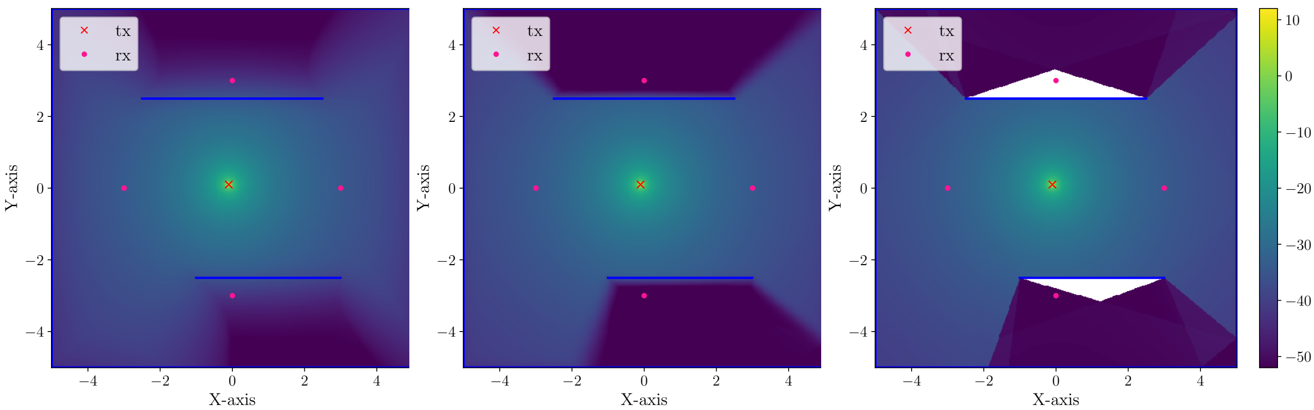


Figure 5.3: Sigmoid function for different values of parameter $\alpha = 1, 2$ and 10. The step function is a representation of the implementation of hard decisions in the RT algorithm, while the sigmoid substitutes illustrate the impact of the PBS strategy.

2. Edge softening for reflections and occlusions: Smoothing also modifies how rays interact with wall surfaces in the lateral directions. Instead of having sharp obstacle boundaries, this approach introduces soft transitions at the edges. Reflections or intersections occurring near an edge are smoothly weighted based on a substitute function, yielding a continuous transition from fully blocked to fully unobstructed.

As a consequence, this smoothing strategy is a modification in the forward model, and will drastically impact the number of feasible rays in the RT simulation. This in turn impacts the execution time of the algorithm.

We note that, opposite to the smoothing factor σ, the smoothed function tends to the original one for **increasing** values of α. For this reason, we will sometimes call it a *realism* factor in this work.



(a) $\alpha = 10$ (b) $\alpha = 50$ (c) $\alpha = 10^{10}$

Figure 5.4: Comparison of the values of the received power $\Phi^{\mathrm{PBS}}(t, r)$, in dB, for a fixed transmitter's position $t = (-0.1, 0.1)$ and over the simple scene $r \in [-5, 5]^2$, with approximation of parameter $\alpha = 10,\ 50$ and 10^{10}. For $\alpha = 10^{10}$, the received power with approximation is nearly everywhere equal to the original received power (Figure 2.4b).

The smoothed received power function $\Phi^{\text{PBS}}(t, r)$ is continuous in both its arguments, therefore specifically with respect to the position of the transmitter t. It implies the definition of the associated smoothed forward model $\mathcal{A}^{\text{PBS}}(t, \bar{r}, \mathcal{S})$ defined as

$$[\mathcal{A}^{\text{PBS}}(t, \bar{r}, \mathcal{S})]_i := \Phi^{\text{PBS}}(t, r_i), \qquad \forall i = 1, \ldots, m,$$

and the resulting PBS loss function:

$$\mathcal{L}^{\text{PBS}}(t, \bar{r}, \mathcal{S}, \tilde{b}) = \frac{1}{m}\|\mathcal{A}^{\text{PBS}}(t, \bar{r}, \mathcal{S}) - \tilde{b}\|^2.$$

This loss function is itself continuous with respect to the position of the transmitter t.

Opposite to the first two approaches presented (LGS and TTS), the ray tracer used with this approach **needs** to be implemented in a framework allowing the use of Automatic Differentiation (AD). Indeed, the PBS smoothing transforms the discontinuous loss landscape into a continuous one by modifying the forward model, and the gradient of the continuous loss function resulting from this new forward model needs to be evaluated using AD.

5.2 Optimization methods

In this section, we discuss the optimization methods employed to solve the inverse localization problem. We begin with a class of derivative-free optimization (DFO) methods that could be applied directly on the discontinuous objective function $\mathcal{L}$. Taking a closer look into these methods, we highlight a key insight: the first two smoothing strategies introduced earlier — namely, LGS and TTS — are not only heuristic tools, but correspond precisely to Gaussian smoothing techniques used in DFO. This connection provides theoretical justification for applying DFO to our problem. We then recall standard first-order methods that can be employed with the third smoothing strategy (PBS), which provides a smooth and everywhere differentiable forward model.

5.2.1 Derivative-free optimization methods

In our inverse problem, we initially aim to solve an optimization problem of the form

$$\min_x f(x),$$

$f : \mathbb{R}^n \to \mathbb{R}$, with the function f being piecewise continuous.

Even if most optimization methods and associated guarantees were designed for continuous function, we could try to apply them to our problem. We focus hereafter on one of the most simple DFO methods, after the *random search* — which consists in sampling a point y randomly according to a Gaussian distribution around the current position x and move to y if $f(y) < f(x)$ [56]. Namely, we consider the Random Finite Differences (RFD) methods, first presented by Polyak in [57], further analyzed by Nesterov in [56].

The principle of this method is, instead of taking a step proportional to the gradient of the function (like in the GM), we take a step proportional to a partial derivative in a random direction, estimating this partial derivative using zero-order oracles and finite differences.

Given a stepsize schedule $\{h_k\}_{k=1}^{T}$, it lead to methods such as [56]

Random forward finite differences:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - h_k\, g_\nu(x^{(k)}), \\ g_\nu(x) &= \frac{f(x+\nu u) - f(x)}{\nu} \cdot Bu, \quad \text{where } u \sim \mathcal{N}(0, B^{-1}). \end{aligned} \tag{5.16}$$

Random centered finite differences:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - h_k\, \tilde{g}_\nu(x^{(k)}), \\ \tilde{g}_\nu(x) &= \frac{f(x+\nu u) - f(x-\nu u)}{2\nu} \cdot Bu, \quad \text{where } u \sim \mathcal{N}(0, B^{-1}). \end{aligned} \tag{5.17}$$

One can see that these methods relate to the first (LGS) approach. Actually, in all generalities, the RFD methods are justified by a Gaussian smoothing. This general Gaussian smoothing consists in convoluting the function f with a centered Gaussian kernel with precision matrix B (symmetric positive-definite) [58]. The resulting Gaussian smoothed function is

$$f_\nu(x) = \sqrt{\frac{\det(B)}{(2\pi)^n}} \int_{\mathbb{R}^n} f(x+\nu u) \exp\left(-\frac{1}{2} u^\top B u\right) du.$$

Computing the gradient of this smoothed loss function yields

$$\begin{aligned} \nabla f_\nu(x) &= \frac{1}{\nu} \sqrt{\frac{\det(B)}{(2\pi)^n}} \int_{\mathbb{R}^n} f(x+\nu u) \exp\left(-\frac{1}{2} u^\top B u\right) Bu\, du \\ &= \frac{1}{\nu} \sqrt{\frac{\det(B)}{(2\pi)^n}} \int_{\mathbb{R}^n} (f(x+\nu u) - f(x)) \exp\left(-\frac{1}{2} u^\top B u\right) Bu\, \mathrm{d}u. \end{aligned}$$

In particular, for a function in $\mathbb{R}^2$ and choosing $B = \frac{1}{\sigma^2} I_2$ and $\nu = 1$, we find back exactly the Gaussian smoothing described in section 5.1.1. In particular, [56] details the centered and forward finite differences schemes, which correspond respectively to:

- Choosing $N_\sigma = 2$ with antithetic sampling (sampling symmetrically),
- Choosing $N_\sigma = 1$ and adding the $f(x)\left(\frac{-u}{\sigma^2}\right)$ term (which can be seen as a control variate), in the LGS method.

Both methods rely on an estimate of the gradient computed based on 2 RT simulations only. The advantage of forward finite differences is that we already have $f(x)$ evaluated, in case we want to track the evolution of the loss function values throughout the iterations. It saves one additional RT simulation per iteration. In Chapter 6, we compare the performance of these different ways of estimating the gradient, on a typical scenario.

This connection shows that the proposed LGS smoothing approach is actually equivalent to applying a well-established DFO method to the discontinuous loss function. This equivalence highlights that the use of this DFO method might be justified in our piecewise continuous settings. The TTS method can be interpreted as the application of the exact same DFO methods on a composite function. As such, both LGS and TTS methods fall within the broader framework of zero-order optimization strategies based on Gaussian smoothing.

5.2.2 First order optimization methods

In the previous section, we discussed how the LGS and TTS strategies yield approximate gradients of a smoothed version of the loss function via stochastic sampling. On the contrary, the PBS smoothing leads to an everywhere differentiable loss function for which the gradients can be computed directly using AD. In this last case, we can apply classical scheme for first order optimization such as the Gradient Method (GM), or more evolved scheme for the computation of the stepsize: Adam, etc. We note that these methods were initially designed for continuous, L-smooth and convex functions. In the third method, it is applied to the continuous PBS loss function.

5.2.3 Additional remarks

- We emphasize the importance of the initial point, given the nonconvexity of the loss function. In the next section, we list some initialization strategies to hopefully find a good initial iterate (to be in the basin of convergence).
- Since the transmitter position is known to belong to the predefined scene, a constraint should actually be added to the optimization problem. Namely, t should belong to the feasible set, the interior of the scene. To account for this constraint, we simply project all iterates inside the scene, assuming that the scene is a closed convex set on which projection is easy (e.g., a square).
- The loss function was observed to have very varying gradient norms. This can lead to very large steps in case of large gradient norm. The behavior of the function however is quickly varying inside the scene, and large steps are not a good idea. Therefore a maximal norm was set for the optimization step. This is sometimes called gradient clipping.

5.3 Initialization

Given the nonconvexity of the smoothed loss function, the success of any method (DFO corresponding to LGS or TTS approaches, or first order on the PBS loss function) will have a strong dependence on initialization. Therefore we employ a multistart strategy: the optimization is run independently from N_{init} different initial points, and the solution achieving the lowest final loss value is retained. The quality of the initial points directly affects the effectiveness of the multistart strategy. To construct a set of plausible candidates, we proceed with the following steps:

1. Grid sampling: We begin with a uniform grid of fixed resolution over the scene.
2. Feasibility filtering via LOS rejection: For each grid point x, we simulate the LOS power received at each receiver. If there exists a receiver r_i for $i = 1, \ldots, m$ such that: *i)* the LOS path from x to r is unobstructed, and *ii)* the simulated LOS power exceeds the measured power $\tilde{b}_i$, then the grid point x is deemed implausible and discarded. This step is justified because additional NLOS components (from paths with higher reflection orders) can only increase the total received power. Therefore, if the LOS alone already exceeds the observed value, the candidate x cannot plausibly be the transmitter's true location. In practice, we introduce a tolerance factor so as to not eliminate too many

points of the grid. We note that evaluating only LOS is very cheap compared to the Image method employed for higher orders.

3. Ranking and selection: Among the remaining plausible candidates, we rank the points according to their distance to the receiver(s) with the highest measured received power. We select the N_{init} closest candidates to initialize the optimization, making the assumption that the transmitter is more likely to be near these dominant receivers.

Summary

The loss function $\mathcal{L}$ defined in section 1.2 is piecewise continuous. Instead of directly minimizing this discontinuous function, we can minimize a continuous substitute: $\mathcal{L}^{\text{GS}}$, $\mathcal{L}^{\text{TTS}}$ or $\mathcal{L}^{\text{PBS}}$.

- In the LGS approach, we take as substitute the Gaussian smoothing of the loss function. This is equivalent to applying DFO methods, estimating the gradient of the substitute loss function based on zero-order oracles of the forward model. At each iteration, we take a step in the opposite direction to this estimated gradient:

$$t^{(k+1)} = t^{(k)} - h_k \, \widehat{\nabla\mathcal{L}}^{\text{GS}}_{\sigma_k}(t), \quad \text{for } k = 0, \dots, N_{\text{iter}} - 1.$$

  In all generalities, the smoothing factor σ_k can take different values at each iteration.

- In the TTS approach, it is the received power function that is replaced by its Gaussian-smoothed substitute. Again, this is equivalent to applying DFO methods, for composite functions here. We estimate the gradient of the resulting continuous loss function, again based on zero-order oracles of the forward model. At each iteration, we take a step in the opposite direction to this estimated gradient:

$$t^{(k+1)} = t^{(k)} - h_k \, \widehat{\nabla\mathcal{L}}^{TTS}_{\sigma}(t), \quad \text{for } k = 0, \dots, N_{\text{iter}} - 1.$$

- In the PBS approach, it is the forward model that is being modified, eliminating directly the sources of the discontinuities. The associated smoothed loss function is continuous with respect to the position of the transmitter, and its gradient can be evaluated at any point in the scene leveraging AD. At each iteration, we take a step in the opposite direction to this gradient:

$$t^{(k+1)} = t^{(k)} - h_k \, \nabla\mathcal{L}^{PBS}(t), \quad \text{for } k = 0, \dots, N_{\text{iter}} - 1.$$

We emphasize that these solutions apply optimization methods to a substitute function, and not the loss function itself. The smoothing factor σ and realism factor α can be varied, and the substitute loss function can be made as close as desired to the initial one.

In the above, we assumed given a stepsize schedule $\{h_k\}_{k=0}^{N_{\text{iter}}-1}$, and an initial guess $t^{(0)}$. The initial guess is determined based on a heuristic, and the minimization is executed several times, for different starting points.

Chapter 6

Results and discussion

In this chapter, we evaluate the three methods described in Chapter 5 against simple representations of environments. These test environments are not intended to model real-world scenarios with high precision. Instead, compromises were made in favor of the simplicity of the tools used. The primary objective is to assess whether the problem can be effectively addressed using the proposed formulation and methods, and to characterize the success rate and the influence of different parameters.

We begin by briefly presenting the tools and test environments used. Next, we perform a test in free space, to test convergence of the methods and fix the stepsize. The results and corresponding observations are finally presented and discussed.

6.1 Simulation tool and environments

Ray tracing tool

In this work, the virtual representation of the environments and the Ray Tracing (RT) methods are implemented using the `DiffeRT`[1] module. This tool is written in Python and leverages the `JAX` library[2], which provides, among other features, Just-In-Time (JIT) compilation, efficient vectorized operations, and an Automatic Differentiation (AD) framework. The `DiffeRT` module allows for both forward and backward modes for differentiation. Since our problem typically presents a scalar loss function and requires the partial derivatives with respect to each of the 2D-coordinates of the transmitter $t \in \mathbb{R}^2$, both forward and reverse modes can be justified. The forward mode is used hereafter, for its lower memory requirements.

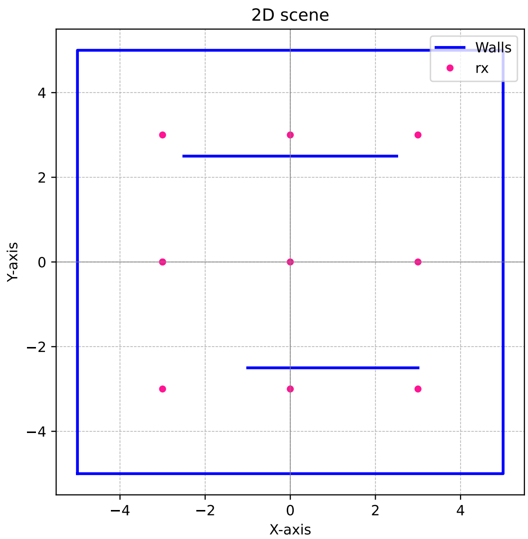
The RT method used in this work (Image method, section 2.5) does not include diffraction modeling but allows for an arbitrarily large number of reflections. However, since higher-order reflections have diminishing influence and increase computation time, the RT simulations of the forward problem solved at each iteration are limited to a maximum of two reflection orders, unless otherwise specified.

[1] https://github.com/jeertmans/DiffeRT

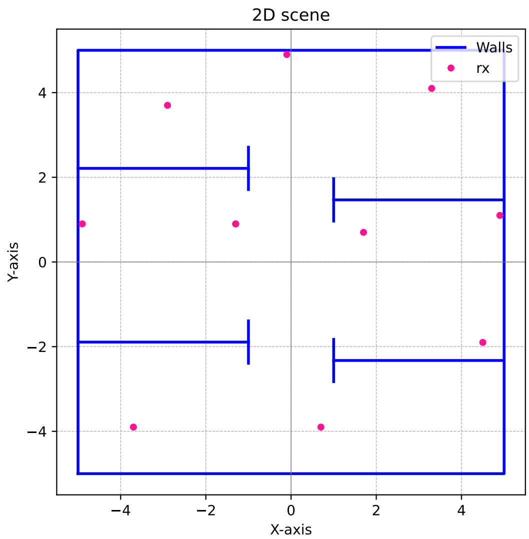
[2] https://docs.jax.dev/en/latest/quickstart.html

Test environments

Although the `DiffeRT` module supports three-dimensional modeling, we restrict this study to two-dimensional environments to simplify the setup and reduce computational cost. Two types of scenes are considered. First, we define a benchmark scene with 6 walls and fixed receivers' positions, as shown on Figure 6.1a. Second, we consider a small 2D indoor map, where the receivers' positions are randomly generated. An example is shown on Figure 6.1b. Receivers positions are randomly sampled under the constraint that they remain sufficiently far apart, based on a predefined rejection criterion and minimum distance. Both scenes are defined within a 10-meter by 10-meter square domain.



(a) Map of the simple benchmark environment.



(b) Map of the rooms environment.

Figure 6.1: 2D maps of the environments considered for the tests.

Tests and parameters

We divide the tests into two main categories, to distinguish as much as possible between 1) **algorithmic limitations** and 2) **intrinsic problem ambiguities**. Namely:

1. The first set of tests focuses on evaluating the **optimization methods** presented in Chapter 5. The objective is to assess their ability to locate the global minimum of the loss function derived from the RT model, as formulated in Chapter 4. In this part, the power measurements $\tilde{b} := b$ are noise-free: generated with the forward model **(P)** and the same parameters as is used throughout the iterations of the methods (i.e., a maximal number of reflections of two). These experiments allow to compare the performance of the proposed methods and to analyze the influence of hyperparameters, such as the number of Monte Carlo samples and the smoothing parameters σ and α.

   Having methods working on this **ideal framework** is a necessary (but not sufficient) condition to hope to solve the inverse problem.

2. In a second time, we assess the performance on the **inverse problem resolution**. Indeed, in practice, the measurements $\tilde{b}$ come from experiments and presents additional modeling errors and measurements noise. In this work, we neglect measurements noise, but try to evaluate the impact of the modeling noise coming from the truncation of the model to a maximal number of two reflections during the resolution. The ground-truth measurements $\tilde{b}$ are generated with a maximum number of reflection of five, in this second part.

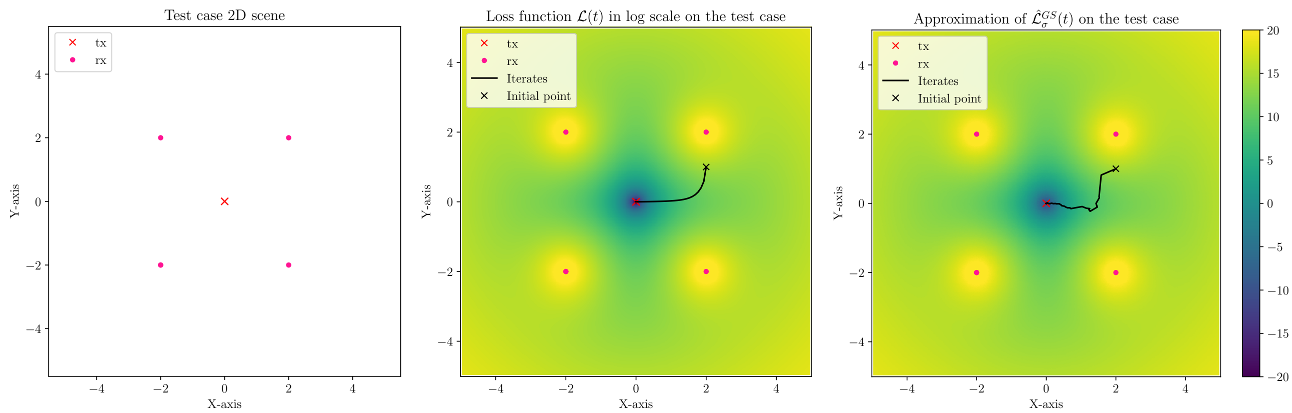
   In this part, the goal is to have an insight of the impact of **modeling noise** on the resolution.

In what follows, we start from initial parameters for each of the methods, assess their performance and vary the parameters one by one in the next sections.

Remark: The results, the sample means and standard deviations accross experiments below were computed on 100 random experiments. This number was limited by computational time — the RT simulations being particularly costly on large scenes or with a large number of receivers. Moreover, the number of initial points (multistart) has been fixed to $N_{\text{init}} = 5$, and the number of iterations to 100.

6.2 Parameter tuning and test in free space

We first consider an empty scene, with no obstacles and 4 receivers (Figure 6.2). In this case, only LOS paths exist, and the loss function is continuous (section 2.4). It constitutes a test case to check the implementation of the methods and fix the stepsize $\{h_k\}_{k\in\mathbb{N}}$ (also called learning rate, lr).



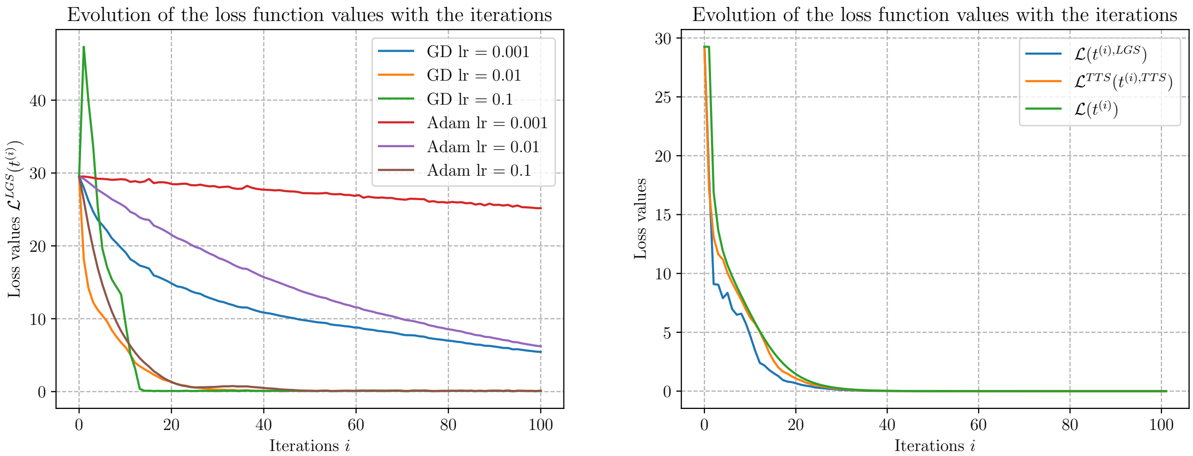
(a) Map of the test case scene with 4 receivers.

(b) Heatmap of the loss function $\mathcal{L}(t)$ in the scene.

(c) Heatmap of the approximation of $\mathcal{L}^{\text{LGS}}(t)$ in the scene.

Figure 6.2: Test in free space. (a) shows the test environment, (b) the loss function and iterates of the GM, (c) the approximation of the LGS loss function, for $N_\sigma = 100$ and $\sigma = 0.01$.

Figure 6.3a shows the evolution of the loss function values with the iterations for different choice of stepsizes: constants or computed through the Adam method [59]. This example encourages a choice of stepsize of 0.01 if constant (constant stepsize Gradient Method (GM)), or 0.1 in the Adam method. Of course, testing on this simple continuous function only gives an indication based on typical function values, but this choice was verified on discontinuous cases of the next section. With this choice of stepsize, all methods converge similarly to the minimizer, as shown on Figure 6.3b.



(a) Comparison of the choice of stepsize in the LGS method.

(b) Comparison of the different methods on the test case.

Figure 6.3: 2D maps of the environments considered for the tests.

6.3 Optimization of the piecewise continuous loss function

In this section, we assess the performance of the optimization methods and the impact of hyperparameters. We know that the loss function value is 0 at the minimizer, since the ground-truth measurements were generated with the same model as the candidate measurements throughout the iterations. These are tests in an idealized situation.

6.3.1 Number of receivers

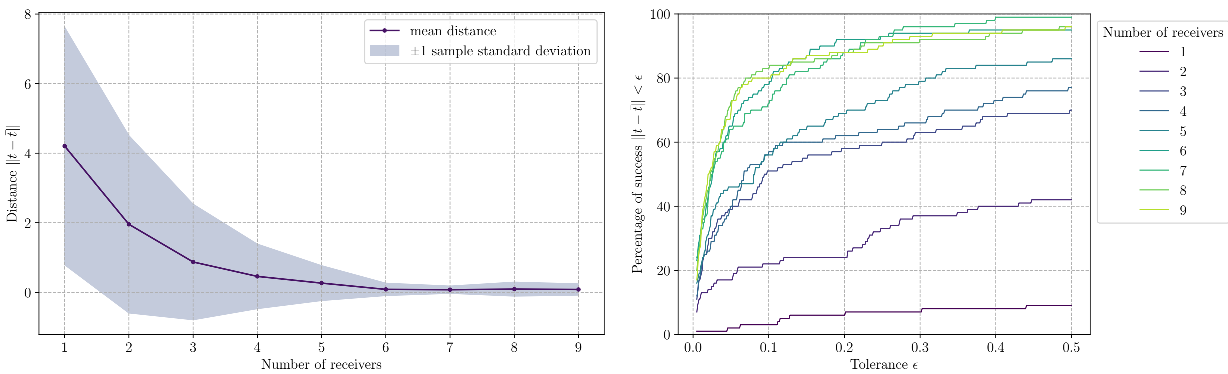
A first step is to identify the number of receivers necessary for the problem to be tractable. Indeed, with few receivers, there might be many positions in the scene yielding the same power measurements at the receivers, and the loss function $\mathcal{L}$ could have multiple local or even global minima. We expect the optimization to be unsuccessful most of the time for few receivers, and increasingly successful for increasing number of receivers. We note that, in practice, increasing the number of receivers not only should be avoided for material reasons, but also because it increases proportionally the cost of the optimization method.

The mean and standard deviation of the distance of the best candidate to the true minimizer are shown on Figure 6.4a, Figure 6.5a and Figure 6.6a for the LGS, TTS and PBS methods respectively. As expected, the mean distance for all three methods decreases with increasing number of receivers until six, from which it stagnates to a non-zero value. This can be explained first because the optimization is limited to a maximum of 100 iterations, then because it is limited in precision by the smoothing factor and the error on the gradient.

The failure of the method for less than five receivers could be explained by the presence of numerous local non-global minima, or non-uniqueness of the solution to the inverse problem.

The *success* is evaluated based on the distance between the candidate and the minimizer. For different tolerances ϵ, an experiment is considered a success if the final candidate t retained after the optimization process satisfies $\|t - \bar{t}\| < \epsilon$. The success rates as functions of the tolerance ϵ for the LGS, TTS and PBS methods are shown on Figure 6.4b, Figure 6.5b and Figure 6.6b respectively.

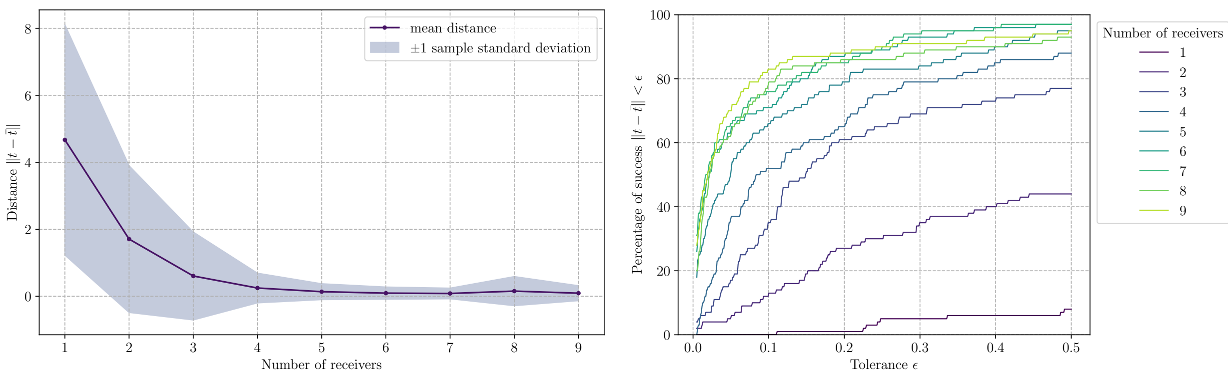
We observe that the first two methods perform similarly, reaching about 80% of success for 7, 8 or 9 receivers and a tolerance of 0.1 [m]. The third method shows a respective success rate of 60%. Having identified 7 as a sufficient number of receivers for this scene, we proceed in the next section to compare the evolution of the distance to the minimizer over the iterations, for the three methods.



(a) Sample mean and standard deviation of the distance of the candidate to the minimum.

(b) Success rate of the LGS method on the benchmark scene.

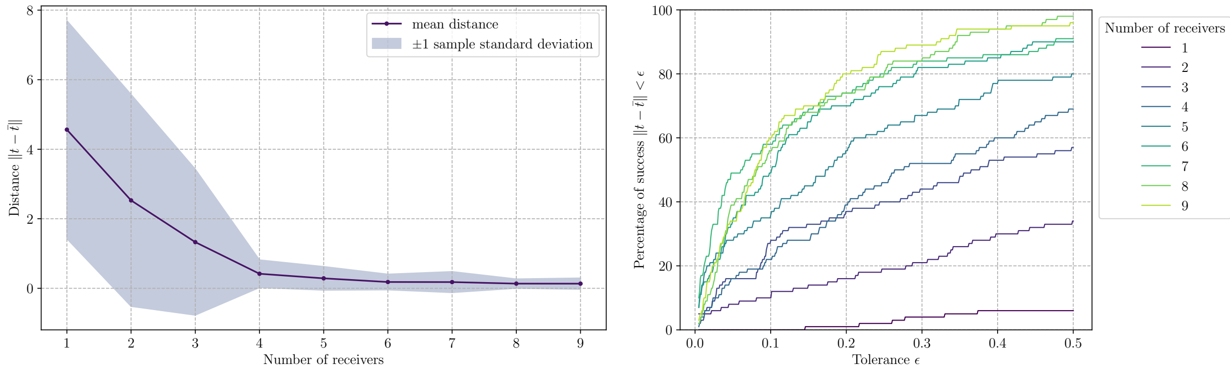
Figure 6.4: Evolution of the distance to the minimizer and of the success rate of the LGS method ($N_\sigma = 100$, $\sigma = 0.01$) with the number of receivers in the benchmark scene.



(a) Sample mean and standard deviation of the distance of the candidate to the minimum.

(b) Success rate of the TTS method on the benchmark scene.

Figure 6.5: Evolution of the distance to the minimizer and of the success rate of the TTS method ($N_\sigma = 100$, $\sigma = 0.01$) with the number of receivers in the benchmark scene.



(a) Sample mean and standard deviation of the distance of the candidate to the minimum.

(b) Success rate of the PBS method on the benchmark scene.

Figure 6.6: Evolution of the distance to the minimizer and of the success rate of the PBS method with the number of receivers in the benchmark scene.

6.3.2 Number of iterations and methods comparison

We visualize on Figure 6.7 the evolution of the distance to the minimizer throughout the iterations, for seven receivers. In order to compare convergence of the different methods, we restrict to the successful experiments, in all three cases. Both LGS and TTS methods behave similarly on this example, while the PBS method is a bit slower to converge.

We also deduce, from the large standard deviation bands, that the convergence is clearly not monotonic, and the method sometimes takes bad steps. The performance may depend heavily on the position of the transmitter, the scene, and the positions of the receivers.

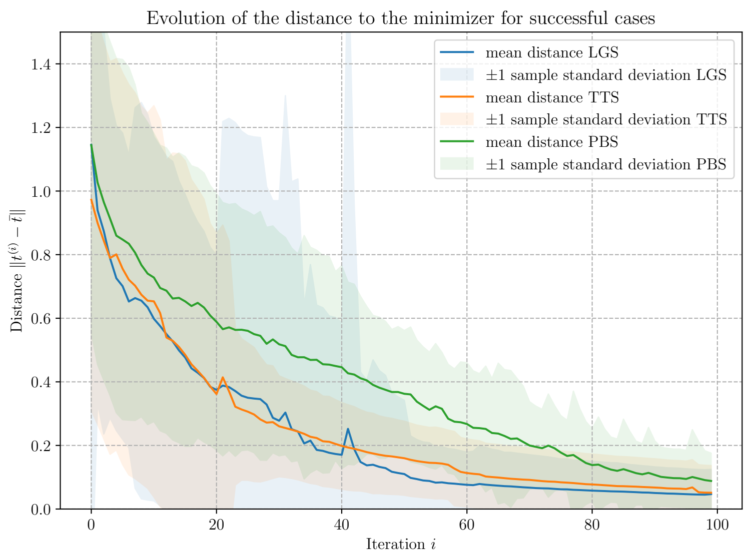
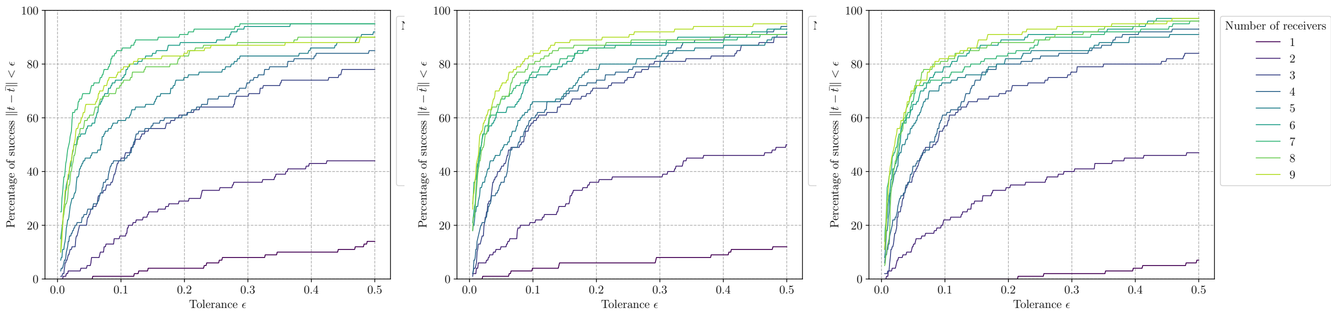


Figure 6.7: Evolution of the distance of the iterates to the minimizer for the three methods studied for successful cases, 100 random experiments on the benchmark scene with 7 receivers.

6.3.3 Number of Monte Carlo samples

Decreasing the number of MC samples in the computation of the gradient implies less accurate estimates. However, maybe this precision is not necessary and more classical DFO techniques with finite differences perform just as good for much lower computational cost. It is indeed the case. Figure 6.8a, Figure 6.8b, and Figure 6.8c show the success rate of the LGS method with different numbers of samples. Performances are sensibly similar, for a same maximum number of iterations — therefore, saving a lot of computations when lowering N_σ.

Figure 6.9 compares the mean distances for the three above, together with the case of 100 samples. Again, performances are similar. Similar results are observed for the TTS method and are shown on Figure 6.10. These graphs suggest that it is reasonable to use few samples or forward or centered finite difference schemes for the estimate of the gradient in the minimization.



(a) Success rate of the LGS method with $N_\sigma = 10$.

(b) Success rate of the LGS method with $N_\sigma = 2$ (centered finite difference scheme).

(c) Success rate of the LGS method with $N_\sigma = 2$ (forward finite difference scheme).

Figure 6.8: Success rate of the LGS method with the number of receivers in the benchmark scene for different numbers of MC samples in the computation of the gradient.

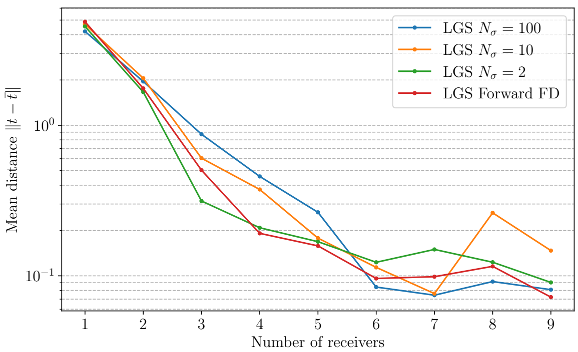


Figure 6.9: Evolution of the sample mean of the distance to the minimum with the number of receivers in the scene for the LGS method and different number of Monte Carlo samples N_σ, in logarithmic scale.

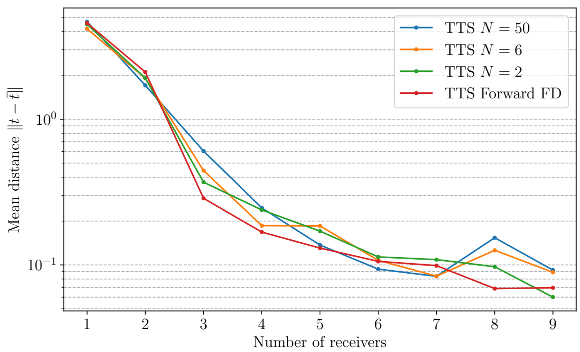


Figure 6.10: Evolution of the sample mean of the distance to the minimum with the number of receivers in the scene for the TTS method and different number of Monte Carlo samples N, in logarithmic scale.

6.3.4 Smoothing factor

The smoothing parameter plays an important role in both the derivative-based and derivative-free (DFO) approaches. For the latter, its interpretation as detailed in Chapter 5 provides some insights.

A range of values for the smoothing factor σ (in LGS and TTS) and for the realism factor α (in PBS) has been tested. Larger values of σ correspond to a heavier smoothing of the loss function (seen as a convolution with a broader Gaussian kernel), which helps soften the discontinuities and mitigate local minima. However, excessive smoothing results in a loss landscape that deviates significantly from the original objective, compromising accuracy. Conversely, smaller values yield functions closer to the true objective but may hinder convergence due to its piecewise continuity.

The mean distances between the best iterate and the true transmitter position for various numbers of receivers are shown on Figure 6.11, Figure 6.12 and Figure 6.13 for the LGS, TTS and PBS methods respectively. These graphs illustrate the trade-off detailed above.

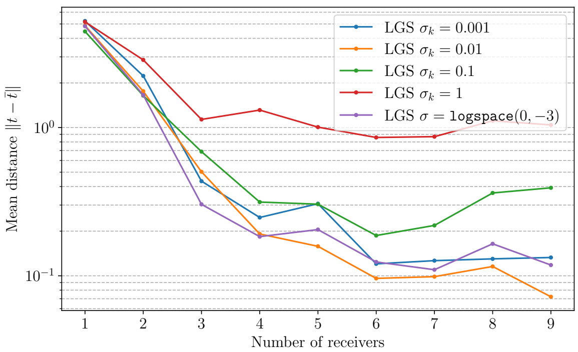


Figure 6.11: Evolution of the sample mean of the distance to the minimum with the number of receivers in the scene for the LGS method with forward finite differences and different values of the smoothing factor σ.

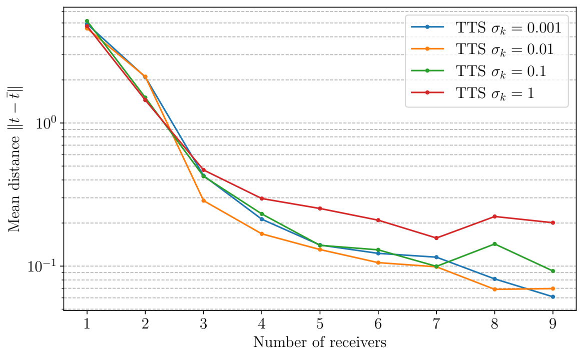


Figure 6.12: Evolution of the sample mean of the distance to the minimum with the number of receivers in the scene for the TTS method with forward finite differences and different values of the smoothing factor σ.

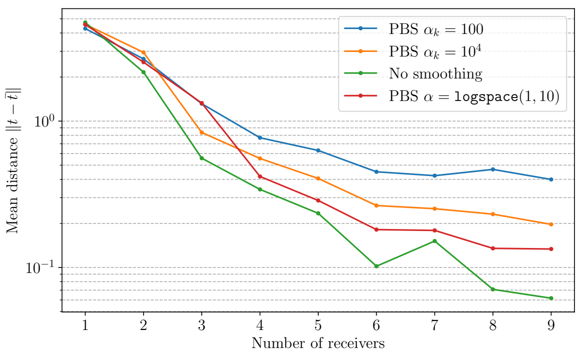


Figure 6.13: Evolution of the sample mean of the distance to the minimum with the number of receivers in the scene for the PBS method and different values of the realism parameter α.

Intuitively, we could propose to employ a continuation strategy, wherein the smoothing factor is gradually decreased throughout the optimization process. It did not seem to improve performances significantly on this example however.

Since the LGS and TTS methods are DFO methods, we can hardly compare them at equal costs with the PBS and the classic GM methods. However, Figure 6.13 compares the PBS and the classic GM methods, and suggests that, on average, the PBS method does not help finding the minimizer of the loss function.

6.3.5 Stochastic approach

As illustrated above, a minimum number of receivers located at distinct points in the scene is required for the solution to this inverse problem to be unique and for the different approaches to perform reasonably well. We physically need enough information points. That does not imply however, that all information — namely, all power measurements simulated through the forward model — should be used (and therefore costly computed) at each iteration. Simulating the power measurements of only a subset of the receivers at each iteration, and comparing them with the corresponding subset of true measurements ($\tilde{b}_i$ for the corresponding indices i) could allow saving a lot of computation.

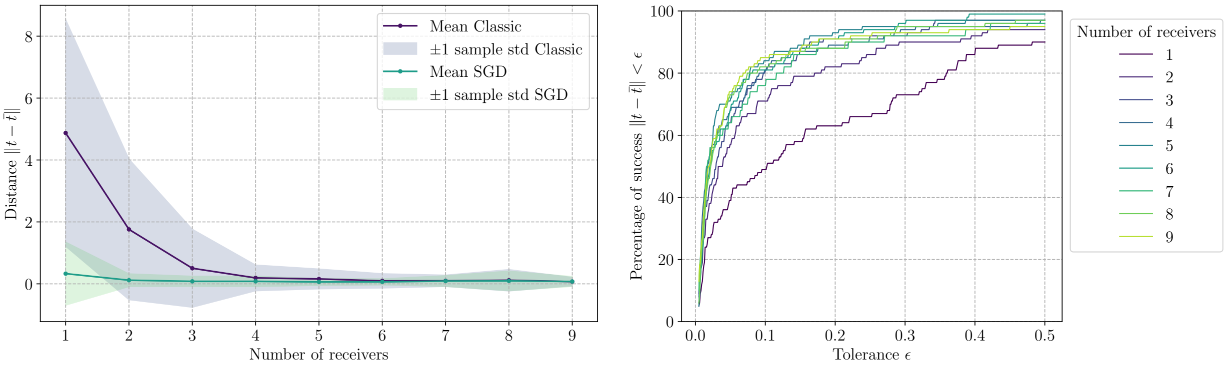
This is exactly the idea of the stochastic gradient descent [60]. This technique is often applied on objective functions written as an average of simpler functions. In our case, we replace the loss function (4.1) by

$$\mathcal{L}_{\mathrm{SGD}_{\mathcal{K}}}(t, \bar{r}, \mathcal{S}, \tilde{b}) = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \left(\Phi(t, r_i) - \tilde{b}_i \right)^2 ,$$

where the set $\mathcal{K}$ is the subset of indices of the chosen receivers, and $|\mathcal{K}|$ denotes its cardinality. These chosen receivers are picked randomly at each iteration.

Similarly to what was done for the classical loss function, the LGS, TTS and PBS continuous substitutes can be defined and the same methods applied. Figure 6.14 illustrates the performance of this stochastic approach, compared to the classical one, on the LGS method with forward finite differences. The classic and SGD methods compared on this graph share the same computational cost per iteration, but the SGD method requires more receivers placed in the scene. In this case, we assume 9 physical receivers in the scene, and the set $\mathcal{K}$ is chosen randomly at each iteration as a subset of the set $\{1, 2, \ldots, 9\}$.

We observe in Figure 6.14b that similar performance is achieved for $|\mathcal{K}| \geq 3$. The performances of the method only deteriorate for $|\mathcal{K}| < 3$. This suggests that a stochastic gradient descent (SGD) approach could be employed to reduce the computational cost per iteration, without compromising performance. It is the number of physical receivers in the scene that matters more. Similar results are expected for the TTS and PBS methods.



(a) Sample mean and standard deviation of the distance of the candidate to the minimum.

(b) Success rate of the LGS method for different number of receivers considered ($|\mathcal{K}|$).

Figure 6.14: Evolution of the distance to the minimizer and of the success rate of the LGS method ($N_\sigma = 2$, forward finite differences $\sigma = 0.01$) with the number of receivers in the set $\mathcal{K}$ considered at each iteration in the benchmark scene. In this case, we assume 9 physical receivers in the scene, and the set $\mathcal{K}$ is chosen randomly at each iteration as a subset of the set $\{1, 2, \dots, 9\}$. On Figure (a), the mean distance evolution is compared with the classic method. The classic and SGD methods on this graph share the same computation cost per iteration, but the SGD method requires more receivers placed in the scene.

6.4 Resolution of the inverse problem

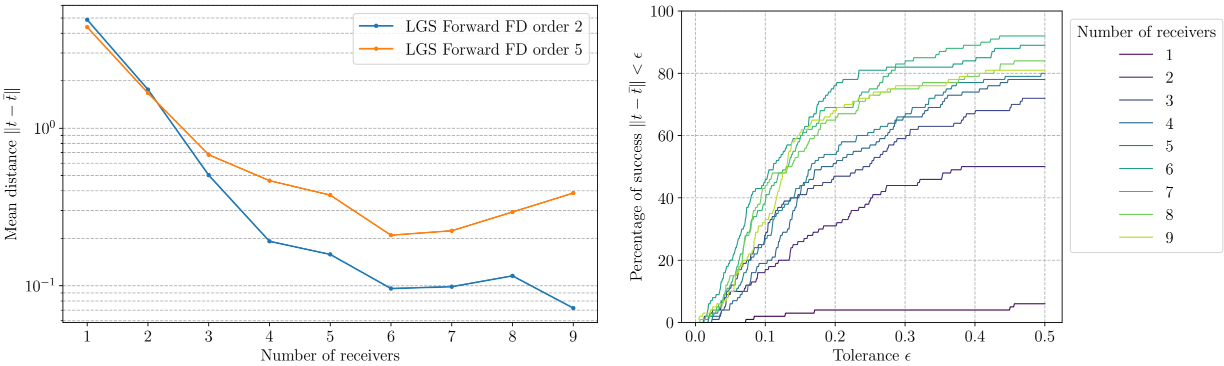
In the previous section, we examined the influence of various parameters under idealized conditions, with neither modeling nor measurement noise. In this section, we evaluate the robustness of the proposed approach by introducing a source of modeling error. Specifically, the *ground-truth* measurements are generated using a model that includes up to five reflections, while the forward model used during optimization is limited to a lower maximum number of reflections. As a result, the loss function is no longer expected to reach zero at the true transmitter location, and its minimizer may not coincide with this true location.

6.4.1 Number of receivers

Figure 6.15a illustrates the evolution of the mean localization error for both the noise-free and modeling noise scenarios on the benchmark scene. It highlights the impact of modeling noise on the accuracy of the methods: the mean error does not reach 0.1 [m] anymore, but only around 0.5 [m] for a sufficient number of receivers. Figure 6.15b shows the associated success rate evolution with the tolerance ϵ.

6.4.2 Complexity of the scene

Finally, we evaluate the method on a more realistic environment, depicted in Figure 6.1b. Figure 6.16a presents the evolution of the sample mean distance between the best candidate and the true transmitter position. Due to the increased number of walls of this scene, a higher number of receivers was required. While the method performs reasonably well in the ideal setting (i.e., without modeling noise), performance degrades significantly when modeling noise is introduced. Interestingly, the relatively good results in the ideal case were not guaranteed, as more complex environments tend to induce a more fragmented and challenging

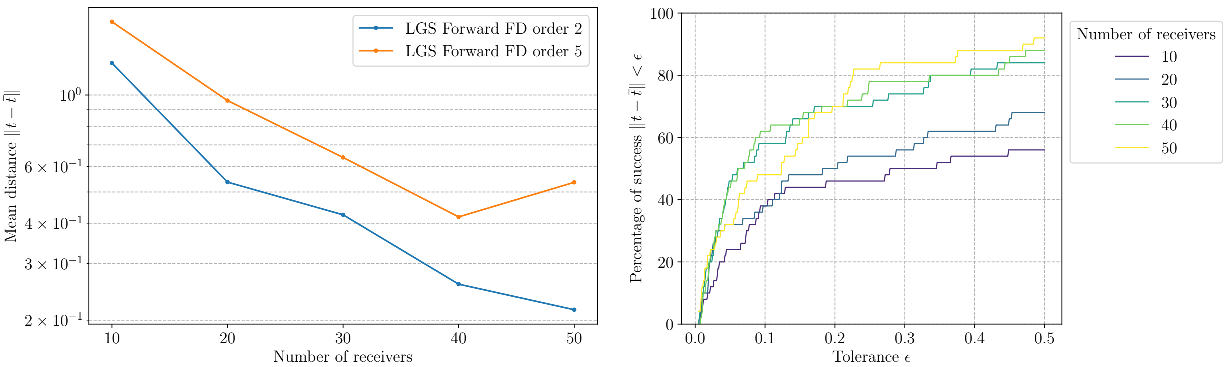


(a) Evolution of the sample mean of the distance with the number of receivers, in logarithmic scale. Comparison between noise-free b (order 2) and noisy measurements $\tilde{b}$ (order 5).

(b) Success rate of the LGS method on the benchmark scene, for several numbers of receivers, in case of noisy measurements $\tilde{b}$ (modeling noise).

Figure 6.15: Evolution of the distance to the minimizer and of the success rate of the LGS method ($N_\sigma = 2$, forward finite differences, $\sigma = 0.01$) with the number of receivers, in the benchmark scene.

loss landscape. Although further parameter tuning might marginally improve performance, the results in the non-ideal case highlight a fundamental limitation of the approach: the level of modeling noise is a critical point.



(a) Evolution of the sample mean of the distance, in logarithmic scale. Comparison noise-free b (order 2) and noisy measurements $\tilde{b}$ (order 5).

(b) Success rate of the LGS method on the rooms scene, for noise-free measurements b (ideal case).

Figure 6.16: Evolution of the distance of the best candidate to the true position for the LGS method ($N_\sigma = 2$, forward finite differences, $\sigma = 0.01$), on the rooms scene (6.1b).

Chapter 7

Conclusion and perspectives

This final section begins by outlining related research directions, followed by a summary of the proposed approach and key observations. It concludes with suggestions for potential improvements and a critical discussion of the applicability and limitations of the method for solving the original wireless indoor source localization problem addressed in this Master's thesis.

7.1 Related inverse problem

As discussed in section 2.3, while the geometric layout of the environment is typically known or can be measured, the electromagnetic properties — particularly the reflection coefficients of the walls — often require estimation. Accurate values of these coefficients are essential for the reliability of the ray tracing (RT) model. One promising approach to estimating them is to formulate the problem as an inverse problem, using the same forward model **(P)** introduced in this work.

Indeed, we recall that the RT operator $\mathcal{A}(t, \bar{r}, \mathcal{S})$ depends not only on the transmitter position t, but also on scene parameters $\mathcal{S}$, which include both geometric and electromagnetic characteristics. Importantly, the operator is differentiable with respect to the reflection coefficients of the walls, which allows the use of gradient-based optimization techniques to estimate them.

With this formulation, the loss function — defined similarly to (4.1) — is continuous and differentiable with respect to the reflection coefficients, and nonconvex. Its structure is a polynomial of degree Z, where Z denotes the maximum number of reflections considered in the model, and $C_p^{(w)}$ the reflection coefficient of a given wall w. This polynomial has positive coefficients for all monomials of positive degree. Furthermore, the reflection coefficient $C_p^{(w)}$ is known to lie within the interval $[0, 1]$, and in many practical cases, a reasonable initial estimate is available. In simplified cases — when a single unknown coefficient is sought, or when all walls share a common coefficient — the loss function has a unique global minimum within the feasible domain. However, in more general scenarios involving multiple unknown reflection coefficients, the solution may not be unique. As with the source localization problem, the number and placement of receivers is then critical.

This inverse problem is therefore significantly simpler than the transmitter localization problem. It relies on the same forward model and RT simulations but benefits from a nicer optimization landscape. Moreover, the method is particularly well-suited for fine-tuning: starting from an approximate initial guess, first-order optimization methods can more reliably converge to an accurate solution.

7.2 Summary

The primary objective of this Master's thesis was to propose and assess a new approach to wireless indoor source localization, as an inverse problem. After having detailed a simplified model for the wave propagation between the transmitter and the receivers, we formulated the forward problem, that is solved through RT simulations. Based on this forward problem, we established a formulation of the wireless indoor source localization as an inverse problem, and listed the particularities and challenges at hand. Encouraged by the discontinuous nature of the resulting loss function, we looked into different smoothing strategies. We explained how the first two relate to derivative-free optimization methods. The third approach modifies the physical model itself, yielding a continuous substitute loss function amenable to gradient-based optimization. This last method requires the RT simulation to support automatic differentiation, whereas the first two approaches can be applied to any ray tracer.

By testing different parameter values, we observed that the derivative-free optimization methods seemed to give better results. We also highlighted that their cost per iteration can be largely reduced by considering forward finite differences for the gradient estimation, and only a stochastic subset of receivers at each iteration. While results in idealized simulation settings are encouraging, we emphasize that significant challenges remain for real-world applications, where modeling inaccuracies and environmental complexities limit the effectiveness of this formulation.

7.3 Conclusion

Possible improvements

While this work focused on first-order and a specific class of derivative-free optimization methods, several other methods could have been explored:

- Second-order optimization techniques, such as Newton or quasi-Newton methods could have been envisaged;
- Further investigation into other zero-order methods, such as trust-region methods is another option;
- Finally, a hybrid approach could have been applied, coupling the proposed optimization strategy with a more classical *scene-analysis* technique (Chapter 3). The idea would be to start with a first guess, chosen among a rather coarse predefined grid, through dictionary-based methods. Then, this first guess could be used as an initial point for fine-tuning via local optimization. It may be promising, particularly for scenarios involving multiple transmitters. If the measurements vectors corresponding to the predefined grid of transmitters are sufficiently different, it could provide good starting points, reducing the risk of convergence to poor local minima. It comes at the cost

of, first, simulating the received powers for all points on this predefined transmitter position grid.

However, it is important to emphasize that the main challenges encountered in this work are likely to persist regardless of the chosen optimization scheme.

On the model and modeling errors

The RT model used in this work is a simplified version of more complete ray tracing frameworks: we considered only specular reflections, limited to a maximum order. Applying this method to real-world measurements would require more accurate modeling — including effects such as higher order reflections, diffraction, transmission through walls for example — as we have witnessed the significant impact of even moderate modeling errors.

The first two methods, with Gaussian smoothing, only need zero-order oracle of the RT simulation and should therefore be easy to translate to another more precise ray tracer. It would be more difficult to implement the physics-based smoothing in such ray tracers.

On the justification of this inverse problem formulation

Even with a more accurate RT model, certain non-idealities persist. Importantly, the discontinuities in the loss function stem from the RT formulation itself, not from our further simplification. Thus, the analysis and unfortunately the challenges encountered remain general. Second, the number of receivers required to guarantee a unique solution is unknown, likely necessitating an impractically large deployment in the scene, for real-world applications.

To conclude, the increasing number of RT tools and increasing accuracy of the RT models makes it tempting to formulate the problem of wireless indoor source localization in the inverse form proposed in this thesis. Furthermore, the recent development of differentiable ray tracers fosters the use of first-order optimization methods. In solving any mathematical problem however, the main challenge often lies in identifying a suitable model, allowing the use of the right resolution methods. In view of the challenges encountered with the proposed model, we believe that this approach is not ideal for the problem as presented. Maybe another model, method and point of view would have allowed getting more theoretical guarantees.

Inspired by the iterative nature of the method, one possible application for our approach lies in robotics scenarios, as mentioned in Chapter 3. If the transmitter moves within the environment, the optimization framework could be used for tracking its position. Initializing each execution of the optimization method with the previously estimated location, the transmitter position can be updated based on new measurements at the current timestep. In such cases — where a good initialization is available — optimization is likely to be more effective.

Ultimately, the approach developed in this thesis should be best interpreted as a fine-tuning mechanism, rather than a complete standalone solution.

Bibliography

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37:6, 2007.

[2] C. G. Manning. Global Positioning System (GPS). In: *NASA Space Communications and Navigation Program*. 2023.

[3] E. D. Kaplan and C. Hegarty. Understanding GPS/GNSS: Principles and Applications. Third edition. Artech House, 2017.

[4] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, and T. Hart. COIN-GPS: indoor localization from direct GPS receiving. In: *Proceedings of the 12th annual international conference on Mobile systems, applications and services*, 2014.

[5] T. S. Rappaport. Wireless communications: principles and practice. Second edition. Prentice Hall communications engineering and emerging technologies series. Prentice Hall, 2009.

[6] J. Tiemann, F. Schweikowski, and C. Wietfeld. Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments. In: *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015.

[7] Q. Vey, F. Spies, B. Pestourie, D. Genon-Catalot, A. Van Den Bossche, T. Val, R. Dalce, and J. Schrive. POUCET: A Multi-Technology Indoor Positioning Solution for Firefighters and Soldiers. en. In: *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2021.

[8] D. Kumar and N. Muhammad. A Survey on Localization for Autonomous Vehicles. In: *IEEE Access*, 2023.

[9] *Introduction to Sionna RT — Sionna 0.19.2 documentation (2025-03-09)*. URL: https://nvlabs.github.io/sionna/examples/Sionna_Ray_Tracing_Introduction.html.

[10] T. K. Geok, F. Hossain, and A. T. W. Chiat. A novel 3D ray launching technique for radio propagation prediction in indoor environments. In: *PLOS ONE* 13:8, 2018.

[11] R. Zekavat and R. M. Buehrer. *Handbook of Position Location: Theory, Practice, and Advances*. John Wiley & Sons, 2019.

[12] J. Khodjaev, Y. Park, and A. Saeed Malik. Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments. In: *Annals of telecommunications* 65:5, 2010.

[13] S. Marano, W. Gifford, H. Wymeersch, and M. Win. NLOS Identification and Mitigation for Localization Based on UWB Experimental Data. In: *Selected Areas in Communications, IEEE Journal on* 28, 2010.

[14] J. Eertmans, L. Jacques, and C. Oestges. Fully Differentiable Ray Tracing via Discontinuity Smoothing for Radio Network Optimization. In: *18th European Conference on Antennas and Propagation (EuCAP)*, 2024.

[15] *DiffeRT2d documentation (2025-03-10)*. URL: https://eertmans.be/DiffeRT2d/index.html.

[16] G. Loubet, N. Holzschuch, and W. Jakob. Reparameterizing discontinuous integrands for differentiable rendering. en. In: *ACM Transactions on Graphics* 38:6, 2019.

[17] M. Nimier-David, Z. Dong, W. Jakob, and A. Kaplanyan. Material and lighting reconstruction for complex indoor scenes with texture-space differentiable rendering. In: 2021.

[18] *10 Lectures on Inverse Problems and Imaging (2025-03-11)*. URL: https://tristanvanleeuwen.github.io/IP_and_Im_Lectures/what_is.html.

[19] X. Han, T. Zheng, T. X. Han, and J. Luo. RayLoc: Wireless Indoor Localization via Fully Differentiable Ray-tracing. In: 2025. arXiv:2501.17881.
[20] C. Wu, Z. Yang, and Y. Liu. Wireless Indoor Localization: A Crowdsourcing Approach. Springer, 2018.
[21] M. Kline. Electromagnetic theory and geometrical optics. In.
[22] D. A. McNamara, C. W. I. Pistorius, and J. A. G. Malherbe. Introduction to the Uniform Geometrical Theory of Diffraction. Artech House, 1990.
[23] J. B. Keller. Geometrical Theory of Diffraction. In: *JOSA* 52:2, 1962.
[24] V. Degli-Esposti. General Theory of Radio Propagation. *ESoA course on Mobile Radio Propagation for 5G and beyond*. 2022.
[25] M. Wang, Y. Wang, W. Li, J. Ding, C. Bian, X. Wang, C. Wang, C. Li, Z. Zhong, and J. Yu. Reflection Characteristics Measurements of Indoor Wireless Link in D-Band. In: *Sensors* 22:18, 2022. Number: 18.
[26] F. Quatresooz. Dynamic ray tracing techniques for mobile radio communications. In: *Master's Thesis*, 2020.
[27] J. Eertmans, E. M. Vittuci, V. Degli-Esposti, L. Jacques, and C. Oestges. Comparing Differentiable and Dynamic Ray Tracing: Introducing the Multipath Lifetime Map. In: 2024. arXiv:2410.14535.
[28] Z. Yun and M. F. Iskander. Ray Tracing for Radio Propagation Modeling: Principles and Applications. In: *IEEE Access* 3, 2015.
[29] F. Puggelli, G. Carluccio, and M. Albani. A Novel Ray Tracing Algorithm for Scenarios Comprising Pre-Ordered Multiple Planar Reflectors, Straight Wedges, and Vertexes. In: *IEEE Transactions on Antennas and Propagation* 62:8, 2014.
[30] J. Eertmans, C. Oestges, and L. Jacques. Min-Path-Tracing: A Diffraction Aware Alternative to Image Method in Ray Tracing. In: *2023 17th European Conference on Antennas and Propagation (EuCAP)*, 2023.
[31] T. Kürner. GIS Data for Radio Network Planning. *ESoA course on Mobile Radio Propagation for 5G and beyond*. 2022.
[32] *Physically Based Rendering: From Theory to Implementation*. URL: https://www.pbr-book.org/.
[33] B. Nicolet. Improved Gradients for Inverse Rendering. In: *PhD Thesis*, 2025.
[34] P. H. W. Hoffmann. A Hitchhiker's Guide to Automatic Differentiation. In: *Numerical Algorithms* 72:3, 2016.
[35] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. *Automatic differentiation in machine learning: a survey*. 2018.
[36] B. Viel and M. Asplund. Why is fingerprint-based indoor localization still so hard? In: *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM workshops)*, 2014.
[37] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed. A Review of Indoor Localization Techniques and Wireless Technologies. In: *Wireless Personal Communications* 119:1, 2021.
[38] K. Cheung, H. So, W.-K. Ma, and Y. Chan. Least Squares Algorithms for Time-of-Arrival-Based Mobile Location. In: *IEEE Transactions on Signal Processing* 52:4, 2004.
[39] X.-m. Yu, H.-q. Wang, and J.-q. Wu. A method of fingerprint indoor localization based on received signal strength difference by using compressive sensing. In: *EURASIP Journal on Wireless Communications and Networking* 2020:1, 2020.
[40] A. Tayebi, J. Gomez, F. Saez De Adana, and O. Gutierrez. The Application of Ray-Tracing to Mobile Localization Using the Direction of Arrival and Received Signal Strength in Multipath Indoor Environments. In: *Progress In Electromagnetics Research* 91, 2009.
[41] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. Mitsuba 2: a retargetable forward and inverse renderer. In: *ACM Trans. Graph.* 38:6, 2019.
[42] B. d. Koning, A. Heemels, A. Adam, and M. Möller. Gradient descent-based freeform optics design for illumination using algorithmic differentiable non-sequential ray tracing. In: *Optimization and Engineering* 25:3, 2024.
[43] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable Monte Carlo ray tracing through edge

sampling. In: *ACM Transactions on Graphics* 37:6, 2018.

[44] B. Nicolet, A. Jacobson, and W. Jakob. Large steps in inverse rendering of geometry. In: *ACM Transactions on Graphics* 40:6, 2021.

[45] M. M. Loper and M. J. Black. OpenDR: An Approximate Differentiable Renderer. In: *Computer Vision – ECCV 2014* 8695, 2014. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Series Title: Lecture Notes in Computer Science.

[46] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic Differentiation in Machine Learning: a Survey.

[47] Z. Zhang, N. Roussel, and W. Jakob. Projective Sampling for Differentiable Rendering of Geometry. In: *ACM Transactions on Graphics* 42:6, 2023.

[48] *Gradient-based optimization - Mitsuba 3*. URL: https://mitsuba.readthedocs.io/en/stable/src/inverse_rendering/gradient_based_opt.html.

[49] B. Nicolet, F. Rousselle, J. Novak, A. Keller, W. Jakob, and T. Müller. Recursive Control Variates for Inverse Rendering. In: *ACM Transactions on Graphics* 42:4, 2023.

[50] M. Leordeanu and M. Hebert. Smoothing-based Optimization. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[51] H. Mobahi and J. Fisher Iii. A Theoretical Analysis of Optimization by Gaussian Continuation. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 29:1, 2015.

[52] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. New York, NY: Springer, 2004.

[53] F. Nobile. EPFL MATH-414 Stochastic Simulation: Lecture notes. 2023.

[54] F. Petersen, A. H. Bermano, O. Deussen, and D. Cohen-Or. *Pix2Vex: Image-to-Geometry Reconstruction using a Smooth Differentiable Renderer*. 2019.

[55] T. Möller and B. Trumbore. Fast, Minimum Storage Ray-Triangle Intersection. In: *Journal of Graphics Tools* 2:1, 1997.

[56] Y. Nesterov and V. Spokoiny. Random Gradient-Free Minimization of Convex Functions — Foundations of Computational Mathematics.

[57] P. Boris. *Introduction to Optimization*. Optimization Software, Inc. 1987.

[58] J. Larson, M. Menickelly, and S. M. Wild. Derivative-free optimization methods. In: *Acta Numerica* 28, 2019.

[59] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017.

[60] D. P. Bertsekas. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey. In: *Computer Science - Data Structures and Algorithms, Computer Science - Systems and Control, Mathematics - Numerical Analysis, Mathematics - Optimization and Control*, 2017.

Appendix

A Implementation details

The implementation of the methods and experiments with the `DiffeRT` module are available at https://github.com/SophieL1/TFE2025-WISL and https://github.com/SophieL1/DiffeRT.

B Algorithms

B.1 Pseudocode for LGS approach (section 5.1.1)

Algorithm 1 LGS optimization

Input: Initial point t_0, number of steps T, schedule for the smoothing factor σ (standard deviation of the Gaussian kernel) $\{\sigma_k\}_{k=1}^{T}$, stepsize schedule $\{h_k\}_{k=1}^{T}$, zero-order oracle of the loss function $\mathcal{L}$, number of Monte Carlo samples N.

Output: Iterates $\{t_k\}_{k=1}^{T}$ and estimates of the GS-loss values $\{\hat{\mathcal{L}}_{\sigma}^{GS}(t_k)\}.{}_{k=1}^{T}$

Initialize $x \leftarrow x_0$
for $k = 1$ to T **do**
    Sample $u_1, \dots, u_N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_k^2 I_2)$
    $\tilde{t}_i = \texttt{project}(t - u_i, \text{room})$ for $i = 1, \dots, N$ ▷ Ensure to evaluate the loss inside the domain
    $\ell_i \leftarrow \mathcal{L}(\tilde{t}_i)$ for $i = 1, \dots, N$ ▷ Evaluate the loss function
    $f \leftarrow \frac{1}{N} \sum_{i=1}^{N} \ell_i$ ▷ Compute the estimate $\hat{\mathcal{L}}_{\sigma_k}^{GS}(t_k)$
    $g \leftarrow \frac{1}{N\sigma_t^2} \sum_{i=1}^{N} -\ell_i u_i$ ▷ Compute the estimate $\widehat{\nabla \mathcal{L}}_{\sigma_k}^{GS}(t_k)$
    $t \leftarrow t - h_k\, g$ ▷ Take the step
    $t \leftarrow \texttt{project}(t, \text{room})$ ▷ Ensure to stay inside the room
    Save $t_k \leftarrow t$, $f_k \leftarrow f$
end for
return $(t_k)_{k=1}^{T}$, $(f_k)_{k=1}^{T}$

B.2 Pseudocode for TTS approach (section 5.1.2)

Algorithm 2 TTS optimization

Input: Initial point t_0, number of steps T, schedule for the smoothing factor σ (standard deviation of the Gaussian kernel) $\{\sigma_k\}_{k=1}^T$, stepsize schedule $\{h_k\}_{k=1}^T$, zero-order oracle of the loss function $\mathcal{L}$, number of Monte Carlo samples N.

Output: Iterates $\{t_k\}_{k=1}^T$ and estimates of the gradients of the TTS loss $\{\widehat{\nabla\mathcal{L}_\sigma}^{TTS}(t_k)\}_{k=1}^T$

Initialize $x \leftarrow x_0$
for $k = 1$ to T **do**
  Sample $u_1, \ldots, u_N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_k^2 I_2)$
  Initialize vectors $\bar{a}, a = \text{zeros}(1, m)$
  for each receiver r_i for $i = 1, \ldots, m$ **do**
    $A_{i,j} \leftarrow \Phi(t - u_j, r_i)$ for $j = 1, \ldots, N$ ▷ Evaluate power at r_i
    $\bar{a}_i \leftarrow \frac{1}{N}\sum_{j=1}^N A_{i,j}$ ▷ Compute the estimate of $\left[\mathcal{A}_{\sigma_k}^{\text{GS}}(t_k)\right]_i$
  end for
  $a = \frac{2}{m}(\bar{a} - b^T)$ ▷ Deduce the estimate of $\nabla_x \, \text{MSE}(x, b)|_{x=\mathcal{A}_\sigma^{\text{GS}}(t)}$
  Sample $v_1, \ldots, v_N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_k^2 I_2)$
  Initialize matrix $J = \text{zeros}(m, 2)$
  for each receiver r_i for $i = 1, \ldots, m$ **do**
    $D_{i,j} \leftarrow \Phi(t - v_j, r_i)$ for $j = 1, \ldots, N$ ▷ Evaluate power at r_i
    $J_{i,:} \leftarrow \frac{1}{N\sigma_k^2}\sum_{j=1}^N -D_{i,j}v_j$ ▷ Compute the estimate of $\left[\text{J}\mathcal{A}_\sigma^{\text{GS}}(t)\right]_{i,:}$
  end for
  $g = (aJ)^T$ ▷ Vector-matrix multiplication to get the estimate of $\nabla\mathcal{L}_\sigma^{\text{TTS}}(t_k)$
  $t \leftarrow t - h_k\, g$
  $t \leftarrow \texttt{project}(t, \text{room})$ ▷ Ensure to stay inside the room
  Save $t_k \leftarrow t$, $g_k \leftarrow g$
end for
return $(t_k)_{k=1}^T$, $(g_k)_{k=1}^T$

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**