Fast Algorithms for Long-Range Wave Propagation over Complex Terrain

by

Max Bright

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in the University of Michigan
2022

Doctoral Committee:
Professor Eric Michielssen, Chair
Professor Anthony Grbic
Professor Robert Krasny
Professor Theodore Norris
Professor Leung Tsang

Max Bright

mabright@umich.edu

ORCID ID: 0000-0003-2496-9070

Acknowledgements

While a doctoral thesis is supposed to reflect my individual contribution to scientific research, this was by no means a one-person process. I owe an immense gratitude to my advisor, Professor Eric Michielssen, for his guidance throughout my PhD work. His attention to detail and pursuit of scientific rigor was instrumental to my growth as a researcher, and perhaps equally importantly, as a technical communicator. I would be nowhere near the academic I am today without his guidance.

I would like to thank each of the members of my doctoral committee Professor Leung Tsang, Professor Anthony Grbic, Professor Robert Krasny, and Professor Theodore Norris, for their participation in this process and for their detailed contributions to the refinement of this thesis.

I also want to thank my professors and friends in the RADLAB and in the broader ECE program. I was among some of the brightest academics in the world, and between new research ideas, advice on navigating graduate school, and even mere office banter, the role my ECE peer group has played in my success cannot be understated. Andrew Wintenberg, Reid Pinkham, and David Geroski were significant sources of advice in my journey.

I would not have completed my doctoral degree without the continuous and unrelenting support of my friend groups outside of graduate school. The list of important people in this category is too extensive to fit into this page, but I want to particularly highlight Trevor Hedges, Casey Gooch, Jacob Watkins, Spencer Peters, Brandon Spicer, Fielder Gall, Wyatt Stone, Aneess Kanjo, William Lewis, Michael Thomasma, Tim Millerman, Sean Estes, and Jalen Register. I also want to acknowledge old mentors Theron Cross, Professor Yasuo Kuga, and Nick Gutierrez for their influence on my decision to pursue an academic career.

My family played a substantial role in their aid and mentorship throughout this experience. My parents Walter and Trish Bright have always been on my side, and my extended family members Roselie Bright and Mark Lewandoski have provided invaluable academic guidance.

Table of Contents

Acknowledgements ii
List of Figures vi
List of Tables xi
Abstract xii
Introduction 1
Motivation 1
Review of previous work 1
Advancements Proposed by this Work 5
Chapter 1. Gabor Frame-Based Sparsification, Radiation Boundary Conditions, and Phase Screens for One-Way Wave Equations 7
1.1 Introduction 7
1.2 Split-Step Fourier Migration 8
1.3 Gabor Transforms 10
1.4 Propagation with Gabor Transforms 13
A. Gabor-Domain Migration 13
B. Sparsification of Gabor Coefficients 16
C. Local Inverse Gabor Transform 16
D. Radiation Boundary Conditions 20
1.5 Phase Screens in the Gabor Domain 21
A. Propagation Matrix with Built-in Phase Screen 22
B. Refractive Index Profiles with Nonuniform Slope 23
1.6 Numerical Results 25
A. Sparsification Error 25
B. Radiation Boundary Conditions 26
C. Earth Curvature 27
D. Sparse Long-Range Propagation 30

Chapter 2. Adaptive Multiresolution Gabor Transforms for Optimal Sparse Field Representations 34
2.1 Introduction 34
2.2 Multiresolution Gabor Transform 34
2.3. Multiresolution Gabor Propagation 41
2.4. Numerical Results 42
A. Validation 42
B. Long-Range Propagation 46
Chapter 3. Hybrid Split-Step Fourier – Finite Difference Solver for Long-Range Propagation over Rural Terrain 50
3.1 Introduction 50
3.2 Construction of Split-Step Fourier and Finite Difference Methods 50
A. Finite Difference Solver 50
3.3 Hybridization of Split-Step Fourier and Finite Difference 65
A. Split Fields into Two Regions 66
B. March Fields Through Upper Region with SSF 68
C. March Fields Through Lower Region with FD 68
D. Smoothly Blend Fields in Upper and Lower Regions 70
E. Additional Enhancements 71
3.4 Numerical Results 71
A. Validation 71
B. Propagation Studies 76
Chapter 4. Sparse Gabor Frame-Based Propagation in 3D 79
4.1 Introduction 79
4.2 3D Propagation with Split-Step Fourier 79
4.3 2D Gabor Analysis 81
4.4 3D Propagation in the Gabor Domain 84
4.5 Sparsification in 3D 87
4.6 Radiation Boundary Conditions in 3D 88
4.7 Gabor-Domain Phase Screens in 3D 89
4.8 Numerical Results 90
A. Validation 90
B. Long-Range Propagation 92

C. Radiation Boundary Conditions 95
D. Sparsity Effects 97
Chapter 5. Terrain Modeling in 3D 100
5.1 Introduction 100
5.2 Impedance Boundary Conditions 100
5.3 Hybridization of Gabor Propagation and IBCs 102
5.4 Numerical Results 104
A. Validation of Reflection Coefficient 104
B. Reflection from Rough Surface 106
C. Propagation over Terrain 108
Chapter 6. Conclusion 112
6.1 Research Contributions 112
6.2 Future Work 113
Bibliography 114

List of Figures

Fig. 1.1. Visualization of the flow of information for a Split-Step Fourier solver. An initial field profile, formed by a transmitter, is defined along a vertical slice on the left-hand side given by $\boldsymbol{\psi}[0, \mathbf{z}]$. Each step in the SSF method marches the field to the right by Δx. Interactions from the atmosphere or ground are modeled at each step. 8

Fig. 1.2. Visualization of the real parts of Gabor window functions, with $N_p = 128$: No modulation or shift (blue), spatially shifted with no frequency modulation (green), no spatial shift with frequency modulation (orange). Note that windows overlap in space and have local support of width N_p. 11

Fig. 1.3. Generation of one $(N \times M)$ submatrix of the propagation matrix set. A beamlet (left) is marched forward by Δx (middle), and its Gabor coefficient set is computed (right) and written into the propagation matrix. 14

Fig. 1.4. Wave propagation in the Gabor domain. Each nonzero coefficient in the Gabor representation of an initial wavefront $\phi_{s,f}[x_0]$ (left) is multiplied by an associated propagation matrix $\mathbf{P}[s, f, \mathbf{n}, \mathbf{m}]$ (center) and added together to form the resultant Gabor coefficients $\phi_{n,m}[x_0 + \Delta x]$ (right). 15

Fig. 1.5. A wavefront is advanced in the Gabor domain (left, center), and Inverse Gabor Transforms are used intermediately to compute the spatial-domain wavefront (right). 16

Fig. 1.6. Error of a propagating Gaussian beam, measured at different sparsification threshold levels. 26

Fig. 1.7. Comparison of the reflection coefficient of three implementations of upper domain truncation (Gabor RBCs, Absorbing Layer, combined Gabor RBCs and Absorbing Layer), versus angle of incidence measured from horizontal. 27

Fig. 1.8. Plot of path loss for a Gaussian beam marching through an ionosphere with Earth curvature correction 29

Fig. 1.9. Log-scale plot of the Gabor coefficient set at $x = 9950$ meters. Colored bins are stored coefficients, empty bins correspond to no stored data. 29

Fig. 1.10. Left: Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the ionosphere + Earth curvature case. Right: Plot of the error incurred by the Gabor method at each position in range for the ionosphere + Earth curvature case. 30
Fig. 1.11. Trilinear ocean refractive index profile. 31
Fig. 1.12. Plot of path loss for a Gaussian marching through an ionosphere with Earth curvature correction. 32
Fig. 1.13. Log-scale plot of the Gabor coefficient set at $x = 49550$ meters. Colored bins are stored coefficients, empty bins correspond to no stored data. 32
Fig. 1.14. Left: Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the ionosphere + Earth curvature case. Right: Plot of the error incurred by the Gabor method at each position in range for the ionosphere + Earth curvature case. 33
Fig. 2.1. Example of a partitioning of the Gabor space-frequency domain; different Gabor window widths, denoted by N_p, are used to model different regions. 35
Fig. 2.2. Example of the Gabor window functions of two neighboring regions with window widths $N_p = 64$ and $N_p = 256$, showing insufficient overlap at the interface if the space-frequency plane were perfectly partitioned. 36
Fig. 2.3. Revision to Fig. 2.1, showing what the true multi-window divisions of the space-frequency plane look like; each of the four subplots shows the portion of the space-frequency plane covered by each window width. Note that such regions overlap; the support for the region $N_p = 8$ bleeds into the coverage of both of the neighboring $N_p = 16$ regions. 37
Fig. 2.4. Example of a simple spatial-domain function. The vertical orange lines represent the different subdivisions of the domain, where within each subdivision, Multi-Gabor picks the optimal window width. The optimally picked window widths are shown in red for each subdivision. 43
Fig. 2.5. Plot of path loss for a synthetic beam excitation. 44
Fig. 2.6. Plot of the initial excitation for the synthetic beam case. 45
Fig. 2.7. Plot of the error incurred by (blue) Multi-Gabor and (orange) uniform-resolution Gabor at each position in range for the synthetic beam case. 45
Fig. 2.8. Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the synthetic beam case for (blue) Multi-Gabor and (orange) uniform-resolution Gabor 46
Fig. 2.9. Plot of path loss for a Gaussian beam excitation over an ocean duct with knife edges. 47
Fig. 2.10. Initial excitation for a Gaussian beam excitation over an ocean duct with knife edges. 48

Fig. 2.11. Plot of the error incurred by (blue) Multi-Gabor and (orange) uniform-resolution Gabor at each position in range for the Gaussian beam excitation over an ocean duct with knife edges. 48
Fig. 2.12. Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the Gaussian beam excitation over an ocean duct with knife edges case for (blue) Multi-Gabor and (orange) uniform-resolution Gabor. 49
Fig. 3.1. Visualization of the sweeping preconditioner. An initial slice is provided along the left side of the domain (top) given by $\boldsymbol{\psi}_{\mathrm{F}}$, and is marched forward (middle, bottom) for $b+1 \le m \le n_x$ by solving a localized Helmholtz equation $\mathbf{H}_m$ to produce $\boldsymbol{\psi}_m$, with a PML along the left side of the local solution region. 61
Fig. 3.2. Visualization of two-front PML. Two field slices are defined along the left and right edges of the domain (top), and the local Helmholtz problems are solved from both sides iteratively sweeping towards the center (bottom). 65
Fig. 3.3. Illustration of how fields are marched in the hybrid scheme; fields in the atmosphere are marched with SSF, while fields near the ground are marched with FD. Fields near the interface are smoothly blended together to form a complete profile. 65
Fig. 3.4. Flow diagram for marching fields with the hybrid scheme. 66
Fig. 3.5. Visualization of the SSF and FD regions, showing the overlap between the regions and the corresponding window functions. 67
Fig. 3.6. The formulation of the FD region, including PML placement, TF/SF region definitions, and locations of incident fields and output fields. Note that two input and output field slices are used due to the TF/SF formulation using two concentric contours. 69
Fig. 3.7. Comparison of the theoretical Fresnel reflection coefficient and the experimental reflection coefficient produced by the solver, for TM (top) and TE (bottom) polarizations. 73
Fig. 3.8. Heatmap of path loss for one realization of scattering from a sinusoidal surface. 74
Fig. 3.9. Field strength along the last vertical slice in x for FD domain height above terrain = 175, FD step size $dz = h$ modulated, and $\Delta x = 10$ m. 74
Fig. 3.10. Field strength along the last vertical slice in x for FD domain height above terrain = 175, FD step size $dz = h = 0.1$ m, and Δx modulated. 75
Fig. 3.11. Field strength along the last vertical slice in x for FD domain height above terrain modulated, FD step size $dz = h = 0.1$ m, and $\Delta x = 10$ m. 75
Fig. 3.12. Heatmap of field strength, in decibels, for the hilly forest terrain simulation. 77
Fig. 3.13. Heatmap of field strength, in decibels, for the long-range Eastern Washington case 78

Fig. 4.1. Coordinate system for Split-Step Fourier in 3D. An initial wavefront defined along the $z-y$ plane, and is sequentially marched from slice to slice by Δx along the $+x$ direction. The vertical slices show a wavefront evolving through space.....81
Fig. 4.2. Visualization of absolute value of real part of 2D Gabor window functions $\mathbf{G}_{n1,n2,m1,m2}[z,y]$, with $N_{p1}=N_{p2}=32$, and with $(n_1,n_2,m_1,m_2)=(0,0,0,0)$ (left) and $(n_1,n_2,m_1,m_2)=(0,0,4,2)$ (right).....82
Fig. 4.3. 3D dual window function $\tilde{\mathbf{G}}_{n1,n2,m1,m2}[z,y]$, with $N_{p1}=N_{p2}=32$ and $(n_1,n_2,m_1,m_2)=(0,0,0,0)$. 82
Fig. 4.4. Heatmap of the field strength (in dB) of a Gaussian beam propagating through a simple linear atmosphere.....91
Fig. 4.5. Percent error of each slice in x for a Gaussian beam propagating through a simple linear atmosphere.....91
Fig. 4.6. Number of nonzero field coefficients at each slice in x, for a Gaussian beam propagating through a simple linear atmosphere. 92
Fig. 4.7. Heatmap of the field strength (in dB) of a Gaussian beam propagating through a trilinear duct. 93
Fig. 4.8. Percent error of each slice in x for a Gaussian beam propagating through a trilinear duct. 94
Fig. 4.9. Number of nonzero field coefficients at each slice in x, for a Gaussian beam propagating through a trilinear duct. 94
Fig. 4.10. Heatmap showing field strength for one realization of the experiment to characterize Gabor RBCs (left) and Gabor RBCs plus an absorbing layer (right), for a steep angle of incidence on the radiation surface. 96
Fig. 4.11. Heatmap showing field strength for one realization of the experiment to characterize Gabor RBCs (left) and Gabor RBCs plus an absorbing layer (right), for a shallow angle of incidence on the radiation surface.....96
Fig. 4.12. Plot of the attenuation of two different boundary conditions versus incident angle: Gabor RBCs (blue), and Gabor RBCs with an added absorbing layer (orange).....97
Fig. 4.13. Heatmap of the field strength of one realization of the sparsification study. 98
Fig. 4.14. Error induced by sparsification of a Gaussian beam at each vertical slice. Each line represents a different level of sparsification. 98
Fig. 5.1. Partitioning of the domain into the Gabor region, where fields are marched with a 3D Gabor propagator, and the IBC region, where fields are marched with 3D Split-Step Fourier augmented with Impedance Boundary Conditions. 103
Fig. 5.2. Simulated reflection coefficient versus theoretical Fresnel reflection coefficient at different incident angles for TM incidence (top) and TE incidence (bottom) 105

Fig. 5.3. Heatmap of field strength (dB) of a Gaussian beam scattering off of a rough surface.....106
Fig. 5.4. Number of stored Gabor coefficients at each location in x in the rough surface scattering problem. 107
Fig. 5.5. Error of fields at each location in x, measured as the square of the difference in magnitudes between the hybrid Gabor-IBC fields, and the fields solved with a full SSF solver with IBCs.....107
Fig. 5.6. Error of fields at each location in x for the rough surface experiment shown in fig. 5.4; each line is for a different compression level. 108
Fig. 5.7. Heatmap of field strength (dB) of propagation over 3D terrain. 109
Fig. 5.8. Number of stored Gabor coefficients at each location in x in the 3D terrain problem.....110
Fig. 5.9. Error of fields at each location in x, measured as the square of the difference in magnitudes between the hybrid Gabor-IBC fields, and the fields solved with a full SSF solver with IBCs.....110

List of Tables

Table 1.1. Comparison of CPU time and number of stored points for the simulation in Fig. 1.9 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated. 30

Table 1.2. Comparison of CPU time and number of stored points for the simulation in Fig. 1.12 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated. 33

Table 4.1. Comparison of CPU time and number of stored points for the simulation in Fig. 4.4 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated. 92

Table 4.2. Comparison of CPU time and number of stored points for the simulation in Fig. 4.7 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated. 95

Abstract

Parabolic Wave Equations are an area of extensive research in the description of wave propagation. The Split-Step Fourier (SSF) method solves the Parabolic Wave Equation spectrally and is a method of choice for long-range propagation through atmosphere. Split-Step Fourier methods, however, are unable to sparsely represent fields and require repeated forward and inverse Fourier Transforms. Furthermore, Radiation Boundary Conditions (RBCs) are cumbersome to implement due to the Periodic Boundary Conditions enforced by the spectral propagator. This thesis solves the one-way wave equation in 2D and 3D with Gabor Transforms, representing propagating fields as a sum of locally supported frame functions with spatial shifts and frequency modulations. Gabor Transforms easily exploit sparsity in the space-frequency representation of structured fields. By precomputing the propagation characteristics of each frame function, a Gabor transformed wavefront can be efficiently propagated from one spatial slice to the next. RBCs are trivially implemented by removing frame functions that propagate outside the computational domain (i.e. beyond certain height bounds) from consideration, a feat that is impossible using classical split-step Fourier methods. Phase screens, formerly requiring immense computational resources to be applied in the spatial domain, are implemented in the Gabor domain.

The choice of Gabor frame is critically important for sparsification; the optimum window width must be selected to match the field characteristics of a particular scenario. Conventional Gabor frames have uniform window size for all spatial and frequency shifts. However, real-world problems rarely have uniform field complexity. This thesis will demonstrate the use of the jigsaw puzzle Gabor frame to optimally sparsify propagating fields while maintaining accuracy. The jigsaw puzzle frame is characterized by Gabor window functions that have different widths at different locations in space, rather than uniform window widths over all space. This framework naturally and efficiently accommodates the multi-scale nature of realistic propagation scenarios.

This thesis introduces hybrid solvers in 2D and 3D to describe propagation over terrain. In 2D, this thesis outlines a hybrid SSF-FD solver that combines the efficiency of SSF and the modeling flexibility of FD-based schemes. The solver maintains the SSF approach of advancing

free-space propagating fields using spectral propagators, but invokes a localized FD scheme to account for field interactions with terrain. From an operational perspective, the hybrid solver slices up the computational domain just like standard SSF- and FD-based solvers. In 3D, a similar hybridization is performed between a sparse 3D Gabor propagator for fields moving through upper atmosphere, and a SSF solver augmented with Impedance Boundary Conditions for fields about the terrain. These hybridizations substantially reduce the computational complexity of propagation problems by only using expensive field descriptions for propagation near terrain, and using fast propagators for all other space.

Introduction

Motivation

Parabolic Wave Equations (PWEs) have been extensively used to model long-range wave propagation [1]. There exist many PWE solvers with varying tradeoffs in accuracy, computational complexity, and ease of implementation. Among the available options, the Split-Step Fourier (SSF) method has become widely used due to its performance and its mathematical simplicity [2]. SSF methods use a spectral propagator to march fields in the Fourier domain, and weak inhomogeneities in the environment can be described by a spatial-domain phase screen. SSF has been successfully used to model propagation through environments such as air above rough oceans [3], troposphere above irregular terrain [4], and air above buildings on flat terrain [5]. The SSF method has several major limitations: 1) Sharp transitions in environment, such as an interface between atmosphere and ground, canot be described by SSF, 2) The repeated Fourier Transforms to move between the spatial and spectral domain scale with $O(n \log n)$, where n is the vertical domain size, and 3) the Fourier-domain propagator enforces periodic boundary conditions, which render it difficult to implement Radiation Boundary Conditions (RBCs) that absorb waves that leave the domain. Terrain can be described rigorously with Finite Difference (FD) or other rigorous solvers, but their scalability is worse than SSF and they are thus impractical for long-range propagation problems. Computational complexity limitations of SSF are unavoidable without major overhauls due to the use of Fourier Transforms. Boundary conditions are also a challenge with SSF; RBCs are typically implemented with Absorbing Boundary Conditions (ABCs) [6] which absorb incoming radiation, but for steep angles of incidence they must be thick layers which leads to excess resource usage.

Review of previous work

Several efforts have been made to improve the computational efficiency of the Split-Step Fourier method, particularly by using representing information with more coarse domains in order to cut down on the time taken to compute successive Fourier Transforms. For example, [7] introduced the use of the Padé approximation to the square root operator used in the time-domain

representation of the split-step Fourier method to increase the speed of computation, and it is compatible with less dense sampling along the domain of propagation [8]. For frequency-domain problems, however, the square root operator does not need to be approximated at all, so compromises in accuracy are unnecessary. Methods covered in [9] leverage numerical integration techniques for more accurate forward and inverse Fourier transforms, which can be paired with coarser domains. Less dense domains are effective provided problem constraints don't dictate a minimum resolution.

Several methods have been proposed for improving RBCs for split-step propagators. Perfectly-Matched Layers (PMLs) for Finite-Difference propagators were introduced by [10], and were further revised by [11]. [12] developed a closed-form expression for transparent boundary conditions for finite-difference schemes. An adaptation of PMLs for Fourier methods was proposed by [13] which models the layer as a complex region where fields decay, which can be as few as 10-50 points thick but loses efficiency for shallow-angle propagation. The ideas were further improved when a method for Absorbing Boundary Conditions that use imaginary rather than complex finite element lengths for decay was developed by [14], which can be adapted for both time and frequency-domain propagation schemes. While such developments substantially reduce the storage requirements of absorbing layers, they ultimately still rely on spending resources to represent regions of decaying fields.

The Gabor-Daubechies transform is a method for time-frequency analysis of a signal. The Gabor transform decomposes a wavefront into an overcomplete set of smooth, locally supported frame functions, consisting of spatially shifted and frequency modulated windows [15]. Gabor Transforms, and more generally Wavelet Transforms, have received recent attention for parabolic wave equation solvers. Gabor Transforms were first used in [16] for PWE-based modeling of geophysical problems, where a function is decomposed into localized propagating beamlets. Subsequently, a method coined "Split-Step Wavelet", developed in [17] and [18], used wavelet-domain propagators very similar to Gabor propagators and was exploited for its ability to sparsely represent wavefronts. To account for atmospheric inhomogeneities, phase screens were typically implemented by computing an inverse wavelet transform and applying the operator in the spatial domain. More recently, [19] moved the phase screen into the wavelet domain by approximating the atmospheric refractivity as constant over each wavelet.

Wavelet methods benefit from multiscale discretizations, and some adaptive multiscale wavelet-based methods have been applied to other types of PDEs. In [20], solutions to several different PDEs are demonstrated, where a multiscale wavelet transform adaptively accommodates shocks and other sharp transitions; the method was upgraded to 2D in [21]. Adaptive wavelet methods were extended to the wave equation in [22] using general time-stepping ODE solver schemes. The sparsification advantages of domain-partitioning with multiscale analysis have also been exploited in PDE literature, particularly for hyperbolic equations in [23] and [24]. In a similar vein, the Schrodinger equation has been sparsely solved with an adaptive wavelet model for finite difference methods [25]. Adaptive methods have yet to be extensively applied to one-way wave equations for propagation problems.

Multi-window Gabor transforms can utilize different window widths over different spatial regions. Multi-window Gabor transforms have been employed to represent features of mixed resolutions for signal analysis [26], but they have not yet been used for propagation. While some wavelet-based propagation methods [27] use multiresolution frames, the adaptability of such methods remain limited; the wavelet frames are still fixed and have to be defined by a user in advance of computing propagated fields, and there is no partitioning of space into regions of higher or lower complexity. A fixed user-selected frame, even if multi-window, may not be optimal when the user is unable to predict what types of field patterns will emerge in the propagation scenario.

The problem of long-range propagation over terrain has been extensively studied, and there are a variety of solution methods. Semi-empirical models such as Longley-Rice [28] or asymptotic methods such as ray-tracing [29][30][31] or GTD [32] may be employed, but they sacrifice accuracy for simplicity of computation. Numerically rigorous schemes like Finite Element [33], Integral Equations and Finite Difference Time Domain [34][35] can also be used when computational resources permit, but they are impractical for large-scale problems.

There also exist many methods for implementing terrain scattering into SSF. While simple PEC boundary conditions via imaging can be easily modeled with SSF, more complex terrain is difficult to implement due to phase screens only being accurate for weak inhomogeneities. Several methods have been proposed to remedy this issue. Hard truncation with knife-edges [36] can approximate diffractive features such as buildings. Staircasing is more robust, but tends to only be useful for modeling obstructive and diffractive terrain elements, and does not accurately describe scattering behavior. Impedance Boundary Conditions [37] may be used to more accurately

describe the ground medium. Many implementations of such boundary models invoke conformal coordinate transformations to better accommodate terrain height changes [4] [38] [39]. Many of these methods still have difficulty describing inhomogeneous ground features. For example, at low frequencies, tree cover can be described as a four-layered medium [40] [41]. If the terrain is flat, some of the existing methods could be adapted to the four-layer model, but implementation becomes extremely cumbersome if the terrain is not flat.

Finite Difference methods rigorously solve the wave equation. The primary advantage of FD solvers is that they can accommodate complex geometry with ease, and they are compatible with a variety of boundary conditions including Perfectly-Matched Layers (PMLs) [42]. In the case of long-range propagation, Finite Difference solvers have been adapted to solve the PWE in the spatial domain. FD PWE solvers were first developed for ocean acoustics [43], and they have been successfully employed to model irregular terrain in electromagnetics problems [44].

Several solvers that hybridize PWEs with numerically rigorous solvers have already been proposed to efficiently simulate complex geometry. Hybrid Parabolic Equation / Finite Difference Time Domain solvers are outlined in [45] and [46] which use FDTD to add a scattering correction to the standard PWE methods. In the case of indoor propagation, there exist several other hybrid solvers invoking PWEs and full-wave solvers [47] [48], or FDTD and asymptotic models such as UTD [49]. The limitation of many existing hybrid methods is that they are restricted to low frequency bands, scattering from electrically small objects, or propagation within confined spaces.

There is an extensive body of literature on solving PWEs in 3D [50] [51] [52], particularly with Split-Step Fourier (SSF) [5]. Studying PWEs in 3D, especially under hybridization with rigorous numerical solvers, enables a full description of environments such as hilly terrain [45][53], forest [54], confined spaces [48], or scenarios with buildings [5] and scatterers [55].

Many 3D schemes suffer from performance issues due to the requirements for large, expensive domain sizes. 3D SSF, due to its use of Fourier Transforms, scales with $O(n^2 \log n)$, where n is one side length of the domain. Some efforts have been made to improve the performance of 3D solvers, for example the $N \times 2\mathrm{D}$ method [56], which solves a set of 2D PWEs along consecutive planes to fill out a 3D region. The $N \times 2\mathrm{D}$ method has the drawback of limited accuracy in modeling interactions transverse to the direction of propagation.

Domain truncation, like in the 2D case, remains an issue with propagation in SSF in 3D. SSF enforces periodic boundary conditions, and it is difficult to implement a model for radiation

boundary conditions that mitigates the tendency for waves to either reflect or wrap around at domain boundaries. Typically, an absorbing layer is used for SSF problems in 3D [57], however this leads to immense waste of memory and computational time in processing fields in absorbing regions. In some cases, a domain can be tuned such that negligible radiation approaches the boundaries in order to subvert domain truncation issues entirely. While in many scenarios it may be easy to avoid unwanted radiation propagating towards the top and bottom domain walls, there are more limited scenarios and excitations where there is minimal spreading towards the domain side walls. Recent literature such as [58] solves the one-way wave equation along an angular width and invokes periodic boundary conditions along the boundaries of the angular slice, under an assumption that radiation will not spread beyond the angular width. While such a method works well in many scenarios, the flexibility remains limited and solving the wave equation in cylindrical coordinates requires the computational domain to have decreasing cartesian resolution as the radius increases. Furthermore, the angular slice must be wide enough that no energy reaches the side walls of the domain, which consumes computational resources.

Limitations in terrain modeling are exacerbated in three dimensions. 2D terrain models can be used in 3D problems via the $N \times 2\text{D}$ method [59], however the limitations of the method mentioned previously in this section persist. More rigorous solvers and scattering models are often impractical for describing detailed, large-scale 3D geometry, and are thus typically delegated to hybrid solvers; many such hybrid solvers, however, still largely rely on Parabolic Wave Equation schemes that store fields in full uncompressed form [5][45][46]. Finite-Difference-Based Parabolic Wave Equation solvers can model geometry accurately and without need for hybridization [53], and simpler terrain models based on IBCs [54], staircasing [60], or knife edges [61][62] can be used for faster simulation, but such solvers still do not compress propagating fields.

Advancements Proposed by this Work

Chapter 1 introduces a Gabor Frame-based propagator to sparsely solve the one-way wave equation. The Gabor propagator is based on the conventional Split-Step Fourier propagator, with the principal advantage that propagating field information can be sparsified for fast computation. The Gabor method is improved beyond the state-of-the-art Split-Step Wavelet models by introducing an exact implementation of phase screens in the Gabor domain, as well as by introducing a zero-memory model for Radiation Boundary Conditions.

Chapter 2 discusses a multiscale improvement to the Gabor method. An adaptive multi-window Gabor frame is best suited to sparsely represent complex multiscale field profiles. In particular, this chapter uses the space-frequency jigsaw puzzle method, introduced by [63]. The jigsaw puzzle method is a multi-window Gabor expansion that splits the space-frequency plane into a set of different regions, and for each region, picks the window width that maximizes sparsity. This multi-window transform can be used to propagate fields in the same manner as in existing split-step wavelet methods.

Chapter 3 proposes a hybrid Split-Step Fourier – Finite Difference solver for one-way wave equations over long ranges. This new solver will invoke SSF to spectrally propagate waves through the atmosphere, and a FD solver will be locally applied in a small domain about the surface of the terrain to model scattering and absorption. This method will maintain the efficiency of SSF for propagation through weakly inhomogeneous atmosphere, while utilizing the precision of FD solvers in minimally sized regions.

Chapter 4 discusses a 3D generalization of the topics in chapter 1. Previous SSW methods were implemented in 3D in [64], however there still exist properties of Gabor Transforms that can be taken advantage of. Gabor Transforms, which are easily generalized to 2D [65], can be used to represent 3D propagating fields within the Split-Step Wavelet framework. The uniform space-frequency grid of Gabor Transforms enables fast, uniform processing of all Gabor-domain data, which proves useful for Gabor-domain phase screens and RBCs in 3D.

Chapter 5 augments the solver introduced in chapter 4 with a terrain model. In a similar manner to chapter 3, a hybrid Gabor-IBC method is used to efficiently march fields in upper atmosphere while rigorously describing fields near the terrain surface. The 3D sparse Gabor solver is used to propagate atmospheric fields, and a 3d Split-Step Fourier solver subjected to Impedance Boundary Conditions is used to compute scattered fields from terrain that is described by staircasing.

Chapter 1. Gabor Frame-Based Sparsification, Radiation Boundary Conditions, and Phase Screens for One-Way Wave Equations

1.1 Introduction

This chapter will introduce the usage of Gabor Transforms to represent propagating fields. The Gabor Transform represents a wavefront as a summation of window functions that exhibit beamlike propagation. The propagation characteristics of each window function are precomputed and expressed in the Gabor domain; this enables the construction of a propagation operator that marches fields entirely in the Gabor domain.

If a field has local support or simple structure, its Gabor-domain representation can be sparsified; Gabor window functions with magnitudes that fall below a threshold can be deleted. The propagator then only needs to interface with nonzero Gabor coefficients, thus reducing the computational cost of the marching step. Furthermore, the precomputed propagation operators themselves can be sparsified, aided by the simplistic beamlike propagation of Gabor window functions. The CPU time per step therefore is determined by the number of nonzero elements of the sparse propagation matrix multiplied by those of the sparse space-frequency representation of the fields.

This chapter exploits several properties of Gabor-domain field propagation that have not yet been utilized in prior study. RBCs, rather than requiring thick absorbing layers, can be trivially implemented in the Gabor domain by deleting excited window functions that escape the domain. This method consumes zero memory and can be further improved via hybridization with a thin absorbing layer. Phase screens can also be described partially or fully in the Gabor domain. In the case of a linear refractive index profile, the phase screen can be built into the propagation characteristics of Gabor window functions, yielding an exact representation of fields marching through inhomogeneous atmosphere. If the refractive index profile is piecewise linear, the same technique can be adapted, with the caveat that local Inverse Gabor Transforms are used about the corners in the refractive index profile in order to apply the corners spatially. Localized transforms enable the processing of complex spatial-domain phenomena without requiring the storage of the

entire spatial domain, thus maintaining the sparsity advantage of the Gabor solver over conventional SSF solvers.

1.2 Split-Step Fourier Migration

The SSF method is a marching algorithm that can be used to solve the one-way wave equation. Consider a two-dimensional slice of atmosphere over either terrain or ocean, shown in Fig. 1.1. A transmitter is mounted above the surface on the left-hand side of the domain, and it supplies a field excitation. The direction to the right of the transmitter is denoted as the range direction, and the direction transverse to the ground is denoted as the vertical direction. The SSF method uses the excitation to propagate fields towards the right-side of the domain, accounting for refraction through the atmosphere and interaction with the surface. SSF solves a scenario by splitting the range direction into a sequence of vertical slices, and spectrally propagating fields from slice to slice using the one-way wave equation.

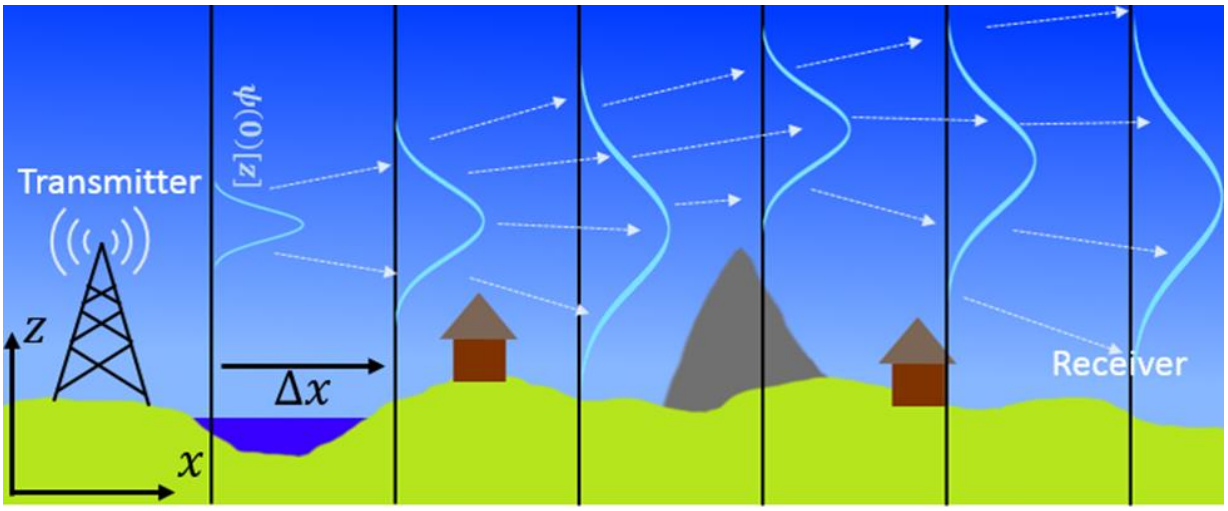


Fig. 1.1. Visualization of the flow of information for a Split-Step Fourier solver. An initial field profile, formed by a transmitter, is defined along a vertical slice on the left-hand side given by $\boldsymbol{\psi}[0, \mathbf{z}]$. Each step in the SSF method marches the field to the right by Δx. Interactions from the atmosphere or ground are modeled at each step.

Consider an electromagnetic wave propagating through 2D space, with the horizontal (range) direction denoted by x and the vertical direction denoted by z. Let the scalar quantity $\psi(x, z)$ be the field component transverse to the $x-z$ plane at a point (x, z). $\psi(x, z)$ represents the electric field strength for the TE propagation case, and the magnetic field strength for the TM propagation case. $\psi(x, z)$ satisfies the 2D Helmholtz equation, given by

$$\frac{\partial^2 \psi(x,z)}{\partial x^2} + \frac{\partial^2 \psi(x,z)}{\partial z^2} + k_0{}^2 n^2(x,z)\psi(x,z) = 0\,, \tag{0.1}$$

where k_0 is the free space wavenumber, and $n(x,z)$ is the refractive index of the medium at a point (x,z). An excitation $\psi(x_t,z)$ is supplied by a transmitter mounted at some $x < x_t$, and fields are assumed to be right-propagating. In terms of the excitation, for $x > x_t$, equation (0.1) has the exact Fourier Domain solution

$$\hat{\psi}(x,k_z) = \hat{\psi}(x_t,k_z)e^{in(x,z)k_x(x-x_t)}\,, \tag{0.2}$$

where $\hat{\psi}(x_t,k_z) = \int_{-\infty}^{\infty} \psi(x_t,z)e^{-ik_z z}dz$ is the Fourier Transform of the fields produced by the transmitter $\psi(x_t,z)$, and $k_x = k_0\cos\alpha$ and $k_z = k_0\sin\alpha$ are the horizontal and vertical wavenumbers respectively, where α is the angle of propagation from horizontal.

The fields $\psi(x,z)$ will be discretized in order to numerically evaluate (0.2). Let the vertical domain be discretized by the vector $\mathbf{z} = [0,\Delta z,...,(N_z-1)\Delta z]$ and the range direction be discretized by the vector $\mathbf{x} = [x_t, x_t+\Delta x,..., x_t+(N_x-1)\Delta x]$. Let $\boldsymbol{\psi}[x,z]$ be the field strength at some point in discrete x,z, and let the vector $\boldsymbol{\psi}[x,\mathbf{z}] = [\boldsymbol{\psi}[x,z_1],\boldsymbol{\psi}[x,z_2],\ldots,\boldsymbol{\psi}[x,z_{N_z}]]$. $\boldsymbol{\psi}[x,\mathbf{z}]$ will be referred to as the field profile at x. Scalar operations on vectors such as exponentiation are applied pointwise, and products of vectors of the same dimension are pointwise unless otherwise notated.

In the discrete domain, (0.2) becomes

$$\hat{\boldsymbol{\psi}}[x,\mathbf{k_z}] = \hat{\boldsymbol{\psi}}[x_t,\mathbf{k_z}]e^{i\mathbf{k_x}[\mathbf{k_z}]\bullet\mathbf{n}[x,\mathbf{z}](x-x_t)}\,, \tag{0.3}$$

where $\hat{\boldsymbol{\psi}}[x,\mathbf{k_z}] = \mathrm{FFT}_z\{\boldsymbol{\psi}[x,\mathbf{z}]\}$ is the Fast Fourier Transform (FFT) of $\boldsymbol{\psi}[x,\mathbf{z}]$, $\mathbf{k_z}$ is the angular frequency vector corresponding to $\mathbf{z}$, and $\mathbf{k_x}[\mathbf{k_z}] = \sqrt{1-\frac{\mathbf{k_z}^2}{k_0{}^2}}$.

Equation (0.3) cannot be used to obtain $\boldsymbol{\psi}[x,\mathbf{z}]$ directly, due to the mixing of spatial and spectral variables $\mathbf{z}$ and $\mathbf{k_z}$. The SSF method approximates a solution by splitting (0.3) into a frequency-domain propagation step to march fields through free space, and a spatial-domain

phase-screen step to approximate the effects of the atmosphere. The refractive index profile is assumed to be weakly inhomogeneous over z and invariant over x, and will be given by $\mathbf{n}[\mathbf{z}]$. The Split-Step Fourier algorithm to march a field slice at some $x_0 \in \mathbf{x}$ forwards to $(x_0 + \Delta x) \in \mathbf{x}$ is

$$\boldsymbol{\psi}[x_0 + \Delta x, \mathbf{z}] = \mathrm{FFT}\{\hat{\boldsymbol{\psi}}[x_0, \mathbf{k_z}] e^{i\mathbf{k_x}[\mathbf{k_z}]\Delta x}\} \cdot e^{ik_0 \Delta x(\mathbf{n}[\mathbf{z}]-1)}. \tag{0.4}$$

Equation (0.4) is called successively over each $x \in \mathbf{x}$ to produce all fields $\boldsymbol{\psi}[x, \mathbf{z}]$, starting from the initial slice $\boldsymbol{\psi}[x_t, \mathbf{z}]$. The phase screen operator $e^{ik_0 \Delta x(\mathbf{n}[\mathbf{z}]-1)}$ introduces an error of $O(\Delta x)$, so the step size must remain small.

To represent 3D field decay in the 2D scenario, a correction is made where fields decay cylindrically from the transmitter, given by

$$\boldsymbol{\psi}_{3\mathrm{D}}[x, \mathbf{z}] = \left(\sqrt{x_t} / \sqrt{x}\right) \boldsymbol{\psi}[x, \mathbf{z}], \tag{0.5}$$

with the condition $\boldsymbol{\psi}_{3\mathrm{D}}[x_t, \mathbf{z}] = \boldsymbol{\psi}[x_t, \mathbf{z}]$ at the excitation.

1.3 Gabor Transforms

This section introduces the Gabor Transform, which expresses a function as a weighted sum of window functions, each with a spatial shift and a frequency modulation.

Consider a vector $\mathbf{g}[\mathbf{z}]$ with length N_z and which is only nonzero on the first N_p elements. The (n, m) th Gabor window function with window width N_p is defined by

$$\mathbf{g}_{n,m}[\mathbf{z}] = \mathbf{g}[\mathbf{z} - na\Delta z] e^{i(mb)(\mathbf{z} - na\Delta z)}. \tag{0.6}$$

The Gabor window function is constructed by translating $\mathbf{g}[\mathbf{z}]$ by $na\Delta z$ and frequency modulating it by mb, where $n \in \mathbf{n} = [0,1,\ldots,N-1], m \in \mathbf{m} = [0,1,\ldots,M-1]$, $M = N_p$, $a = N_z / N = N_p / 2$, and $b = N_z / M$. Examples of window functions $\mathbf{g}_{n,m}[z]$ are shown in Fig. 1.2.

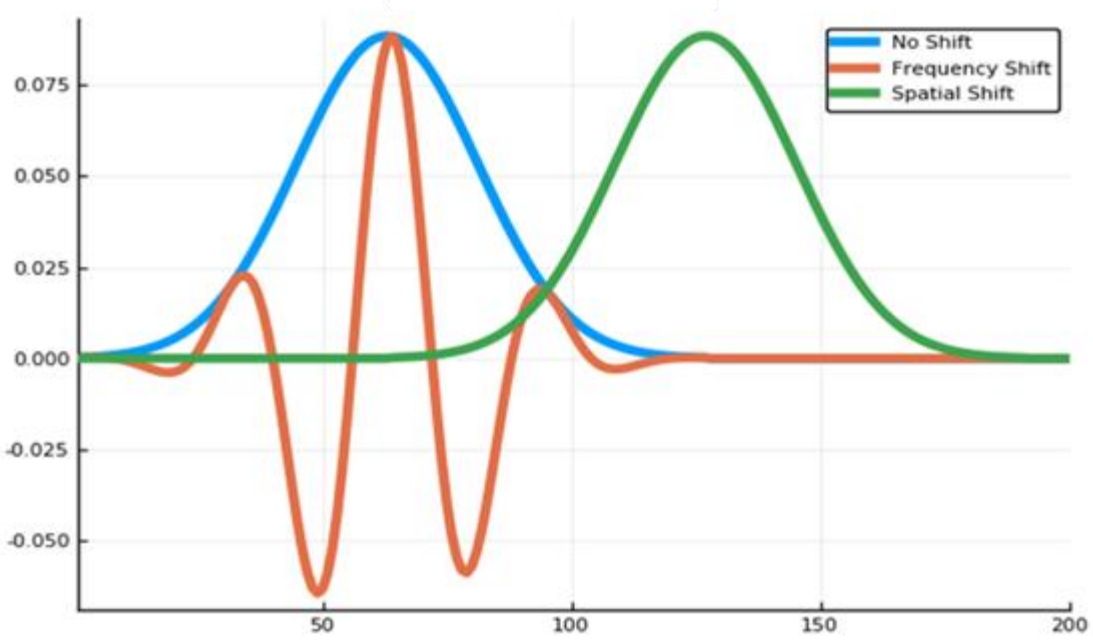


Fig. 1.2. Visualization of the real parts of Gabor window functions, with $N_p = 128$: No modulation or shift (blue), spatially shifted with no frequency modulation (green), no spatial shift with frequency modulation (orange). Note that windows overlap in space and have local support of width N_p.

A vector can be decomposed into a weighted sum of Gabor window functions, given by

$$\boldsymbol{\psi}[\mathbf{z}] = \sum_{n,m} \mathbf{g}_{n,m}[\mathbf{z}]\phi_{n,m}, \tag{0.7}$$

where each weight $\phi_{n,m}$ is a Gabor coefficient. The Gabor coefficients can not be computed by simply inverting (0.7), because the Gabor window functions are not orthogonal. In order to obtain an expression for $\phi_{n,m}$, a pseudo-inverse operation is required, which motivates the construction of a dual window function, which will now be discussed.

Let a $N_z \times MN$ matrix $\mathbf{C}$ contain all of the Gabor window functions, and let a vector $\mathbf{a}$ contain a collection of associated weights, so that

$$\mathbf{C} = \begin{bmatrix} \mathbf{g}_{1,1}[\mathbf{z}] & \mathbf{g}_{1,2}[\mathbf{z}] & \cdots & \mathbf{g}_{1,N}[\mathbf{z}] & \cdots & \mathbf{g}_{M,N}[\mathbf{z}] \end{bmatrix} \tag{0.8}$$

and

$$\mathbf{a} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} & a_{2,1} & \cdots & a_{M,N} \end{bmatrix}^{\mathrm{T}}, \tag{0.9}$$

where $a_{m,n} = \left\langle \boldsymbol{\psi}[\mathbf{z}], \mathbf{g}_{m,n}[\mathbf{z}] \right\rangle$, and $\langle\ \rangle$ denotes a dot product.

The vector $\mathbf{a}$ is written as a matrix product

$$\mathbf{a} = \mathbf{C}^{\mathrm{H}} \boldsymbol{\psi}[\mathbf{z}], \tag{0.10}$$

where H denotes the Hermitian. Equation (0.10) is solved for $\boldsymbol{\psi}[z]$ by computing the pseudoinverse of $\mathbf{C}$, yielding

$$(\mathbf{C}\mathbf{C}^{H})^{-1}\mathbf{C}\mathbf{a} = \boldsymbol{\psi}[z]. \tag{0.11}$$

The columns of the pseudoinverse of $\mathbf{C}$ are defined as the dual window functions $\mathbf{g}_{n,m}[z]$. It follows that the dual window functions have the expression

$$\tilde{\mathbf{g}}_{n,m}[z] = (\mathbf{C}\mathbf{C}^{H})^{-1}\mathbf{g}_{n,m}[z]. \tag{0.12}$$

The expression (0.11) can be written in terms of dual window functions as

$$\boldsymbol{\psi}[\mathbf{z}] = \sum_{n,m} \tilde{\mathbf{g}}_{n,m}[\mathbf{z}] \left\langle \boldsymbol{\psi}[\mathbf{z}], \mathbf{g}_{mn}[\mathbf{z}] \right\rangle. \tag{0.13}$$

Because (0.12) is invertible, the window and dual window functions in (0.13) can be interchanged to produce

$$\boldsymbol{\psi}[z] = \sum_{n,m} \mathbf{g}_{n,m}[z] \left\langle \boldsymbol{\psi}[z], \mathbf{g}_{n,m}[z] \right\rangle, \tag{0.14}$$

which is the representation of $\boldsymbol{\psi}[z]$ as a weighted sum of Gabor window functions. Either (0.13) or (0.14) may be used, depending on the application.

The Gabor coefficient matrix $\boldsymbol{\Phi}[\mathbf{n},\mathbf{m}]$ of size $N \times M$ is defined as the matrix of all Gabor coefficients of a function, where each of its elements $\phi_{n,m} = \boldsymbol{\Phi}[n,m]$ is given by

$$\phi_{n,m} = \left\langle \boldsymbol{\psi}[\mathbf{z}], \mathbf{g}_{n,m}[\mathbf{z}] \right\rangle. \tag{0.15}$$

The operator to build a matrix $\boldsymbol{\Phi}[\mathbf{n},\mathbf{m}]$ associated with $\boldsymbol{\psi}[\mathbf{z}]$ will be denoted as “GT”, standing for Gabor Transform, and is written as

$$\mathbf{\Phi}[\mathbf{n},\mathbf{m}] = \mathrm{GT}\{\boldsymbol{\psi}[\mathbf{z}]\}. \tag{0.16}$$

Equation (0.7) will henceforth be referred to as the Inverse Gabor Transform and the operator is denoted by $\boldsymbol{\psi}[\mathbf{z}] = \mathrm{IGT}\{\mathbf{\Phi}[\mathbf{n},\mathbf{m}]\}$, and the Gabor domain refers to the set of all $\phi_{n,m}$ associated with a vector $\boldsymbol{\psi}[\mathbf{z}]$.

1.4 Propagation with Gabor Transforms

This section will describe the propagation of fields with Gabor Transforms. Gabor-based propagation will be performed in the Gabor domain with a Gabor-domain operator, in an analogous manner to how Split-Step Fourier uses a frequency-domain operator to march fields in the Fourier domain.

A. Gabor-Domain Migration

Consider a scenario where a field profile at x_0 is specified by some dual window function $\tilde{\mathbf{g}}_{n,m}[\mathbf{z}]$. If this field is marched with Split-Step Fourier, its propagation will be beamlike given a suitable choice of dual window function, such as a Gaussian. Hence, a beamlet $\mathbf{b}_{n,m}[x,\mathbf{z}]$ is defined as the solution to a propagation scenario for $x \geq x_0$ where the initial profile is given by

$$\mathbf{b}_{n,m}[x,\mathbf{z}] = \tilde{\mathbf{g}}_{n,m}[\mathbf{z}]. \tag{0.17}$$

Beamlets will be the basis for the construction of a Gabor-domain propagator. First, the vector $\mathbf{b}_{n,m}[x_0 + \Delta x,\mathbf{z}]$ is computed by marching the initial profile forward by Δx with a SSF solver. The Gabor representation of the beamlet is subsequently computed, and the Gabor coefficient set of each marched beamlet is referred to as a propagation matrix. The propagation matrix is written as an element of a matrix of matrices $\mathbf{P}$. $\mathbf{P}$ is a 4-dimensional data structure of size $(N \times M) \times (N \times M)$, and is referred to as the propagation matrix set. The inner elements of $\mathbf{P}$, one example of which is shown in Fig. 1.3, are given by

$$\mathbf{P}[s,\, f\,,\mathbf{n},\mathbf{m}] = \mathrm{GT}\{\mathbf{b}_{s,\,f}[x_0 + \Delta x,\mathbf{z}]\}, \tag{0.18}$$

where the $(s,\, f)$ element of the propagation matrix set is referred to as a $N \times M$ propagation matrix, representing the Gabor Transform of a beamlet with spatial shift index $s \in [0,1,\ldots,N-1]$

and frequency modulation index $f \in [0,1,\ldots,M-1]$, marched forward through space by Δx. Each entry of an individual propagation matrix is denoted by

$$p_{n,m}[s,\, f\,] = \mathbf{P}[s,\, f\,, n, m] = \left\langle \mathbf{b}_{s,f}[\Delta x, \mathbf{z}], \mathbf{g}_{n,m}[\mathbf{z}] \right\rangle . \tag{0.19}$$

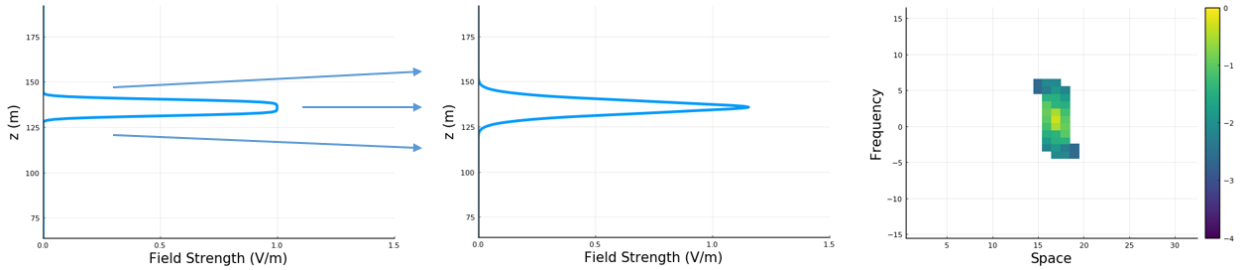


Fig. 1.3. Generation of one $(N \times M)$ submatrix of the propagation matrix set. A beamlet (left) is marched forward by Δx (middle), and its Gabor coefficient set is computed (right) and written into the propagation matrix.

The beamlet $\mathbf{b}_{s,f}[\Delta x, \mathbf{z}]$ is conversely expressed in terms of the propagation matrix set as

$$\mathbf{b}_{s,f}[\Delta x, \mathbf{z}] = \sum_{n,m} p_{n,m}[s,\, f\,]\tilde{\mathbf{g}}_{n,m}[\mathbf{z}] . \tag{0.20}$$

The propagation matrix set is generated as a precomputation step. At runtime, the procedure for marching fields begins with the computation of the Gabor Transform of an initial field profile $\boldsymbol{\psi}[x_0, \mathbf{z}]$ to produce coefficients $\phi_{s,f}[x_0]$. $\boldsymbol{\psi}[x_0, \mathbf{z}]$ is represented as an Inverse Gabor Transform

$$\boldsymbol{\psi}[x_0, \mathbf{z}] = \sum_{s,f} \phi_{s,f}[x_0]\tilde{\mathbf{g}}_{s,f}[\mathbf{z}] . \tag{0.21}$$

Since the initial profile of a beamlet is equivalent to a Gabor dual window function, (0.17) can be substituted into (0.21) to yield

$$\boldsymbol{\psi}[x_0, \mathbf{z}] = \sum_{s,f} \phi_{s,f}[x_0]\mathbf{b}_{s,f}[x_0, \mathbf{z}] . \tag{0.22}$$

$\boldsymbol{\psi}[x_0, \mathbf{z}]$ is then marched forward to $\boldsymbol{\psi}[x_0 + \Delta x, \mathbf{z}]$ with Split-Step Fourier. The Fourier Transform operator in SSF can be distributed across the summation on the right-hand side of (0.22) so that the marched field can be described as a sum of beamlets, written as

$$\boldsymbol{\psi}[x_0+\Delta x,\mathbf{z}]=\sum_{s,f}\phi_{s,f}[x_0]\mathbf{b}_{s,f}[\Delta x,\mathbf{z}]\,. \tag{0.23}$$

By using (0.20), (0.23) can be expressed in terms of the propagation matrix set as

$$\boldsymbol{\psi}[x_0+\Delta x,\mathbf{z}]=\sum_{s,f}\phi_{s,f}[x_0]\left(\sum_{n,m}p_{n,m}[s,\,f\,]\tilde{\mathbf{g}}_{n,m}[\mathbf{z}]\right). \tag{0.24}$$

This can be seen as rewriting the propagation step as the inverse Gabor transform of a set of Gabor-domain matrix operations, given by

$$\boldsymbol{\psi}[x_0+\Delta x,\mathbf{z}]=IGT\{\sum_{s,f}\phi_{s,f}[x_0]\mathbf{P}[s,\,f\,,\mathbf{n},\mathbf{m}]\}\,. \tag{0.25}$$

The profile $\boldsymbol{\psi}[x_0+\Delta x,\mathbf{z}]$ is equivalently described in the Gabor domain by $\boldsymbol{\Phi}[x_0+\Delta x,\mathbf{n},\mathbf{m}]$, whose elements $\phi_{n,m}[x_0+\Delta x]$ are given by

$$\phi_{n,m}[x_0+\Delta x]=\sum_{s,f}\phi_{s,f}[x_0]p_{n,m}[s,\,f\,]\,. \tag{0.26}$$

Equation (0.26) is an entirely Gabor-domain operator on the fields; the Gabor coefficients $\phi_{n,m}[x_0+\Delta x]$ of $\boldsymbol{\psi}[x_0+\Delta x,\mathbf{z}]$ are expressed as a function of the Gabor coefficients $\phi_{s,f}[x_0]$ of $\boldsymbol{\psi}[x_0,\mathbf{z}]$. In an analogous manner to Split-Step Fourier, a field profile is converted into the Gabor domain, marched with the Gabor-domain propagator (0.26), and converted back to the spatial domain for plotting and for spatial-domain operations such as phase screens. The application of the Gabor-domain propagator is visualized in Fig. 1.4 and Fig. 1.5.

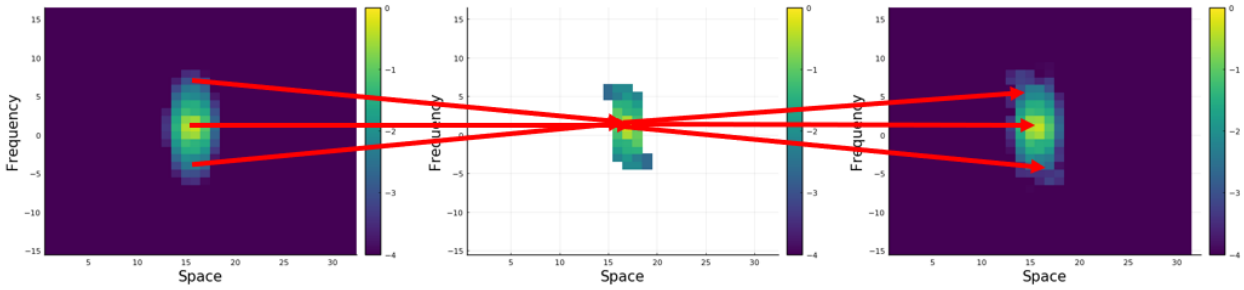


Fig. 1.4. Wave propagation in the Gabor domain. Each nonzero coefficient in the Gabor representation of an initial wavefront $\phi_{s,f}[x_0]$ (left) is multiplied by an associated propagation

matrix $\mathbf{P}[s, f, \mathbf{n}, \mathbf{m}]$ (center) and added together to form the resultant Gabor coefficients $\phi_{n,m}[x_0 + \Delta x]$ (right).

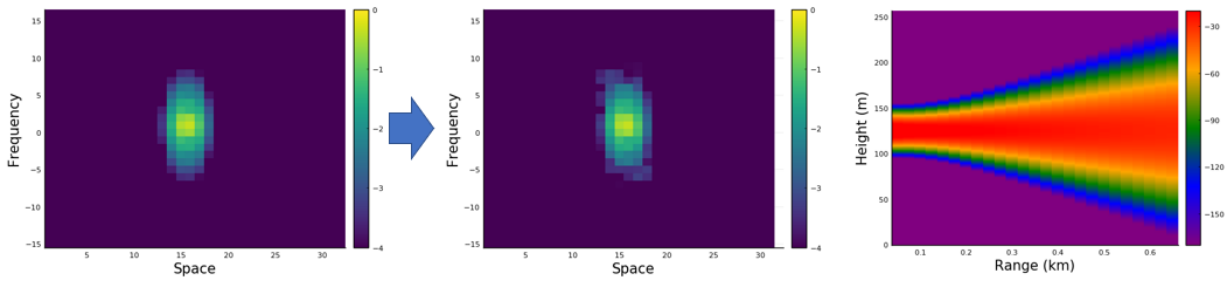


Fig. 1.5. A wavefront is advanced in the Gabor domain (left, center), and Inverse Gabor Transforms are used intermediately to compute the spatial-domain wavefront (right).

B. Sparsification of Gabor Coefficients

The principal advantage of propagation in the Gabor domain is the ability to sparsify field representations. A Gabor Coefficient matrix can be thresholded such that only coefficients above a certain tolerance are stored. By sparsifying the coefficient matrix at x_0 given by $\mathbf{\Phi}[x_0, \mathbf{n}, \mathbf{m}]$, (0.26) will only operate over a small number of s, f indices. This substantially reduces memory consumption and improves speed for propagation steps.

The sparsification operation is applied to $\mathbf{\Phi}[x_0, \mathbf{n}, \mathbf{m}]$ at each $x_0 \in \mathbf{x}$. In cases where a forward Gabor Transform is computed, either in generation of fields or in local spatial-domain processing, a sparsification is also performed.

In the same manner that propagating fields can be sparsified, the propagation matrices $\mathbf{P}[s, f, \mathbf{n}, \mathbf{m}]$ can also be sparsified. By reducing the number of nonzero elements of a propagation matrix, the propagation step (0.26) becomes an addition of sparse matrices, reducing computation time.

C. Local Inverse Gabor Transform

A critical design feature of Gabor-based propagation is the reconciliation of Gabor representations with spatial domain phenomena. In some cases, the spatial-domain operators can be represented exactly in the Gabor domain. In other cases, it is necessary to convert a field profile back to the spatial domain in order to apply an operator accurately. This section introduces a framework that will be used extensively in this thesis for processing spatial-domain phenomena while maintaining the sparsity advantages of Gabor frames.

In many cases, the spatial-domain operation to be performed only concerns some subset of the z domain; for example, an absorbing layer only occupies the edge of the domain. An inherent advantage of Gabor analysis is that it locally represents spatial information; this property can be exploited to only perform an inverse Gabor transform about some subset of the Gabor plane, rather than of the entire plane.

This section introduces the local inverse Gabor transform. The procedure is straightforward in principle: When computing an inverse Gabor transform of a Gabor representation with size $N \times M$, only some subset $\mathcal{N}_{\text{inv}} \subset \mathbf{n}$ of all N spatial indices are iterated over to obtain a spatial domain signal. To perform a forward transform, again only some subset $\mathcal{N}_{\text{fwd}} \subset \mathbf{n}$ of all N spatial indices are iterated over to construct a set of local Gabor coefficients.

A nontriviality in this procedure that must be addressed is that Gabor representations are not unique. If a spectrum supported by $\mathcal{N}_{\text{inv}}$ is transformed to the spatial domain and then transformed back to the Gabor domain, the support of the new representation is not necessarily $\mathcal{N}_{\text{inv}}$; rather, it can only be guaranteed that the support spans $\mathcal{N}_{\text{fwd}}$ plus all immediately neighboring indices n to the support of $\mathcal{N}_{\text{fwd}}$. So, for example, if $\mathcal{N}_{\text{inv}} = \{3,4,5\}$, then $\mathcal{N}_{\text{fwd}} = \{2,3,4,5,6\}$. This behavior is due to Gabor windows overlapping spatially.

Let an indicator vector $\mathbf{s}_n[\mathbf{z}]$ be defined by

$$\mathbf{s}_n[z] = \begin{Bmatrix} 1, & \mathbf{g}_{n,m}[z] \neq 0 \\ 0, & \mathbf{g}_{n,m}[z] = 0 \end{Bmatrix}, \tag{0.27}$$

where the vector $\sum_{n \in \mathcal{N}_{\text{inv}}} \mathbf{s}_n[\mathbf{z}]$ represents the total support of all $\mathbf{g}_{n,m}[\mathbf{z}]$ for all $n \in \mathcal{N}_{\text{inv}}$. If $\boldsymbol{\psi}[\mathbf{z}]$ is some spatial domain profile, the local Inverse Gabor Transform is defined as

$$\boldsymbol{\psi}_{\text{local}}[\mathbf{z}] = \sum_{n \in \mathcal{N}_{inv}, m \in \mathbf{m}} \mathbf{g}_{n,m}[\mathbf{z}] \phi_{n,m}, \tag{0.28}$$

where $\boldsymbol{\psi}_{local}[\mathbf{z}] = \sum_{n \in \mathcal{N}_{\text{inv}}} \boldsymbol{\psi}[\mathbf{z}] \mathbf{s}_n[\mathbf{z}]$, and the Gabor coefficient spectrum $\boldsymbol{\Phi}$ has elements $\phi_{n,m} = \left\langle \boldsymbol{\psi}[\mathbf{z}], \mathbf{g}_{n,m}[\mathbf{z}] \right\rangle$. The procedure for updating the Gabor coefficient matrix $\boldsymbol{\Phi}$ is as follows: First, the spatial domain vector is constructed as per (0.28). Next, the spatial domain operation is applied to $\boldsymbol{\psi}_{\text{local}}[\mathbf{z}]$. The new Gabor-domain fields are then constructed by setting

$$\phi_{n,m} = \left\{ \begin{array}{cc} \left\langle \boldsymbol{\psi}_{\text{local}}[\mathbf{z}], \mathbf{g}_{n,m}[\mathbf{z}] \right\rangle, & n \in \mathcal{N}_{\text{inv}} \\ \phi_{n,m} + \left\langle \boldsymbol{\psi}_{\text{local}}[\mathbf{z}], \mathbf{g}_{n,m}[\mathbf{z}] \right\rangle, & n \in \mathcal{N}_{\text{fwd}} - \mathcal{N}_{\text{inv}} \end{array} \right\}. \tag{0.29}$$

A difficulty with this procedure is that the entire support of $\boldsymbol{\psi}_{\text{local}}[\mathbf{z}]$ must be stored, and extra time is needed in order to compute a forward Gabor transform over the set $\mathcal{N}_{fwd}$ which is larger than $\mathcal{N}_{inv}$. In order to describe a method that resolves this problem, it helps to re-define the Gabor Transform operations in terms of Fourier Transforms and linear algebra notation.

Consider a Gabor coefficient matrix $\boldsymbol{\Phi}[\mathbf{n},\mathbf{m}]$ corresponding to a spatial domain vector $\boldsymbol{\psi}[\mathbf{z}]$. The matrix can be represented in row-vector format as $\boldsymbol{\Phi}[\mathbf{n},\mathbf{m}] = [\phi_0[\mathbf{m}], \phi_1[\mathbf{m}], \ldots, \phi_{N-1}[\mathbf{m}]]^T$. Recall that the elements of the row vectors are computed as $\phi_n[\mathbf{m}] = \mathbf{g}[\mathbf{z} - na]e^{i(mb)(\mathbf{z}-na)}$. This expression is equivalently a Fourier Transform, given by

$$\phi_n[\mathbf{m}] = \text{FFT}^n\{\mathbf{g}[\mathbf{z} - na\Delta z]\boldsymbol{\psi}[\mathbf{z}]\}, \tag{0.30}$$

where

$$\text{FFT}^n\{\mathbf{f}[\mathbf{z}]\} = \text{FFT}\{\mathbf{f}[\mathbf{z}_n]\}, \tag{0.31}$$

with the local vector

$$\mathbf{z}_n = [(n-1)a\Delta z, (n-1)a\Delta z + \Delta z, \ldots, na\Delta z - \Delta z] \subset \mathbf{z} \tag{0.32}$$

corresponding to the support of the nth Gabor window function. Likewise, $\boldsymbol{\psi}[\mathbf{z}]$ can be reconstructed from the row vectors by using an Inverse Fourier Transform, given by

$$\boldsymbol{\psi}[\mathbf{z}] = \sum_{n=0}^{N-1} \tilde{\mathbf{g}}[\mathbf{z} - na\Delta z]\text{IFFT}^n\{\phi_n[\mathbf{m}]\}, \tag{0.33}$$

where

$$\text{IFFT}^n\{\hat{\mathbf{f}}[\mathbf{k}_z]\}[z] = \left\{ \begin{array}{cc} \text{IFFT}\{\hat{\mathbf{f}}[\mathbf{k}_z]\}[z - na\Delta z], & z \in \mathbf{z}_n \\ 0, & z \notin \mathbf{z}_n \end{array} \right\} \tag{0.34}$$

is a localized Inverse Fourier Transform written to the support of the nth Gabor window function.

Suppose a linear spatial domain operator is to be applied pointwise to $\boldsymbol{\psi}[z]$; this operator can be described by a diagonal matrix $\mathbf{D}$, where the aim is to obtain a representation for $\mathbf{D}\boldsymbol{\psi}[z]$.

The objective of this procedure is to efficiently formulate some Gabor coefficient set, expressed as a set of vectors $\mathbf{\Phi}_D[\mathbf{n},\mathbf{m}] = [\mathbf{h}_0[\mathbf{m}], \mathbf{h}_1[\mathbf{m}], \ldots, \mathbf{h}_{N-1}[\mathbf{m}]]^T$ from some operation on $\mathbf{\Phi}[\mathbf{n},\mathbf{m}]$, such that $\mathbf{D}\boldsymbol{\psi}[\mathbf{z}] = \mathrm{IGT}\{\mathbf{\Phi}_D[\mathbf{n},\mathbf{m}]\}$. Theorem 1.1 is the basis for the improved method.

Theorem 1.1: Consider a vector $\boldsymbol{\psi}[\mathbf{z}]$ and its Gabor coefficient matrix $\mathbf{\Phi}[\mathbf{n},\mathbf{m}] = [\phi_0[\mathbf{m}], \phi_1[\mathbf{m}], \ldots, \phi_{N-1}[\mathbf{m}]]^T$. If $\mathbf{D}$ is a diagonal matrix, then

$$\mathbf{D}\boldsymbol{\psi}[\mathbf{z}] = \sum_{n=0}^{N-1} \tilde{\mathbf{g}}[\mathbf{z} - na\Delta z]\mathrm{IFFT}^n\{\mathbf{h}_n[\mathbf{m}]\} \tag{0.35}$$

Where $\mathbf{h}_n[\mathbf{m}]$ is given by $\mathbf{h}_n[\mathbf{m}] = \mathrm{FFT}^n\{\mathbf{D}\mathrm{IFFT}^n\{\phi_n[\mathbf{m}]\}\}$.

Proof: Let

$$\boldsymbol{\psi}_c[\mathbf{z}] = \sum\nolimits_{n=0}^{N-1} \tilde{\mathbf{g}}[\mathbf{z} - na\Delta z]\mathrm{IFFT}^n\{\mathbf{h}_n[\mathbf{m}]\}, \tag{0.36}$$

where $\mathbf{h}_n[\mathbf{m}] = \mathrm{FFT}^n\{\mathbf{D}\mathrm{IFFT}^n\{\phi_n[\mathbf{m}]\}\}$. After substituting in the definition for $\mathbf{h}_n[\mathbf{m}]$, successive forward and inverse FFT operations cancel one another out and (0.36) reduces to

$$\boldsymbol{\psi}_c[\mathbf{z}] = \sum\nolimits_{n=0}^{N-1} \tilde{\mathbf{g}}[\mathbf{z} - na\Delta z]\mathbf{D}\,\mathrm{IFFT}^n\{\phi_{n_i}[\mathbf{m}]\}. \tag{0.37}$$

Since $\mathbf{D}$ is diagonal and linear, it commutes and distributes and thus it can be pulled out in front of the summation, yielding

$$\boldsymbol{\psi}_c[\mathbf{z}] = \mathbf{D}\sum\nolimits_{n=0}^{N-1} \tilde{\mathbf{g}}[\mathbf{z} - na\Delta z]\mathrm{IFFT}^n\{\phi_n[\mathbf{m}]\}. \tag{0.38}$$

And so by (0.33), $\boldsymbol{\psi}_c[\mathbf{z}] = \mathbf{D}\boldsymbol{\psi}[\mathbf{z}]$. ■

Theorem 1.1 can be regarded as a slight variation on a conventional Inverse/Forward Gabor Transform procedure. Theorem 1.1 shows that a spatial-domain function multiplied with a diagonal operator (such as a phase screen) $\mathbf{D}\boldsymbol{\psi}[\mathbf{z}]$ is equivalent to $\boldsymbol{\psi}_c[\mathbf{z}]$, which has a Gabor coefficient set $\mathbf{h}_n[\mathbf{m}]$ that can be computed with a series of localized FFTs/IFFTs rather than a full Inverse and Forward Gabor Transform.

In other words, rather than computing a series of Inverse Fourier Transforms and multiplying each transformed vector by a window function before summing together, as in

equation (0.33), the window function and summation step is skipped and the pointwise operator $\mathbf{D}$ is applied to each individual vector. By avoiding summing vectors together, these individually transformed and updated vectors can be immediately Fourier Transformed back into the Gabor domain, without need for an expensive forward Gabor Transform operation given by (0.30). This procedure is only possible because of $\mathbf{D}$ being a linear and diagonal operator. There are three performance improvements yielded by this procedure: 1) No time is spent multiplying vectors by window functions, 2) only one vector of length N_p has to be stored at a time, and 3) the extra steps caused by $\mathcal{N}_{\text{inv}}$ being a proper subset of $\mathcal{N}_{\text{fwd}}$ are avoided.

Theorem 1.1 can be used to efficiently update a Gabor coefficient set in-place. Suppose that $\mathbf{\Phi}_0$ is the Gabor coefficient set for $\boldsymbol{\psi}_0[\mathbf{z}]$, and that some diagonal linear operator $\mathbf{D}$ is to be locally applied to generate $\mathbf{D}\boldsymbol{\psi}_0[z]$, where the diagonal of $\mathbf{D}$ has support such that $\mathbf{D}\boldsymbol{\psi}_\mathbf{0}[\mathbf{z}] = \sum_{n\in\mathcal{N}_{\text{inv}}} \left(\mathbf{D}\boldsymbol{\psi}_\mathbf{0}[\mathbf{z}]\right) \odot \mathbf{s}_n[\mathbf{z}]$, where $\odot$ is the pointwise product. The revised update procedure, for a diagonal matrix $\mathbf{D}$, is given by

$$\phi_n[\mathbf{m}] = \begin{Bmatrix} \text{FFT}^n\{\mathbf{D}\text{IFFT}^n\{\phi_n[\mathbf{m}]\}, & n \in \mathcal{N}_{inv} \\ \phi_n[\mathbf{m}], & n \notin \mathcal{N}_{inv} \end{Bmatrix}. \tag{0.39}$$

D. Radiation Boundary Conditions

This section discusses Radiation Boundary Conditions on the top edge of the domain, where upward-traveling radiation is expected to propagate forever into the sky without reflection. A problem inherent to spectral propagators such as Split-Step Fourier is that the Fourier Transform enforces periodic boundary conditions. Since Gabor propagation is based on SSF, this issue persists. Many existing methods to mitigate this issue and absorb outgoing radiation are expensive or difficult to implement, but a Gabor-domain RBC can be implemented by simply deleting excited Gabor coefficients that escape the domain.

Consider a field profile $\mathbf{\Phi}[x_0, \mathbf{n}, \mathbf{m}]$ in the Gabor domain with elements $\phi_{n,m}[x_0]$, that is to be marched forward to produce $\mathbf{\Phi}[x_0 + \Delta x, \mathbf{n}, \mathbf{m}]$ with elements $\phi_{n,m}[x_0 + \Delta x]$. Consider an arbitrarily large auxiliary domain, where Gabor coefficients are allowed to spread over $-\infty < n < \infty$. Let ${\phi^{\text{aux}}}_{n,m}[x_0 + \Delta x]$ be the fields marched from $\phi_{n,m}[x_0]$ with (0.26) according to this

new domain. The resultant fields are bounded to the true domain by simply deleting coefficients that escape $0 \le n \le N-2$ on the auxiliary domain. More precisely,

$$\phi_{n,m}[x_0 + \Delta x] = \begin{Bmatrix} \phi^{\text{aux}}{}_{n,m}[x_0 + \Delta x], & 0 \le n \le N-2 \\ 0, & \text{otherwise} \end{Bmatrix}. \tag{0.40}$$

$N-2$ is chosen as the upper bound rather than $N-1$, because the spatial support for $\mathbf{g}_{N-1,m}[\mathbf{z}]$ wraps around the end of the z domain; this is due to the overlapping nature of the Gabor window functions.

This implementation of Radiation Boundary Conditions is extremely memory-efficient. No storage space inside the domain is required for its implementation. While the marching operation makes use of an auxiliary domain, such an auxiliary domain doesn't need to be stored in practice; before each addition in (0.26) is executed, a check can be performed on whether the fields will escape the true domain. If they are found to be out of bounds, the addition is not carried out and no resources are consumed except for the bounds checking.

There are two sources of error associated with this method. One is that, due to Gabor window functions overlapping in the spatial domain, coefficients corresponding to $n = 0$ and $n = N-2$ will be slightly inaccurate since they neighbor the deleted indices. Secondly, a reflection is introduced by the hard truncation. This reflection is larger for shallow angles of incidence, and smaller as the angle of incoming radiation approaches the normal direction to the boundary. Since conventional absorbing layers have the opposite relationship between absorption and incident angle, a practical solver can hybridize both a thin absorbing layer and Gabor RBCs, enabling each method to overcome the shortcomings of the other. The absorbing layer can be implemented via local Inverse Gabor Transforms as per (0.39).

1.5 Phase Screens in the Gabor Domain

This section discusses Gabor propagation through inhomogeneous space. In conventional Split-Step Fourier, the propagator assumes free space; inhomogeneities must be handled in the spatial domain. This is achieved with phase screens, as shown in (0.4). Since the phase screens are spatially applied, repeated Fourier Transforms must be computed to convert between the spatial domain for phase screens and the frequency domain for propagation, and this is the main performance bottleneck of Split-Step Fourier.

Much like the Fourier domain, the phase screen operation has no direct analog in the Gabor domain, since each Gabor atom corresponds to wide, overlapping spatial domain coverage. One approach is to apply phase screens in the spatial domain, which is exact, but requires expensive forward/inverse transforms and uses excessive resources in storing the spatial domain. In other implementations of wavelet propagators, the refractive index profile is sampled in and directly applied in the Gabor domain.

This section will improve upon existing methods for phase screens in wavelet-based propagation by modifying the propagation matrix set to directly store atmospheric information. One possible approach to this is to generate a propagation matrix for each spatial bin s and store the phase screen associated with such bins inside the propagation matrix, but this would require precomputing the propagation characteristics of $N * N_p$ fields which is costly in time and memory. In the interest of computing only one propagation matrix set and recycling it for all s, simplifications must be made.

Consider the simplest smooth inhomogeneous atmosphere, a linear profile $\mathbf{n}[\mathbf{z}] = \alpha\mathbf{z}$, where α is some small positive constant. Since the derivative $\mathbf{n}'[\mathbf{z}] = \alpha$ is a constant, every element of a wavefront $\boldsymbol{\psi}[x_0, \mathbf{z}]$ will refract in exactly the same manner regardless of height when marched through space. This property can be exploited; a phase screen that refracts according to a linear profile can be built into a single propagation matrix set.

A. Propagation Matrix with Built-in Phase Screen

The (s, f) element of the propagation matrix set for an arbitrary refractive index profile can be written as

$$\mathbf{P}[s, f, \mathbf{n}, \mathbf{m}] = \mathrm{GT}\{\mathrm{IFFT}\{\mathbf{b}_{s,f}[0, \mathbf{k}_\mathbf{z}] e^{i\mathbf{k}_x[\mathbf{k}_\mathbf{z}]\Delta x} \cdot e^{ik(\mathbf{n}[\mathbf{z}]-1)\Delta x}\}\}, \tag{0.41}$$

where $\mathbf{b}_{s,f}[0, k_z]$ is the Fourier Transform of a beamlet shifted and modulated by (s, f) at $x = 0$. The linear refractive index profile $\mathbf{n}[\mathbf{z}] = \alpha\mathbf{z}$ is substituted in to (0.41) to produce

$$\mathbf{P}_\alpha[s, f, \mathbf{n}, \mathbf{m}] = \mathrm{GT}\{\mathrm{IFFT}\{\mathbf{b}_{s,f}[0, \mathbf{k}_\mathbf{z}] e^{i\mathbf{k}_x[\mathbf{k}_\mathbf{z}]\Delta x} \cdot e^{ik(\alpha\mathbf{z}-1)\Delta x}\}\}, \tag{0.42}$$

where $\mathbf{P}_\alpha[s, f, \mathbf{n}, \mathbf{m}]$ is $\mathbf{P}[s, f, \mathbf{n}, \mathbf{m}]$ generated according to a linear refractive index profile with slope α. The objective is to reformulate $\mathbf{P}_\alpha[s, f, \mathbf{n}, \mathbf{m}]$ in terms of a runtime operation on $\mathbf{P}_\alpha[1, f, \mathbf{n}, \mathbf{m}]$, such that only M matrices of size $N \times M$ need to be generated and stored for a linear refractive index profile.

The phase screen term in (0.42) can be rearranged and expanded to produce

$$e^{ik(\alpha \mathbf{z}-1)\Delta x} = e^{ik(\alpha(\mathbf{z}-(s-1)a\Delta z)-1)\Delta x} e^{ik\alpha(s-1)\Delta x}. \tag{0.43}$$

As the term $e^{ik\alpha(s-1)\Delta x}$ is only a function of s, it can be pulled out of the Gabor Transform operator. At $s=1$, (0.42) combined with (0.43) simplifies to

$$\mathbf{P}_\alpha[1, f, \mathbf{n}, \mathbf{m}] = \mathrm{GT}\{\mathrm{IFFT}\{\mathbf{b}_{1,f}[0, \mathbf{k}_\mathbf{z}] e^{i\mathbf{k}_x \Delta x}\} \cdot e^{ik(\alpha \mathbf{z}-1)\Delta x}\}. \tag{0.44}$$

By exploiting a shifting property of the Gabor frame used in this thesis

$$\mathbf{\Phi}[\mathbf{n} - n_0, \mathbf{m}] = \mathrm{GT}\{\boldsymbol{\psi}[\mathbf{z} - n_0 a\Delta z]\}, \tag{0.45}$$

(0.44) can be rewritten as

$$\mathbf{P}_\alpha[s, f, n, m] = \mathbf{P}_\alpha[1, f, (n-s)+1, m] e^{ik\alpha(s-1)\Delta x}. \tag{0.46}$$

So for a linear refractive index profile with slope α, an initial propagation matrix set is precomputed as (0.44), and (0.46) is used at runtime to compute the propagation matrix for any spatial bin s in terms of a shift and a multiplication of (0.44). Using this propagation matrix set on the propagation operator (0.26) is an exact representation of fields propagating according to a linear refractive index profile.

B. Refractive Index Profiles with Nonuniform Slope

The next complication arises in the case of a bilinear or trilinear atmospheric profile, as is common in ducts over oceans or in piecewise linear approximations to smooth profiles. When marching fields far away from corners in the refractive index profile, the process outlined in (0.44) and (0.46) can be repeated, however a new set of propagation matrices must be computed for each unique slope in the refractive index.

Near corners, the method derived in section A is not valid, therefore corners must be treated separately. The most effective option is to simply perform a local inverse Gabor Transform over

only the spatial bins that have a corner in their regions of support, apply the exact phase screen in the spatial domain, and transform back to the Gabor domain. The pitfalls of spatial domain processing are mitigated by the localization of the inverse transforms.

The procedure for marching through space with a piecewise linear refractive index is as follows. First, let N_s be the number of unique slopes in the piecewise linear refractive index profile $\mathbf{n}[\mathbf{z}]$, and let these unique slopes be denoted by $\alpha_1, \alpha_2, ..., \alpha_{N_s}$. Once all of the slopes are determined, propagation matrices $\mathbf{P}_{\alpha_1}, \mathbf{P}_{\alpha_2}, \ldots, \mathbf{P}_{\alpha_{N_s}}$ are generated for each slope as per (0.44).

Each vertical point $z \in \mathbf{z}$ is associated with the refractive index slope at that point, and each Gabor atom is assigned a slope based on which point in $\mathbf{z}$ it lines up with. The spatial location is given by $z_{winloc}(n) = na\Delta z$ where n is the spatial Gabor index. The equation mapping a Gabor atom index (n,m) to a slope α is given by

$$\alpha(n) = \mathbf{n}'[na\Delta z], \tag{0.47}$$

provided $\mathbf{n}'[z] \in \{\alpha_1, \alpha_2, \ldots \alpha_{N_s}\}$ for all $z \in \mathbf{z}$.

Equation (0.26) is modified to

$$\phi_{n,m}(x_0 + \Delta x) = \sum_{s,f} \phi_{s,f}(x_0) \mathbf{P}_{\alpha(n)}[s, f, n, m], \tag{0.48}$$

where $\mathbf{P}_{\alpha(n)}[s, f, n, m]$ is computed from $\mathbf{P}_{\alpha(n)}[1, f, n, m]$ at runtime according to (0.46).

Once all Gabor coefficients are marched, corners must be treated. Let $\mathcal{N}_c$ be the set of all spatial bin indices n where a corner exists inside their regions of spatial support. A local inverse Gabor transform is performed, modified such that the Gabor-domain phase screens are locally undone in the process,

$$\boldsymbol{\psi}_{\text{local}}[x + \Delta x, \mathbf{z}] = \sum_{n \in \mathcal{N}_c, m} (\mathbf{g}_{n,m}[\mathbf{z}] \phi_{n,m}[x + \Delta x] \cdot e^{-ik_0 \Delta x (\alpha(n) \cdot (\mathbf{z} - na\Delta z) - 1)} e^{-ik\Delta x \alpha(n) \cdot (n-1)}). \tag{0.49}$$

Once the local inverse transform is complete, a spatial-domain phase screen is applied, given by

$$\boldsymbol{\psi}_{local}[x + \Delta x, z] = \boldsymbol{\psi}_{local}[x + \Delta x, z] e^{ik_0 \Delta x (\mathbf{n}[z] - 1)}, \tag{0.50}$$

which exactly represents the effect of $\mathbf{n}[z]$ on propagation. A forward partial Gabor transform is then applied, and the coefficients are written back into $\mathbf{\Phi}[x_0 + \Delta x, n, m]$ as per (0.29).

The procedure can be easily modified to follow (0.39) as well, and results in a substantial performance improvement when implemented. With this modification, the update formula is

$$\phi_n[\mathbf{m}] = \begin{Bmatrix} \mathrm{FFT}^n\{\mathbf{p}[\mathrm{n},\mathbf{z}_n]\mathrm{IFFT}^n\{\phi_n[\mathbf{m}]\}, & n \in \mathcal{N}_c \\ \phi_n[\mathbf{m}], & n \notin \mathcal{N}_c \end{Bmatrix}, \tag{0.51}$$

where $\mathbf{p}[n, \mathbf{z}_n] = e^{-ik_0\Delta x(\alpha(n)\bullet(\mathbf{z}_n - na\Delta z) + \alpha(n)\bullet(n-1) - \mathbf{n}[\mathbf{z}_n])}$.

1.6 Numerical Results

This section shows the numerical implementation of Gabor frame-based propagation in 2D.

A. Sparsification Error

Since the Gabor coefficient set is thresholded at regular intervals, there is an associated error that grows as the sparsification operation is applied repeatedly. A simple scenario of a Gaussian marching through free space is studied in order to characterize the behavior of such inaccuracy. In Fig. 1.6a, a scenario is defined where a 300 MHz Gaussian beam of beam waist 15 m is marched through free space with two schemes: One with Split-Step Fourier, another with the new Gabor-domain propagator. The Gabor propagator is sparsified according to the procedure outlined in section 1.6. Seven scenarios are simulated, each with a different sparsification threshold τ. One such scenario is visualized in Fig. 1.6b. At each step in x, the error is computed, defined by

$$E(x) = \left\| \boldsymbol{\psi}_0(x)[\mathbf{z}] - \boldsymbol{\psi}_\tau(x)[\mathbf{z}] \right\|_2 / \left\| \boldsymbol{\psi}_0(x)[\mathbf{z}] \right\|_2. \tag{0.52}$$

Where $\boldsymbol{\psi}_\tau$ is $\boldsymbol{\psi}(x)[\mathbf{z}]$ with thresholding applied and $\boldsymbol{\psi}_0$ is $\boldsymbol{\psi}(x)[\mathbf{z}]$ with no thresholding. The domain is discretized with $\Delta x = 100\,\text{meters}$, $N_x = 50$, $\Delta z = 0.501\lambda$, and $N_z = 1024$. An absorbing layer is active with a 2 km thick error function window tuned such that a beam incident at 0.5 degrees from horizontal is attenuated by 40 dB. In the Gabor method, RBCs are active, and $N_p = 64$.

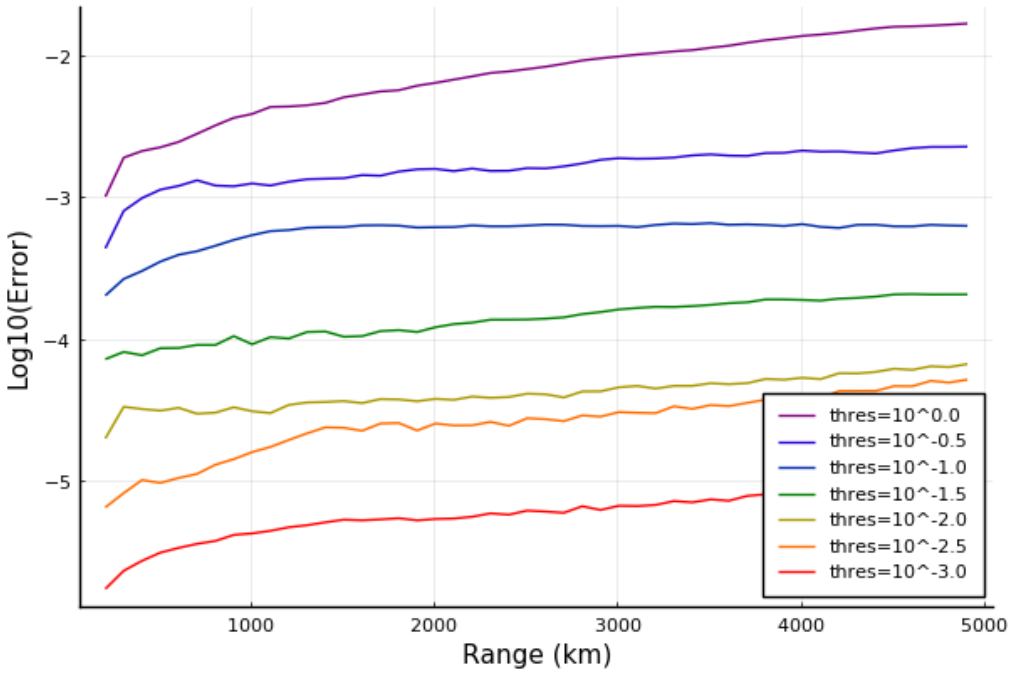


Fig. 1.6. Error of a propagating Gaussian beam, measured at different sparsification threshold levels.

Fig. 1.6 shows that a higher sparsification threshold corresponds to a lower error, as a higher threshold means less coefficients are used to represent a signal. The relative benefit from decreasing the threshold, and therefore increasing the number of stored coefficients, diminishes in this case once the tolerance reaches $10^{-1.5}$. Another observed behavior is that after a dramatic increase in the first 500 meters, the error remains nearly flat over all space. In larger-scale problems, this is similarly observed; sparsification error tends to remain relatively stable in 2D atmospheric propagation.

B. Radiation Boundary Conditions

The Gabor-domain Radiation Boundary Conditions are tested in isolation. Efficacy of an RBC implementation can be determined by launching a plane wave towards the boundary and measuring the amount of energy that remains inside the domain once the beam is expected to have fully crossed through the boundary.

To approximate a plane wave, a Gaussian beam with a wide beamwidth is launched at the upper boundary for a range of angles of incidence between 0 and 10 degrees from horizontal. Three cases are studied: Gabor RBCs in isolation, a conventional Absorbing Layer in isolation, and then a hybrid Gabor RBCs + Absorbing Layer. The relationship between the reflection coefficient and

the incident angle is shown in Fig. 1.7; a smaller reflection coefficient indicates that the boundary is more effectively behaving like a radiation surface. The frequency is 300 MHz, the beam waist is 1 km, and the absorbing layer is a 2 km thick error function window tuned such that a beam incident at 0.5 degrees from horizontal is attenuated by 40 dB. The vertical domain size is 2^{16} wavelengths, with $\Delta z = 1\lambda$ and $\Delta x = 64000$. The Gabor window width is 128, and the threshold for sparsification is set to 10^{-5} multiplied by the 1-norm of the Gabor coefficient set; setting the sparsity threshold this low effectively nullifies the error due to sparsification.

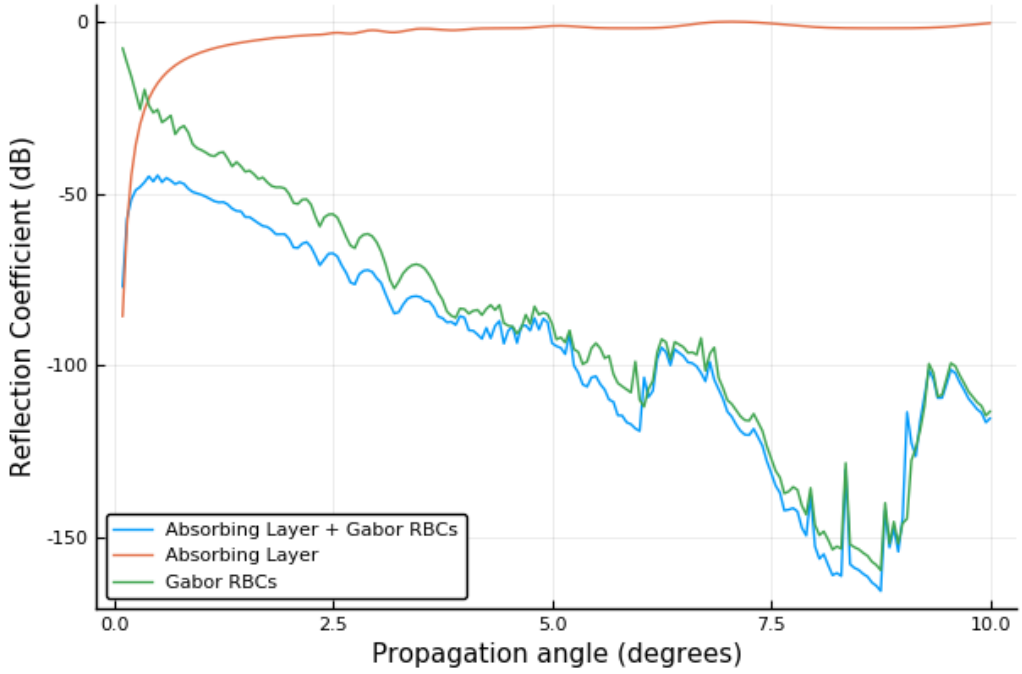


Fig. 1.7. Comparison of the reflection coefficient of three implementations of upper domain truncation (Gabor RBCs, Absorbing Layer, combined Gabor RBCs and Absorbing Layer), versus angle of incidence measured from horizontal.

Fig. 1.7 shows that Gabor RBCs reflect more radiation back into the domain as the angle of incidence is more shallow; however their absorption is very effective as the angle of incidence is steeper. An absorbing layer shows the opposite relationship; since incident radiation at a shallow angle travels a longer distance through the absorbing medium, the measured attenuation is higher than if it were at a steep angle and only traveled a negligible distance through the absorbing medium. When the two methods are hybridized, they work extremely well together as a radiation boundary condition that achieves minimal reflection for a broad range of incident angles.

C. Earth Curvature

SSF is based on a flat-Earth coordinate system; to correct for this, the curvature of the Earth can be modeled as an equivalent refractive index profile:

$$\mathbf{n}_{eq}[\mathbf{z}] = \mathbf{n}[\mathbf{z}] + \frac{\mathbf{z}}{r_e} \tag{0.53}$$

Where r_e is the radius of Earth in meters.

A Gaussian beam centered at 100 meters with waist 10 m, with a frequency of 8 GHz, is marched through an ionospheric profile with Earth curvature correction. Since the atmospheric profile is approximately linear, the phase screens can be entirely applied in the Gabor domain while maintaining accuracy. This facilitates extremely fast simulation time relative to conventional SSF. The ionospheric profile is given by

$$\mathbf{n}[\mathbf{z}] = 315 \times 10^{-6} \exp(-0.136\mathbf{z}). \tag{0.54}$$

The domain parameters are $\Delta x = 100$ meters, $\Delta z = 0.501\lambda$, $N_z = 32768$, $N_x = 500$, and an error function absorbing layer of height 123 meters tuned to attenuate incident radiation at 0.5 degrees by 200 dB. Gabor parameters include active RBCs, $N_p = 1024$, and sparsification threshold $\tau = 10^{-1}$. The error computed is defined as

$$E(x) = \frac{\left\| \boldsymbol{\psi}_{SSF}(x)[\mathbf{z}] - \boldsymbol{\psi}_{Gabor}(x)[\mathbf{z}] \right\|_2}{\left\| \boldsymbol{\psi}_{SSF}(x)[\mathbf{z}] \right\|_2}. \tag{0.55}$$

The field strength is visualized in Fig. 1.8, the stored field coefficients in one slice is shown in Fig. 1.9, and the number of stored field coefficients and error are shown in Fig. 1.10.

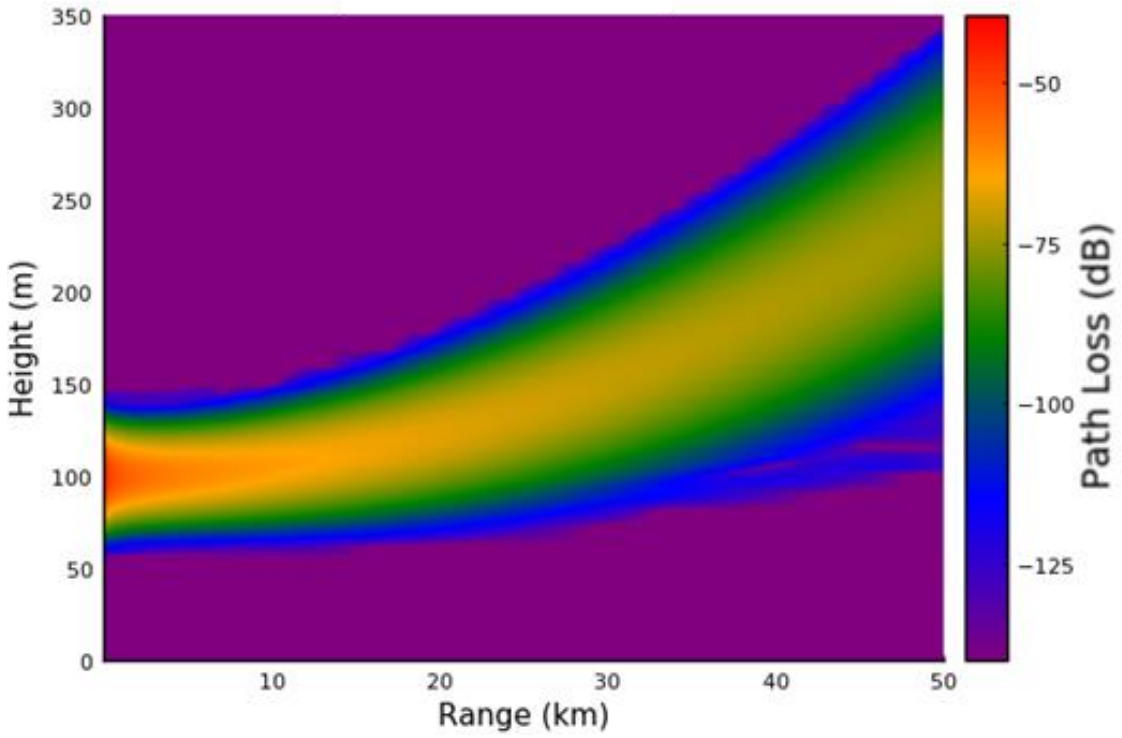


Fig. 1.8. Plot of path loss for a Gaussian beam marching through an ionosphere with Earth curvature correction

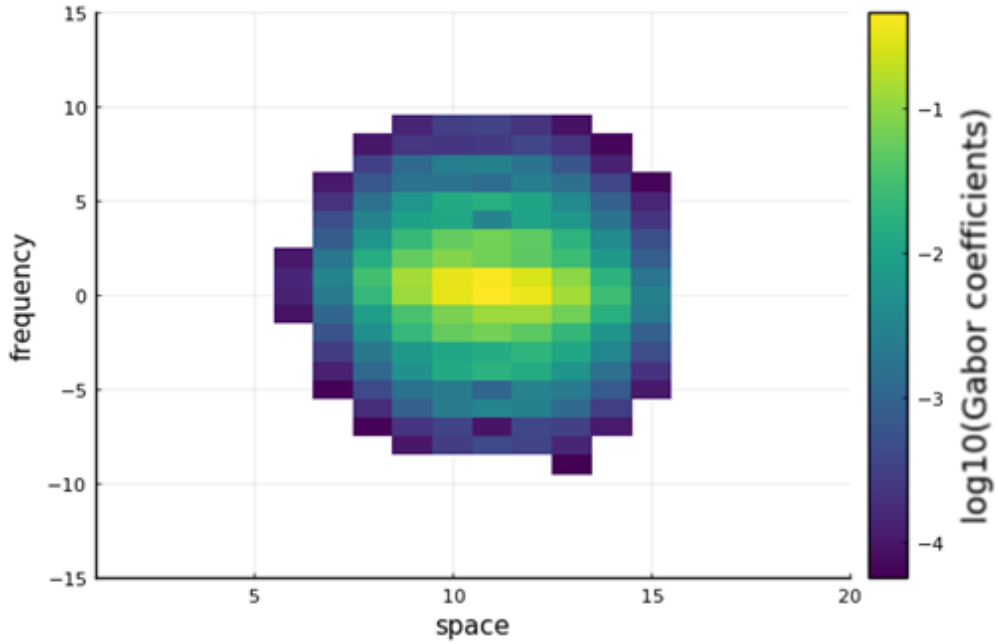


Fig. 1.9. Log-scale plot of the Gabor coefficient set at $x = 9950$ meters. Colored bins are stored coefficients, empty bins correspond to no stored data.

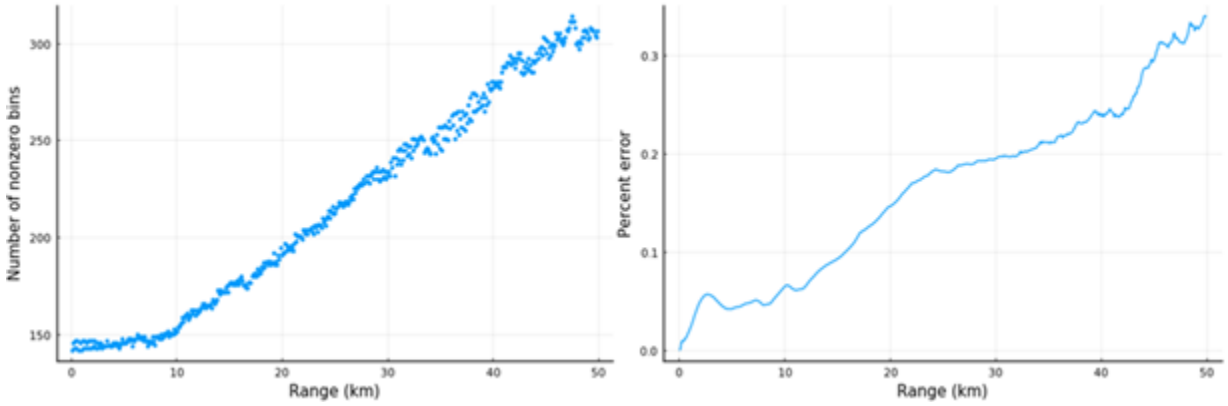


Fig. 1.10. Left: Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the ionosphere + Earth curvature case. Right: Plot of the error incurred by the Gabor method at each position in range for the ionosphere + Earth curvature case.

Solver Type	Average Step Time (ms)	Peak Memory Usage (# points)
Split-Step Fourier	4.8	32768
Gabor	0.535	310

Table 1.1. Comparison of CPU time and number of stored points for the simulation in Fig. 1.9 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated.

The Earth Curvature scenario was described extremely well by the Gabor method. Fields can be observed smoothly bending upwards in accordance with the curvature of the Earth in Fig. 1.8. Fig. 1.9 shows that most nonzero field coefficients are clustered around a small region. Resulting from this structure, Fig. 1.10 shows that fields were modeled with an error of approximately 0.3%, and at most only about 300 Gabor coefficients were excited, in contrast to SSF which used 32768 coefficients. The time taken to march through space and apply the phase screen was on average 4.80 ms per step for SSF, and the Gabor method took on average 0.535 ms per step to march and apply the phase screen, as shown in Table 1.1. This means that Gabor propagation was approximately nine times faster than SSF propagation.

D. Sparse Long-Range Propagation

The classic problem in Split-Step Fourier is that of marching over an ocean in the presence of an atmospheric duct. This procedure is implemented in the Gabor framework, exploiting Gabor-domain phase screens and sparsification in order to achieve faster performance than Split-Step Fourier.

A 60 GHz horizontally polarized Gaussian beam with a 2 meter waist centered at $z = 10$ meters is marched above an ocean, which is modeled as a perfect electric conductor. Image theory is used to reflect fields off of the ocean; the domain and initial wavefront are mirrored about $z = 0$, with a reflection coefficient of -1. Above the ocean exists an atmospheric duct, which in conjunction with Earth curvature correction, can be described as a trilinear refractive index profile visualized in Fig. 1.11.

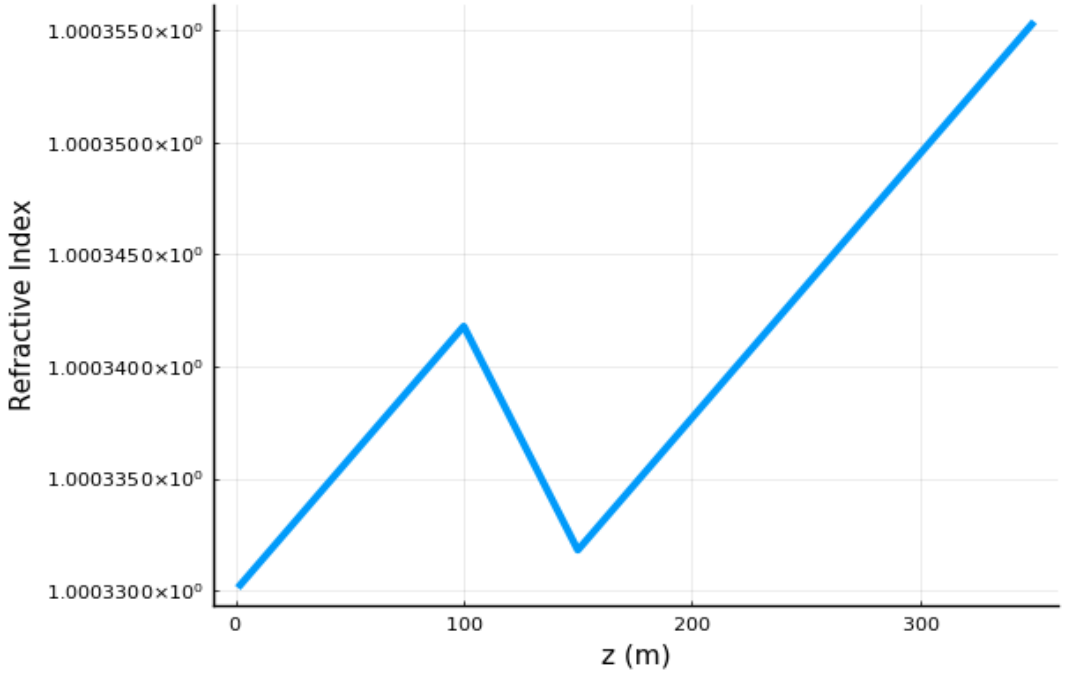


Fig. 1.11. Trilinear ocean refractive index profile.

Since the profile is not uniformly linear, partial inverse Gabor transforms are used about the changes in slope, as per section 1.8. When the refractive index profile is mirrored about $z = 0$ for imaging, the total profile has four unique slopes, so a propagation matrix set is generated for four different slopes. The domain is discretized with $\Delta x = 50$ meters, $\Delta z = 0.501\lambda$, $N_x = 2000$, $N_z = 524288$, and an error function absorbing layer of height 123 meters tuned to attenuate incident radiation at 0.5 degrees by 200 dB. The Gabor window width was set to $N_p = 2048$ with a sparsification threshold of $\tau = 10^{-0.8}$ multiplied by the 1-norm of the Gabor coefficient set.

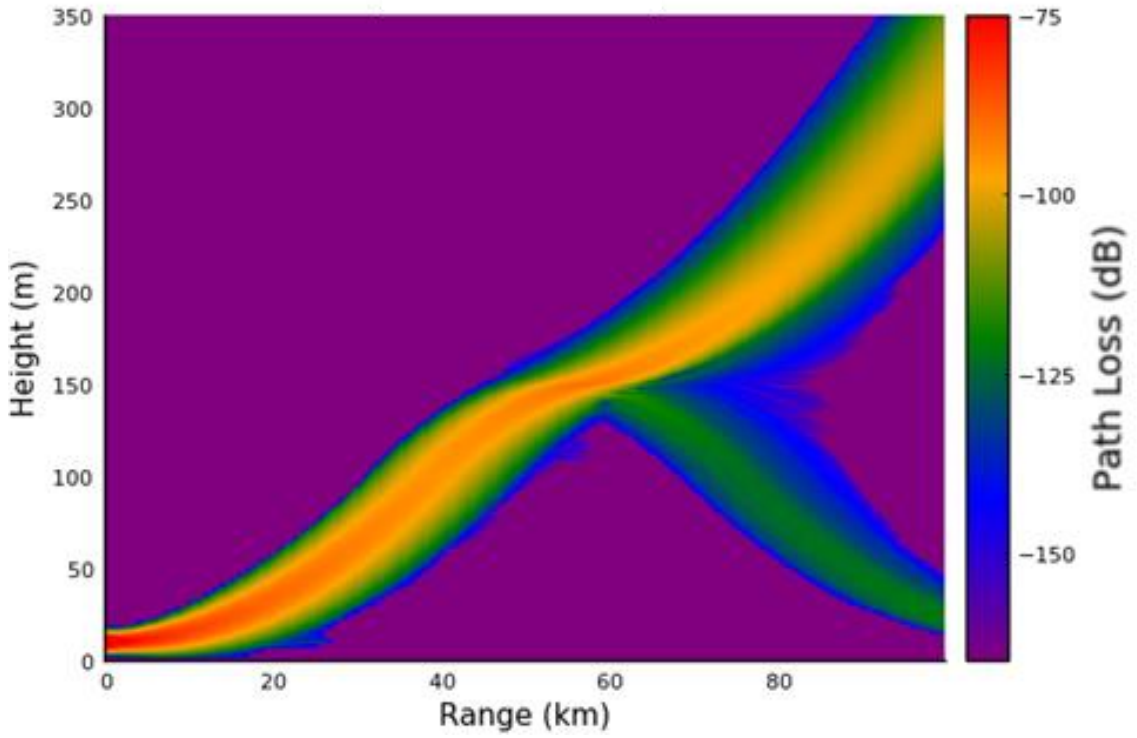


Fig. 1.12. Plot of path loss for a Gaussian marching through an ionosphere with Earth curvature correction.

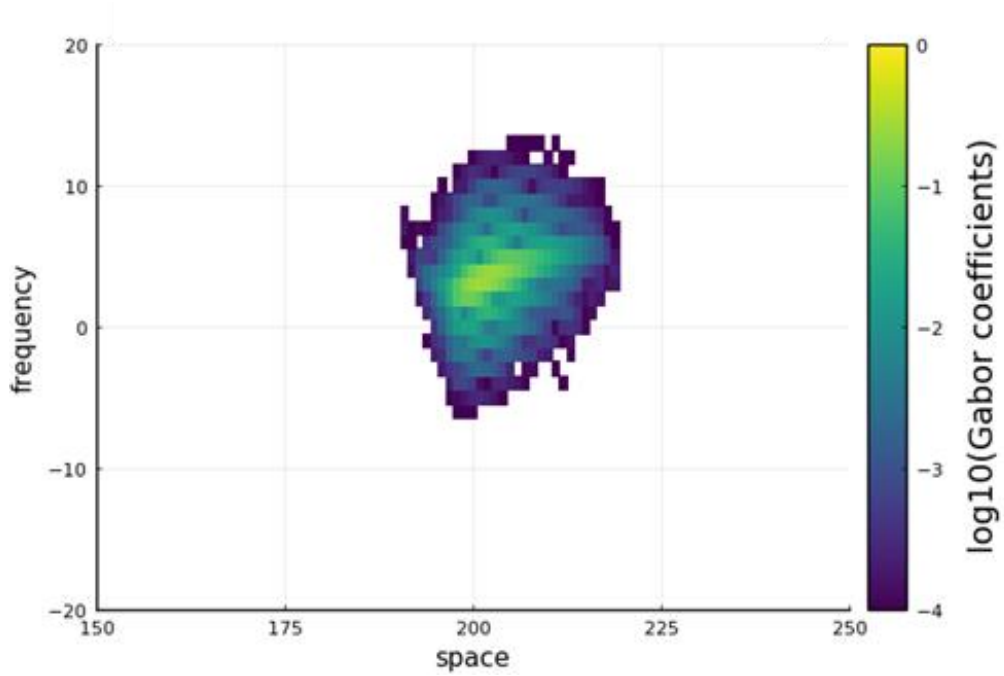


Fig. 1.13. Log-scale plot of the Gabor coefficient set at $x = 49550$ meters. Colored bins are stored coefficients, empty bins correspond to no stored data.

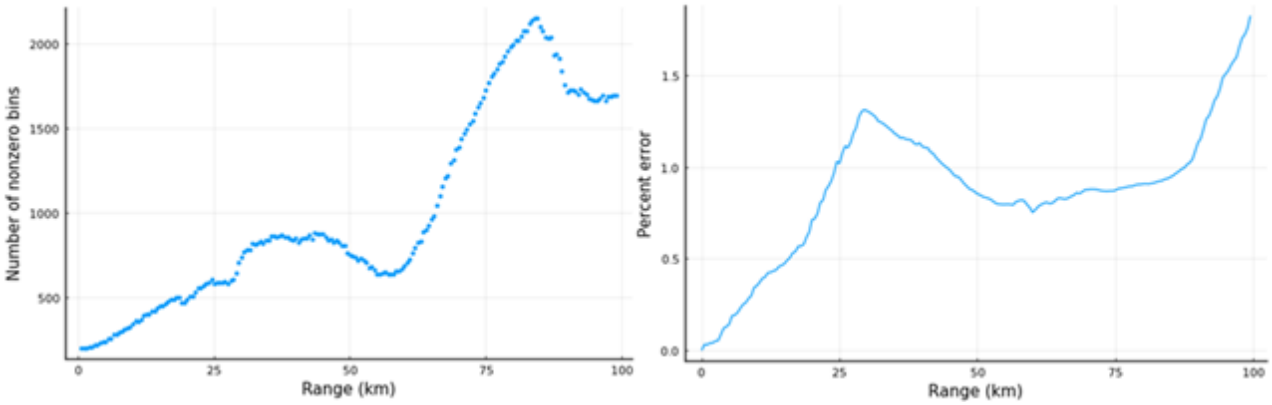


Fig. 1.14. Left: Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the ionosphere + Earth curvature case. Right: Plot of the error incurred by the Gabor method at each position in range for the ionosphere + Earth curvature case.

Solver Type	Average Step Time (ms)	Peak Memory Usage (# points)
Split-Step Fourier	82.6	524288
Gabor	24.5	2200

Table 1.2. Comparison of CPU time and number of stored points for the simulation in Fig. 1.12 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated.

Fig. 1.12 shows a beam smoothly propagating through a duct above an ocean. While some radiation deflects upwards due to the curvature of the Earth, other radiation gets trapped inside the duct. As propagation inside the duct is beamlike, the wavefront can be represented in an extremely sparse manner with negligible error. In Fig. 1.13, at a particular field slice at about 50 km, the sparse field data is all clustered about a small region in the Gabor space-frequency plane. This is further shown in Fig. 1.14, where the error remains below 2% while the number of stored field coefficients is bounded by about 2000. The time taken to march through space and apply the phase screen was on average 82.6 ms per step for SSF, and the Gabor method took on average 24.5 ms per step to march and apply the phase screen, as shown in Table 1.2. This means that Gabor propagation was approximately 3.4 times faster than SSF propagation. The fast computation time and accurate modeling of fields marching through complex atmosphere indicates that the phase screen method works as intended; fields are refracted without performing expensive full-domain inverse and forward Gabor Transforms.

Chapter 2. Adaptive Multiresolution Gabor Transforms for Optimal Sparse Field Representations

2.1 Introduction

Wave propagation scenarios involve complex interactions with environments and atmosphere. Excitations from transmitters themselves even may be complex. Such complexities induce fields that may have steep gradients in some locations and shallow gradients in other locations. This creates a problem for discretizing numerical problems, as a one-size-fits-all solution is difficult to establish without sacrificing resources or accuracy.

Gabor frame-based propagation is limited in its ability to model multiscale fields. The window functions utilized by the Gabor Transform have the same width at every spatial and frequency index. Such uniformity may not be the optimal way to describe complex field profiles, as different window widths can encapsulate different features, and a field that is sparsely represented by a transform under one window width may not be sparsely described by a transform under a different window width. There exists a strong need for an adaptive Gabor transform that can accommodate multiscale features.

This chapter outlines an adaptive multi-window Gabor Transform to optimally sparsify field representations. The adaptive Multi-Gabor Transform is introduced, which partitions the Gabor domain into different regions, and the Gabor discretization of each region is locally picked for maximum sparsity. The propagation framework from chapter 1 is modified to accommodate the Multi-Gabor Transform.

2.2 Multiresolution Gabor Transform

This section recounts and adapts the work of [63] by defining a framework for a multiwindow and multilayered Gabor expansion of a signal, referred to as the Multi-Gabor transform. The Multi-Gabor transform is intended to achieve more sparse signal representations than the uniform Gabor transform by adaptively selecting the best Gabor discretizations for subsections of the space-frequency plane.

The multi-Gabor transform can be understood as a partitioning of the space-frequency plane into different regions. Each region will use a different Gabor window width. An example of such a partition is visualized in Fig. 2.1. A signal is represented in the multi-Gabor domain by locally applying a Gabor transform in each region; the multi-Gabor coefficient set can be regarded as a collocation of different Gabor coefficient sets covering different regions of the space-frequency plane, where there is one set for each unique window width. The window width of each local region is selected for optimal sparsity.

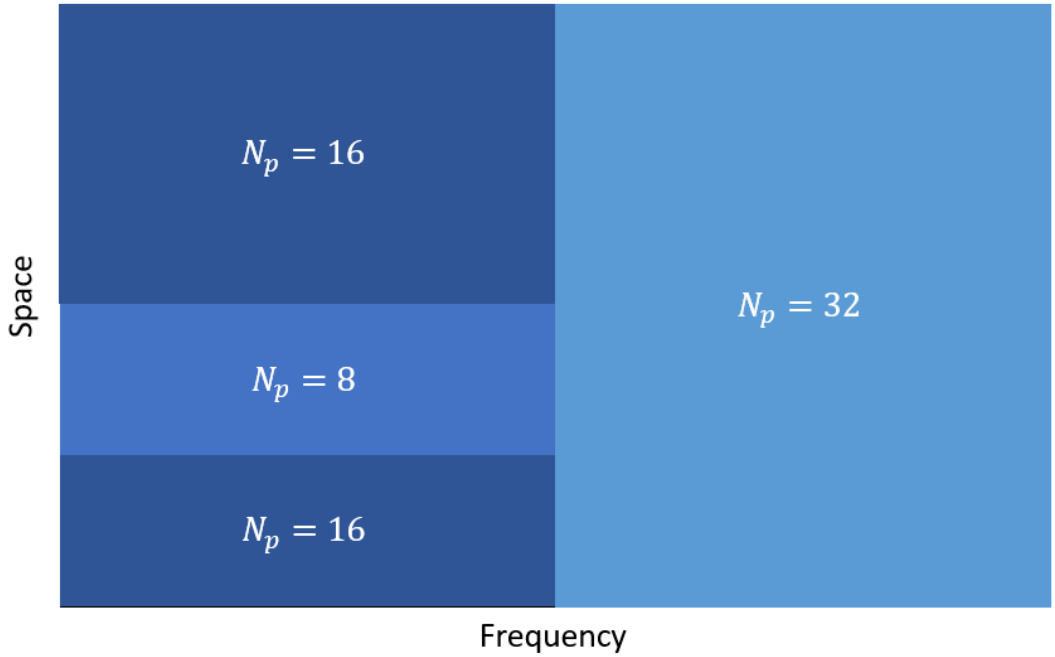


Fig. 2.1. Example of a partitioning of the Gabor space-frequency domain; different Gabor window widths, denoted by N_p, are used to model different regions.

In implementation, the Multi-Gabor framework is not a straightforward partition. The space-frequency plane canot be elegantly partitioned into non-overlapping regions due to complications with the mapping between the Gabor and spatial domains. The window functions used by the Gabor transform have a spatial overlap factor of $1/2$. In order for it to be possible to perfectly reconstruct a function from the information in a Gabor representation, the condition

$$\sum_{n=0}^{\mathbf{n}} \mathbf{g}_{n,0}{}^{2}[z] \geq 1 \tag{2.1}$$

must be satisfied for all $z \in \mathbf{z}$, where $\mathbf{n}$ is the set of all possible values for the Gabor spatial index n.

The overlap factor ensures completeness of the Gabor transform; however, the condition in (2.1) is not satisfied if $\mathbf{n}$ is partitioned into regions of different window widths for the corresponding $\mathbf{g}_{n,0}[z]$ functions. If two neighboring regions in the space-frequency plane use different window widths, there exists space between the regions that isn't fully supported by a window from either region. This is shown in Fig. 2.2.

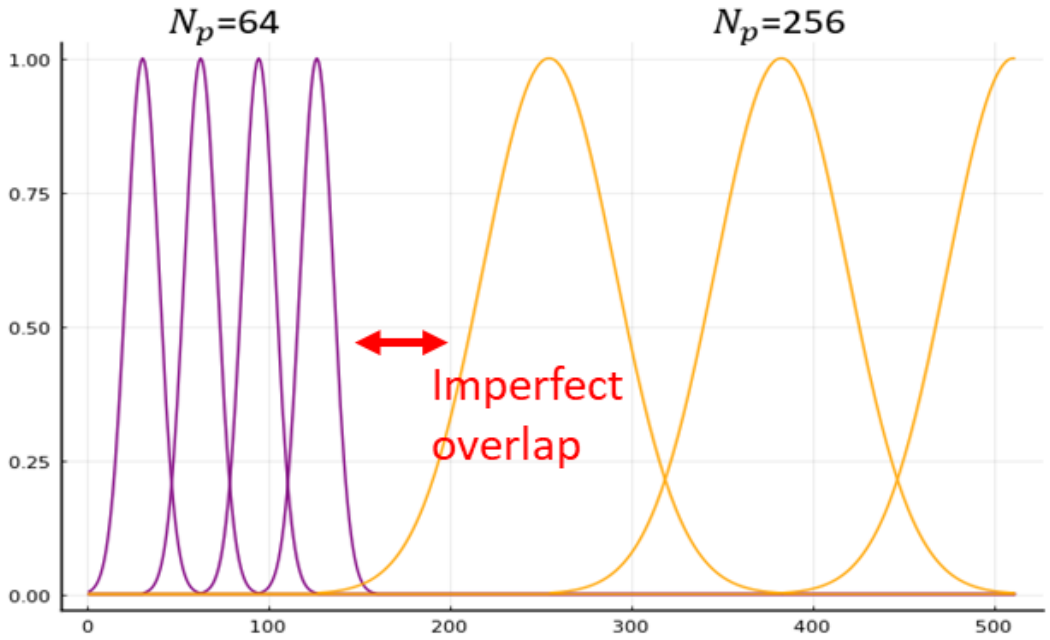


Fig. 2.2. Example of the Gabor window functions of two neighboring regions with window widths $N_p = 64$ and $N_p = 256$, showing insufficient overlap at the interface if the space-frequency plane were perfectly partitioned.

To resolve the overlap factor issue, the regions of different window widths are expanded and allowed to overlap slightly in the space-frequency plane, to ensure that the Multi-Gabor representation is overcomplete. With this correction, the actual Gabor space-frequency representation is treated as a collection of distinct space-frequency planes, all with overlapping spatial support in their regions of interest, shown in Fig. 2.3.

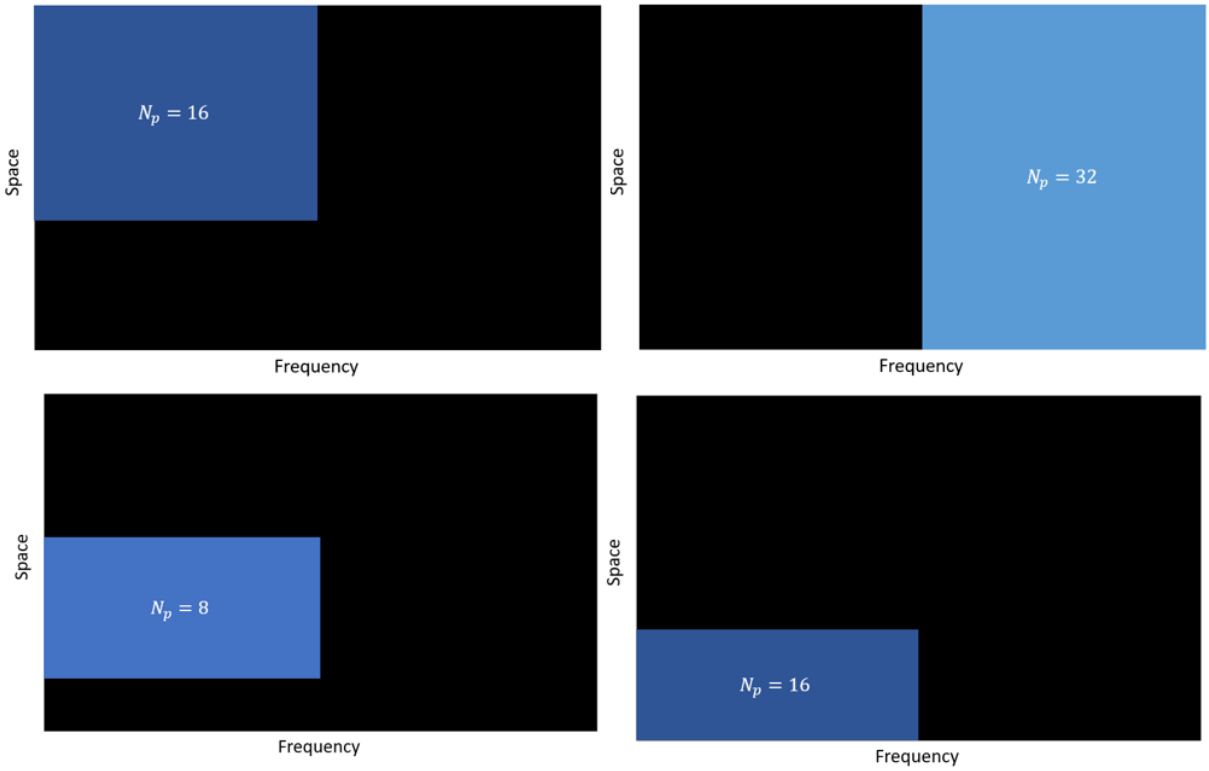


Fig. 2.3. Revision to Fig. 2.1, showing what the true multi-window divisions of the space-frequency plane look like; each of the four subplots shows the portion of the space-frequency plane covered by each window width. Note that such regions overlap; the support for the region $N_p = 8$ bleeds into the coverage of both of the neighboring $N_p = 16$ regions.

A formulaic description of the Multi-Gabor transform follows. The Jigsaw puzzle framework will be used to permit different Gabor window widths to represent different regions of the space-frequency plane. Let the Gabor-domain space-frequency plane be denoted by $\mathcal{P}^{N_{p0}} = \{n_0, m_0 : 0 \le n_0 \le N_0 - 1, 0 \le m_0 \le N_{p0} - 1\}$, associated with some reference window width N_{p0}, with $N_0 = 2N_z / N_{p0}$. Consider a collection of N_s different subsets of the space-frequency plane, where the ith subset is given by $(n_i, m_i) \in (\mathcal{N}^{N_{p0}}{}_i, \mathcal{M}^{N_{p0}}{}_i) \subseteq \mathcal{P}^{N_{p0}}$. The collection is constructed such that $\bigcup_{i=0}^{N_s-1} (\mathcal{N}^{N_{p0}}{}_i, \mathcal{M}^{N_{p0}}{}_i) = \mathcal{P}^{N_{p0}}$, and the sets $(\mathcal{N}^{N_{p0}}{}_i, \mathcal{M}^{N_{p0}}{}_i)$ do not necessarily need to be connected. For each set $(\mathcal{N}^{N_{p0}}{}_i, \mathcal{M}^{N_{p0}}{}_i)$, a local Gabor frame is chosen.

The construction $\bigcup_{i=0}^{N_s-1}(\mathcal{N}^{N_{p0}}{}_i, \mathcal{M}^{N_{p0}}{}_i) = \mathcal{P}^{N_{p0}}$ is defined for a fixed window width N_{p0}. Since this section is concerned with multi-window Gabor representations, the description of the discretization must account for varying window widths within different subsections of a field representation. If some N_{p1} is chosen as a power of 2 multiple of N_{p0}, then reference space-frequency domain $\mathcal{P}^{N_{p0}}$ can be subdivided or combined to form the new space-frequency domain $\mathcal{P}^{N_{p1}}$, and $(\mathcal{N}^{N_{p0}}{}_i, \mathcal{M}^{N_{p0}}{}_i)$ can be similarly transformed to be $(\mathcal{N}^{N_{p1}}{}_i, \mathcal{M}^{N_{p1}}{}_i)$ by letting $n_1 = n_0(N_{p0}/N_{p1}), m_1 = m_0(N_{p0}/N_{p1})$. To simplify notation, if $(\mathcal{N}_i, \mathcal{M}_i)$ is specified without a superscript, then the choice of N_p is left to be arbitrary.

Let the local frame be denoted by the window width N_p. N_p belongs to the set of all available window widths $\mathbf{N_p} = \{N_{p0}, N_{p1}, ..., N_{pNN_p}\}$. Let the set of Gabor coefficients associated with each $(\mathcal{N}_i, \mathcal{M}_i)$ be denoted by the matrix $\mathbf{\Phi}^{(\mathcal{N}_i, \mathcal{M}_i)}$.

Consider the collection of all Gabor coefficient sets over all subsets of the space-frequency plane denoted by $\mathbf{\Phi}_{\text{total}} = \{\mathbf{\Phi}^{(\mathcal{N}_0, \mathcal{M}_0)}[\mathbf{n}, \mathbf{m}], \mathbf{\Phi}^{(\mathcal{N}_1, \mathcal{M}_1)}[\mathbf{n}, \mathbf{m}], ..., \mathbf{\Phi}^{(\mathcal{N}_{N_s-1}, \mathcal{M}_{N_s-1})}[\mathbf{n}, \mathbf{m}]\}$. Recalling that the forward Gabor Transform operator is notated as $\mathbf{\Phi}[\mathbf{n}, \mathbf{m}] = \text{GT}\{\mathbf{f}[\mathbf{z}]\}$, let a local Gabor Transform over the spatial indices $n \in \mathcal{N}$ and frequency indices $m \in \mathcal{M}$ be written as $\mathbf{\Phi}^{(\mathcal{N}, \mathcal{M})}[n, m] = \begin{Bmatrix} \text{GT}\{\mathbf{f}[\mathbf{z}]\}[n, m], & (n, m) \in (\mathcal{N}, \mathcal{M}) \\ 0, & (n, m) \notin (\mathcal{N}, \mathcal{M}) \end{Bmatrix}$. The process of constructing $\mathbf{\Phi}_{\text{total}}$ is given by algorithm 2.1.

Algorithm 2.1. Elementary Multiresolution Gabor Transform

1. Set $\mathbf{f}_{\text{res}}[\mathbf{z}] = \mathbf{f}[\mathbf{z}]$
2. For $i = 0...N_s - 1$ do:
    a. Select N_{pi} for region i
    b. Set $\mathbf{\Phi}^{(\mathcal{N}^{N_{pi}}{}_i, \mathcal{M}^{N_{pi}}{}_i)}[\mathbf{n}, \mathbf{m}] = \text{GT}\{\mathbf{f}_{\text{res}}[\mathbf{z}]\}(\mathcal{N}^{N_{pi}}{}_i, \mathcal{M}^{N_{pi}}{}_i)$
    c. Set $\mathbf{f}_1[\mathbf{z}] = \text{IGT}\{\mathbf{\Phi}^{(\mathcal{N}^{N_{pi}}{}_i, \mathcal{M}^{N_{pi}}{}_i)}\}$
    d. Set $\mathbf{f}_{\text{res}}[\mathbf{z}] = \mathbf{f}[\mathbf{z}] - \mathbf{f}_1[\mathbf{z}]$
3. Set $\mathbf{\Phi}_{\text{total}} = \{\mathbf{\Phi}^{(\mathcal{N}^{N_{p0}}{}_0, \mathcal{M}^{N_{p0}}{}_0)}[\mathbf{n}, \mathbf{m}], \mathbf{\Phi}^{(\mathcal{N}^{N_{p1}}{}_1, \mathcal{M}^{N_{p1}}{}_1)}[\mathbf{n}, \mathbf{m}], ..., \mathbf{\Phi}^{(\mathcal{N}^{N_{p(N_s-1)}}{}_{N_s-1}, \mathcal{M}^{N_{p(N_s-1)}}{}_{N_s-1})}[\mathbf{n}, \mathbf{m}]\}$

Algorithm 2.1 iteratively computes a local Gabor transform of $\mathbf{f}[\mathbf{z}]$ on each region $(\mathcal{N}^{N_{pi}}{}_i, \mathcal{M}^{N_{pi}}{}_i)$, chipping away at intervals of the residual $\mathbf{f}_{\text{res}}[\mathbf{z}]$ with each region. The representation is exact if the window functions at the interface between neighboring regions $(\mathcal{N}^{N_{pi}}{}_i, \mathcal{M}^{N_{pi}}{}_i)$ overlap enough such that (2.1) is satisfied for $\mathbf{n} = \bigcup_{i=0}^{N_s-1} \mathcal{N}^{N_{pi}}{}_i$, however if the overlap is insufficient, then there will still exist some residual $\mathbf{f}_{\text{res}}[\mathbf{z}]$ left over after the algorithm is completed. Provided that for all $n \in \bigcup_{i=0}^{N_s-1} \mathcal{N}^{N_{pi}}{}_i$, there exists some $z \in \mathbf{z}$ such that $\mathbf{g}_{n,m}[z] \neq 0$, then algorithm 2.1 may be repeatedly computed on the remaining residual, adding each new set of Gabor coefficients to the results from the previous iteration of the algorithm. It may occur that the optimum Gabor window width set N_{pi} of the residual $\mathbf{f}_{\text{res}}[\mathbf{z}]$ is different than that of the initial function $\mathbf{f}[\mathbf{z}]$. Because of this, the concept of layers is introduced: Each region will have multiple layers, indexed by j, each with a corresponding window width $N_p{}^{i,j} \in \mathbf{N_p}$. Such regions will be denoted as $(\mathcal{N}_i, \mathcal{M}_i)^j$. A new layer is added for each pass of the algorithm; in other words, the number of layers corresponds to the number of iterations needed to minimize $\mathbf{f}_{\text{res}}[\mathbf{z}]$.

If some $\mathbf{\Phi}_{\text{total}}$ were constructed as a collection of layers, the data structure would have many different layers, regions, and window widths, and after the initial forward transform, keeping track of all of them is cumbersome and unnecessary. Rather, each Gabor coefficient can simply be grouped by its associated N_p. The structure and number of elements of $\mathbf{\Phi}_{\text{total}}$ can be simplified into a structure called $\mathbf{\Phi}_{\text{multi}}$, given by

$$\mathbf{\Phi}_{\text{multi}} = \{\mathbf{\Phi}^{N_{p1}}[\mathbf{n},\mathbf{m}], \mathbf{\Phi}^{N_{p2}}[\mathbf{n},\mathbf{m}], ..., \mathbf{\Phi}^{N_{pNNp}}[\mathbf{n},\mathbf{m}]\}, \tag{2.2}$$

where each $\mathbf{\Phi}^{N_{pk}}[\mathbf{n},\mathbf{m}] \in \mathbf{\Phi}_{\text{multi}}$ is the collection of all Gabor coefficients in the entire set of layers and regions that are associated with a window width N_{pk}. Note that if two overlapping layers or regions have the same N_{pk}, then the overlapping coefficients can be summed together to form $\mathbf{\Phi}^{N_{pk}}[\mathbf{n},\mathbf{m}]$.

The iterative process of iteratively building $\mathbf{\Phi}_{\text{multi}}$ by eliminating a residual layer-by-layer until $\|\mathbf{f}_{\text{res}}[\mathbf{z}]\| / \|\mathbf{f}[\mathbf{z}]\|$ falls below some prespecified tolerance tol is described in algorithm 2.2.

Algorithm 2.2. Multiresolution Gabor Transform

1. Set the residual $\mathbf{f}_{\text{res}}[\mathbf{z}] = \mathbf{f}[\mathbf{z}]$
2. Initialize Gabor coefficient sets $\mathbf{\Phi}^{N_p}[\mathbf{n},\mathbf{m}] \equiv 0$ for each $N_{pk} \in \mathbf{N_p}$
3. Initialize the layer index $j = 0$
4. While $\|\mathbf{f}_{\text{res}}[\mathbf{z}]\| / \|\mathbf{f}[\mathbf{z}]\| > \text{tol}$ do:
    a. For $i = 0...N_s - 1$ do:
        i. Pick an optimal $N_{pk} \in \mathbf{N_p}$ for the Gabor transform
        ii. Set $\mathbf{\Phi}^{\text{curr}}[\mathbf{n},\mathbf{m}] = \text{GT}\{\mathbf{f}_{\text{res}}[\mathbf{z}]\}(\mathcal{N}^{N_{pk}}{}_i, \mathcal{M}^{N_{pk}}{}_i)^j$
        iii. Set $\mathbf{f}_1[\mathbf{z}] = \text{IGT}\{\mathbf{\Phi}^{\text{curr}}[\mathbf{n},\mathbf{m}]\}$
        iv. Set $\mathbf{f}_{\text{res}}[\mathbf{z}] = \mathbf{f}[\mathbf{z}] - \mathbf{f}_1[\mathbf{z}]$
        v. Set $\mathbf{\Phi}^{N_{pk}}[\mathbf{n},\mathbf{m}] = \mathbf{\Phi}^{N_{pk}}[\mathbf{n},\mathbf{m}] + \mathbf{\Phi}^{\text{curr}}[\mathbf{n},\mathbf{m}]$
    b. Set $j = j + 1$
5. Set $\mathbf{\Phi}_{\text{multi}} = \{\mathbf{\Phi}^{N_{p1}}[\mathbf{n},\mathbf{m}], \mathbf{\Phi}^{N_{p2}}[\mathbf{n},\mathbf{m}], ..., \mathbf{\Phi}^{N_{pNNp}}[\mathbf{n},\mathbf{m}]\}$

It should be noted that if the chosen $N_p{}^{i,j}$ was the same for all subsets i of the space-frequency plane, then the resulting transform is equivalent to the uniform-window Gabor Transform.

With $\mathbf{\Phi}^{N_p}$ defined above, the inverse multiresolution Gabor transform is given by algorithm 2.3.

Algorithm 2.3. Inverse multiresolution Gabor transform

1. Initialize the spatial domain signal $\mathbf{f}[\mathbf{z}] \equiv 0$
2. For all $N_{pk} \in \mathbf{N_p}$ do:
    a. Set $\mathbf{f}[\mathbf{z}] = \mathbf{f}[\mathbf{z}] + \text{IGT}\{\mathbf{\Phi}^{N_{pk}}[\mathbf{n},\mathbf{m}]\}$

What remains is the process for choosing an optimal Gabor window width $N_p{}^{i,j}$ for each region and layer $(\mathcal{N}_i, \mathcal{M}_i)^j$. The representation should ensure maximal sparsity for some thresholding tolerance. Let $s(\mathbf{\Phi}^i[\mathbf{n},\mathbf{m}])$ be some function that computes the sparsity of $\mathbf{\Phi}^i[\mathbf{n},\mathbf{m}]$ for the ith

region and j th layer $(\mathcal{N}_i, \mathcal{M}_i)^j$. Let $\mathbf{N_p}$ be a list of all available Gabor window widths to choose from. The process for determining the optimal window width at layer j is given by algorithm 2.4.

Algorithm 2.4. Process for determining optimal Gabor representation

1. For $i = 0...N_{s-1}$ do
    a. Initialize $s_{best} = \infty$
    b. For $N_p{}^{cand}$ in $\mathbf{N_p}$ do
        i. Compute the Gabor Transform $\mathbf{\Phi}^{cand}[\mathbf{n},\mathbf{m}] = \mathrm{GT}\{\mathbf{f}_{\mathrm{res}}[\mathbf{z}]\}(\mathcal{N}^i \times \mathcal{M}^i)^j$ with Gabor window width $N_p{}^{cand}$
        ii. Threshold the Gabor coefficients $\mathbf{\Phi}^{cand}[\mathbf{n},\mathbf{m}]$ according to some tolerance specified by the problem statement
        iii. If $s(\mathbf{\Phi}^{cand}[\mathbf{n},\mathbf{m}]) < s_{best}$ do
            1. Set $\mathbf{\Phi}^i[\mathbf{n},\mathbf{m}] = \mathbf{\Phi}^{cand}[\mathbf{n},\mathbf{m}]$
            2. Set $N_p{}^i = N_p{}^{cand}$

The choice of sparsity measure $s(\mathbf{\Phi}^{cand}[\mathbf{n},\mathbf{m}])$ depends on the problem. The original paper [63] uses a Shannon Entropy measure. In the case of propagation with multi-Gabor systems, choosing a direct measure of the number of nonzero bins in some thresholded Gabor coefficient matrix tends to result in the sparsest representation for the lowest introduced error. The choice of $s(\mathbf{\Phi}^{cand}[\mathbf{n},\mathbf{m}])$ may also inform whether the thresholding of Gabor coefficients should be performed before or after the optimal discretization is found.

2.3. Multiresolution Gabor Propagation

This section describes how the multiresolution jigsaw puzzle method will be used to augment the traditional uniform-resolution Gabor method and increase total sparsity levels.

Consider a collection of subsets of the Gabor domain $(\mathcal{N}_i, \mathcal{M}_i)$ that are allowed to intersect; the union of all regions covers the entire Gabor domain. Consider an initial field profile $\boldsymbol{\psi}[x_0, \mathbf{z}]$. Let $\mathbf{\Phi}_{\mathrm{multi}}[x_0]$ be the optimum multiresolution Gabor transform of $\boldsymbol{\psi}[x_0, \mathbf{z}]$.

Multiresolution Gabor-domain propagation is an extension of the uniform-resolution case. For each unique value of $N_{pi} \in \mathbf{N_p}$, let $\mathbf{P}^{N_{pi}}[s, f, \mathbf{n}, \mathbf{m}]$ be the propagation matrix set generated according to N_{pi}. The expression for the propagation matrix set is

$$\mathbf{P}^{N_{pi}}[s,\, f\,, \mathbf{n}, \mathbf{m}] = \mathrm{GT}\{\mathbf{b}^{N_{pi}}{}_{s,f}[x_0 + \Delta x, \mathbf{z}]\}, \tag{2.3}$$

where $\mathbf{b}^{N_{pi}}{}_{s,f}[x_0 + \Delta x, \mathbf{z}]$ is computed from $\mathbf{b}^{N_{pi}}{}_{s,f}[x_0, \mathbf{z}]$ in free space using Split-Step Fourier, $\mathbf{b}^{N_{pi}}{}_{s,f}[x_0, \mathbf{z}] = \tilde{\mathbf{g}}^{N_p}_{s,f}[\mathbf{z}]$, and $\tilde{\mathbf{g}}^{N_p}_{s,f}[\mathbf{z}]$ is a Gabor dual window function with window width N_p, spatial shift index s, and frequency modulation index f.

The Multi-Gabor coefficient set $\mathbf{\Phi}_{\text{multi}}[x_0]$ is to be marched to produce $\mathbf{\Phi}_{\text{multi}}[x_0 + \Delta x]$. The propagation matrix set $\mathbf{P}^{N_{pi}}[s,\, f\,, \mathbf{n}, \mathbf{m}]$ is applied to each $\mathbf{\Phi}^{N_{pi}}[x_0, \mathbf{n}, \mathbf{m}] \in \mathbf{\Phi}_{\text{multi}}[x_0]$. In terms of the elements of $\mathbf{\Phi}^{N_{pi}}[x_0, \mathbf{n}, \mathbf{m}]$ given by $\phi^{N_{pi}}{}_{n,m}[x_0]$, the marching formula is

$$\phi^{N_{pi}}{}_{n,m}[x_0] = \sum_{s,\,f} \phi^{N_{pi}}{}_{s,f}[x_0]\, p^{N_{pi}}{}_{n,m}[s,\, f\,], \tag{2.4}$$

where $p^{N_{pi}}{}_{n,m}[s,\, f\,] = \mathbf{P}^{N_{pi}}[s,\, f\,, \mathbf{n}, \mathbf{m}]$.

As a Gabor representation is marched sequentially through space, two problematic behaviors arise: Fields contained in $\mathbf{\Phi}^{N_{pi}}[x_0, \mathbf{n}, \mathbf{m}]$ may spread excessively into the region of support for some $\mathbf{\Phi}^{N_{pj}}[x_0, \mathbf{n}, \mathbf{m}]$ since each section is marched independently, and the field profile may also evolve through space such that the given multi-window discretization is no longer optimal. Both cases lead to excess memory consumption. Therefore, the fields must be re-adapted periodically: After some number of spatial steps in x, the fields are transformed back to the spatial domain, and the process of determining an optimal Gabor discretization are re-run from scratch. At terrain features such as knife edges that can induce multiscale field features, it is also often favorable to trigger a re-adaptation.

2.4. Numerical Results

In this section, three results will be discussed: A simple Multi-Gabor discretization of a multi-scale function, a Multi-Gabor simulation of a synthetic beam, and a Multi-Gabor simulation of realistic long-range propagation through a duct over knife edges.

A. Validation

The first study is of a simple field function with multiscale features. The function is partitioned into four regions, and the optimal window width is selected for each region. The function, partitions, and optimum window widths are visualized in Fig. 2.4.

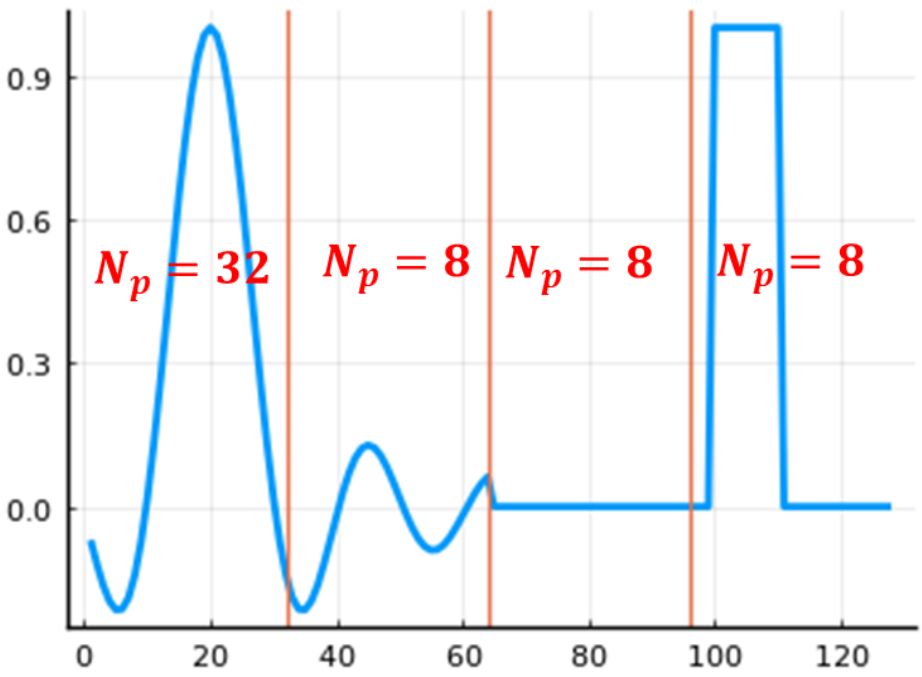


Fig. 2.4. Example of a simple spatial-domain function. The vertical orange lines represent the different subdivisions of the domain, where within each subdivision, Multi-Gabor picks the optimal window width. The optimally picked window widths are shown in red for each subdivision.

In fig. 2.4, narrow window widths are selected for regions with high frequency content, and a wide window width is selected for the region with low frequency content. Using the same threshold between both scenarios, a best-case uniform Gabor discretization of $N_p = 32$ is represented with 114 nonzero coefficients and 0.8% error, while the Multi-Gabor discretization of mixed window widths has 107 nonzero coefficients with 0.6% error; locally optimizing window widths achieves both better sparsity and better accuracy than a uniform discretization.

The first simulation is a synthetic beam profile designed to highlight the performance of Multi-Gabor. The initial profile is composed of two beams of different widths. The profile is marched through space with Multi-Gabor picking the optimal Gabor discretization. The beams have a frequency of 1 GHz. The domain is discretized with $\Delta x = 10$ m, $\Delta z = 0.501\lambda$, and $N_z = 16384$. The atmosphere is a trilinear ocean duct. The ground boundary condition is implemented with image theory, with a reflection coefficient of -1. Gabor RBCs and ABCs are active. Every 10 steps in x, the window widths are re-optimized. The marched fields are shown

in Fig. 2.5, the initial excitation is shown in Fig. 2.6, and a comparison to the performance of uniform Gabor is shown in Fig. 2.7 and Fig. 2.8.

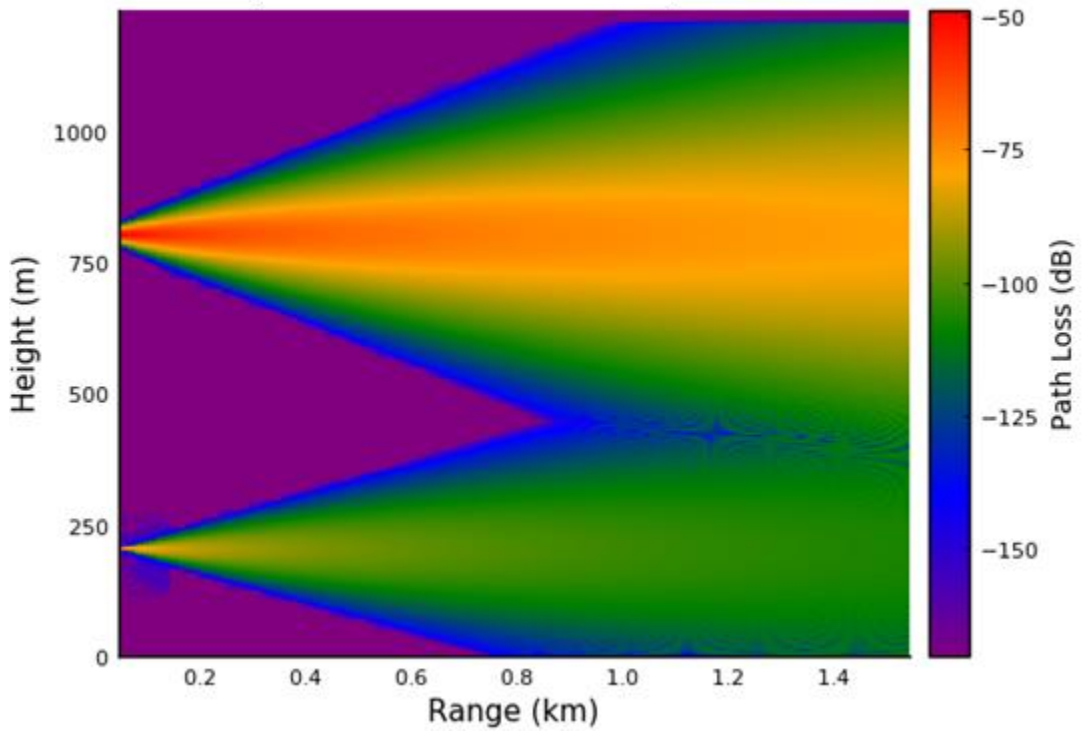


Fig. 2.5. Plot of path loss for a synthetic beam excitation.

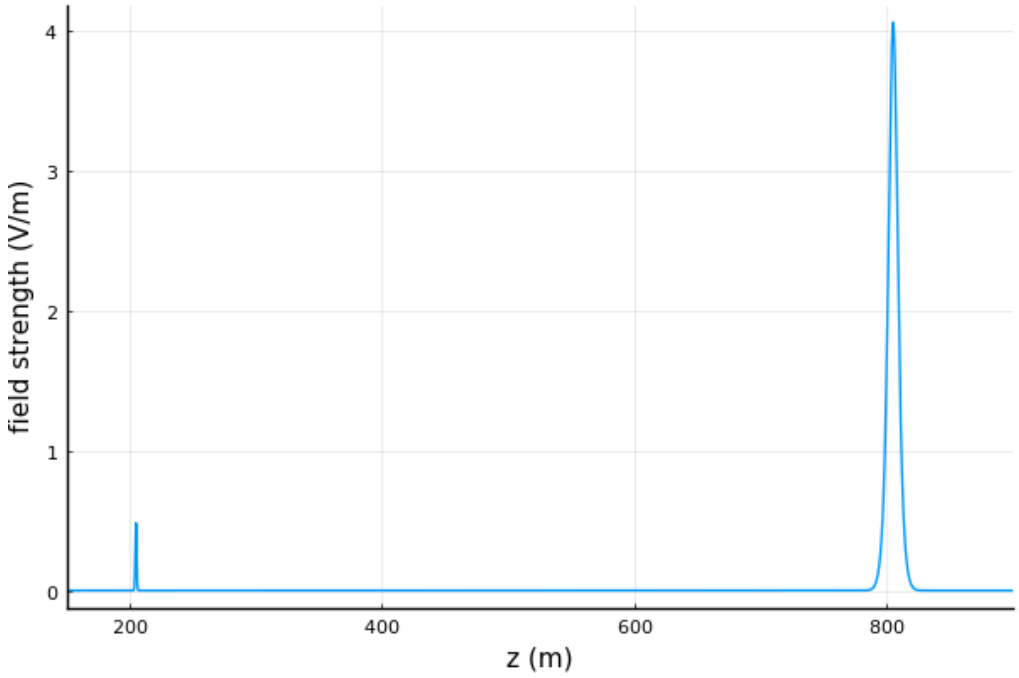


Fig. 2.6. Plot of the initial excitation for the synthetic beam case.

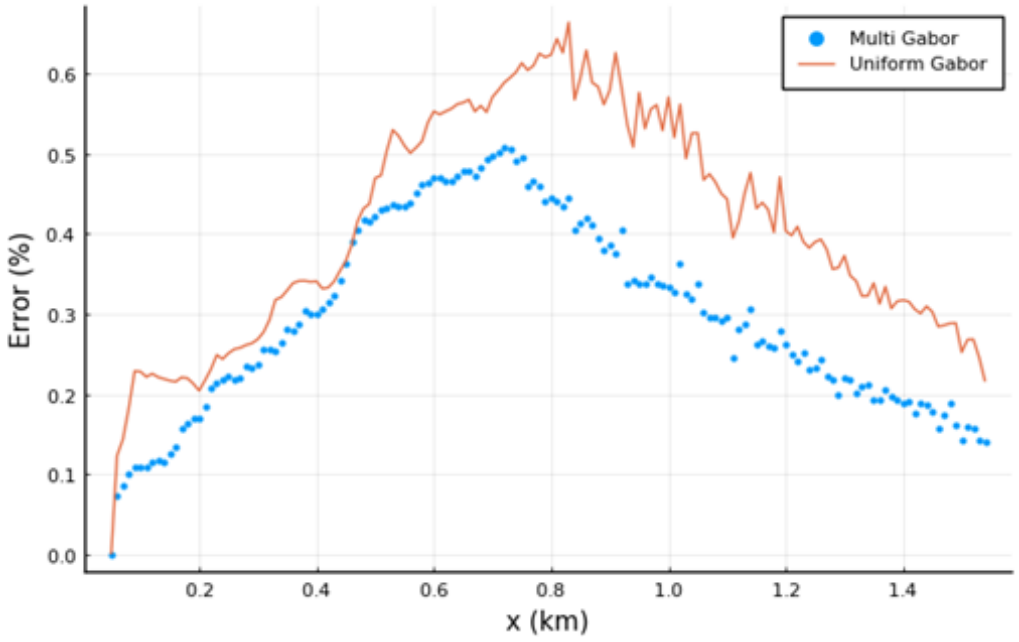


Fig. 2.7. Plot of the error incurred by (blue) Multi-Gabor and (orange) uniform-resolution Gabor at each position in range for the synthetic beam case.

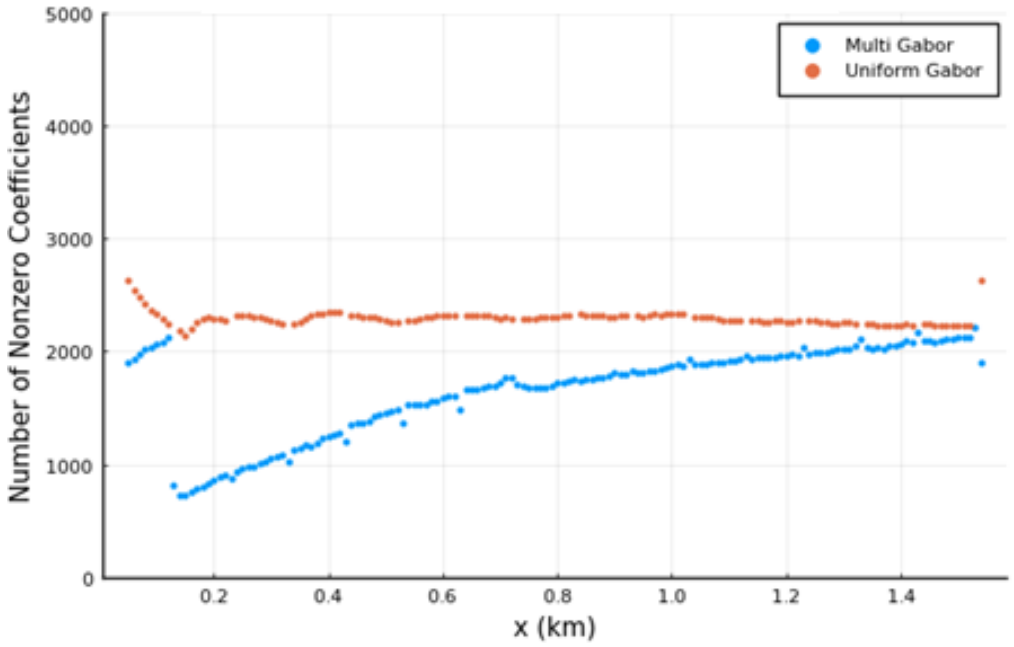


Fig. 2.8. Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the synthetic beam case for (blue) Multi-Gabor and (orange) uniform-resolution Gabor

In Fig. 2.5, the two beams propagate through space and spread outwards. Fig. 2.7 and Fig. 2.8 show that even compared to the best-case discretization using a uniform Gabor window width, the Multi-Gabor method achieves both increased accuracy and a lower number of stored coefficients.

B. Long-Range Propagation

In the next simulation, knife edges are introduced in order to induce multiscale field profiles. Perfectly conducting knife edges truncate fields and produce a diffraction pattern; immediately past the edge of a knife exists a sharp transition from otherwise smooth fields to the shadow region of the knife. This renders an environment with multiple knife edges an ideal scenario for demonstrating the utility of the multi-Gabor transform; the optimal Gabor representation for the diffraction pattern is not necessarily the same as for propagation through smooth atmosphere.

The excitation is a Gaussian beam with a frequency of 2 GHz. The domain is discretized with $\Delta x = 50$ m, $\Delta z = 0.501\lambda$, and $N_z = 16384$. The atmosphere is a trilinear ocean duct. The ground boundary condition is implemented with image theory, with a reflection coefficient of -1. Gabor RBCs and ABCs are active. Every 10 steps in x, the window widths are re-optimized; the window widths are also re-optimized following each knife edge. Knife edges are implemented by

computing a local inverse Gabor Transform about the knife, and zeroing out fields along the knife. The marched fields are shown in Fig. 2.9, the initial excitation is shown in Fig. 2.10, and a comparison to the performance of uniform Gabor is shown in Fig. 2.11 and Fig. 2.12.

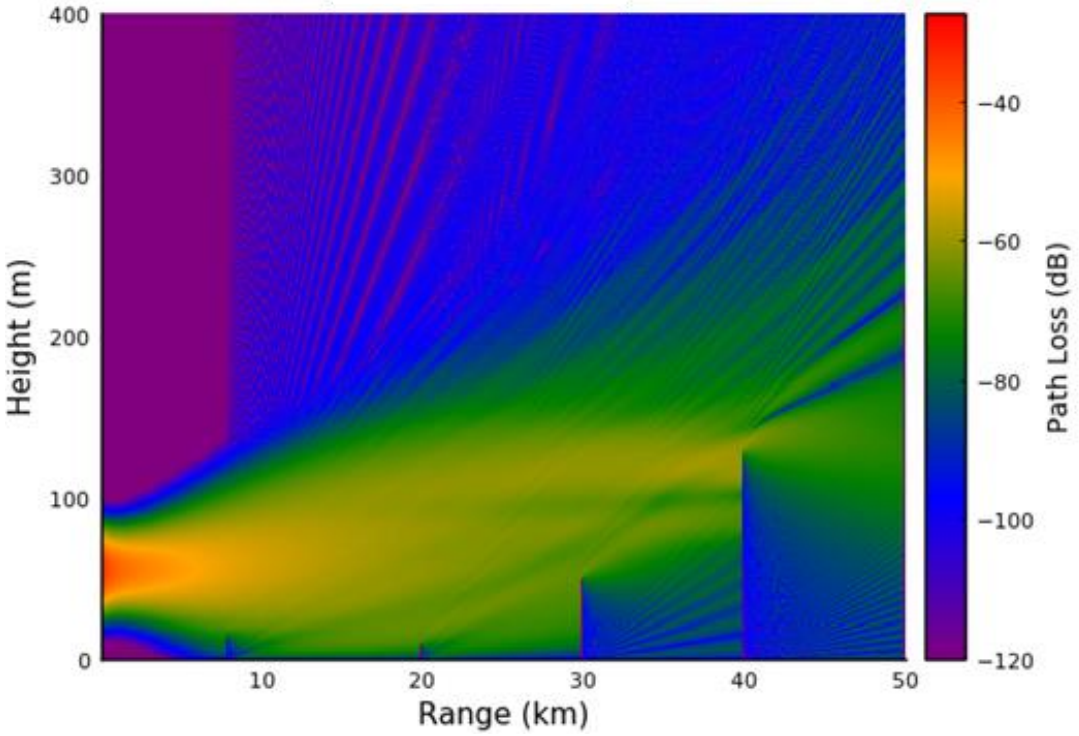


Fig. 2.9. Plot of path loss for a Gaussian beam excitation over an ocean duct with knife edges.

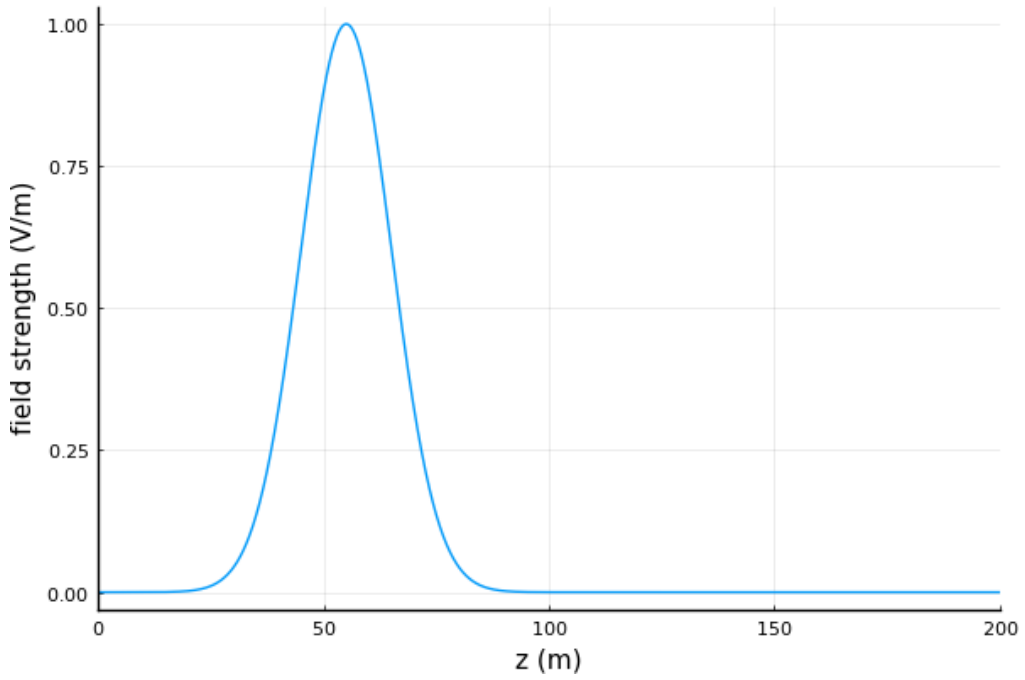


Fig. 2.10. Initial excitation for a Gaussian beam excitation over an ocean duct with knife edges.

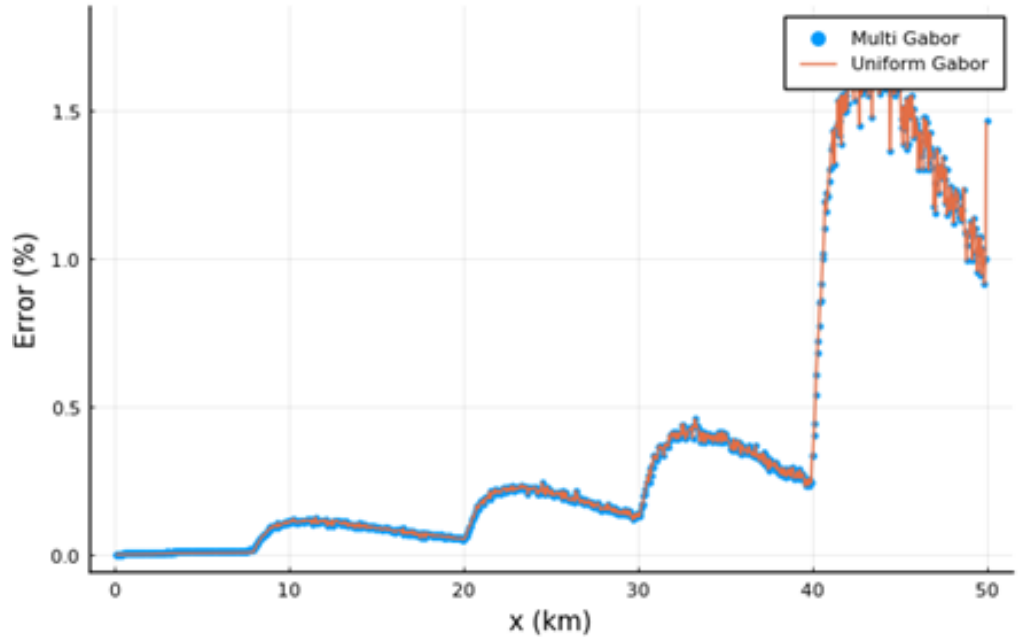


Fig. 2.11. Plot of the error incurred by (blue) Multi-Gabor and (orange) uniform-resolution Gabor at each position in range for the Gaussian beam excitation over an ocean duct with knife edges.

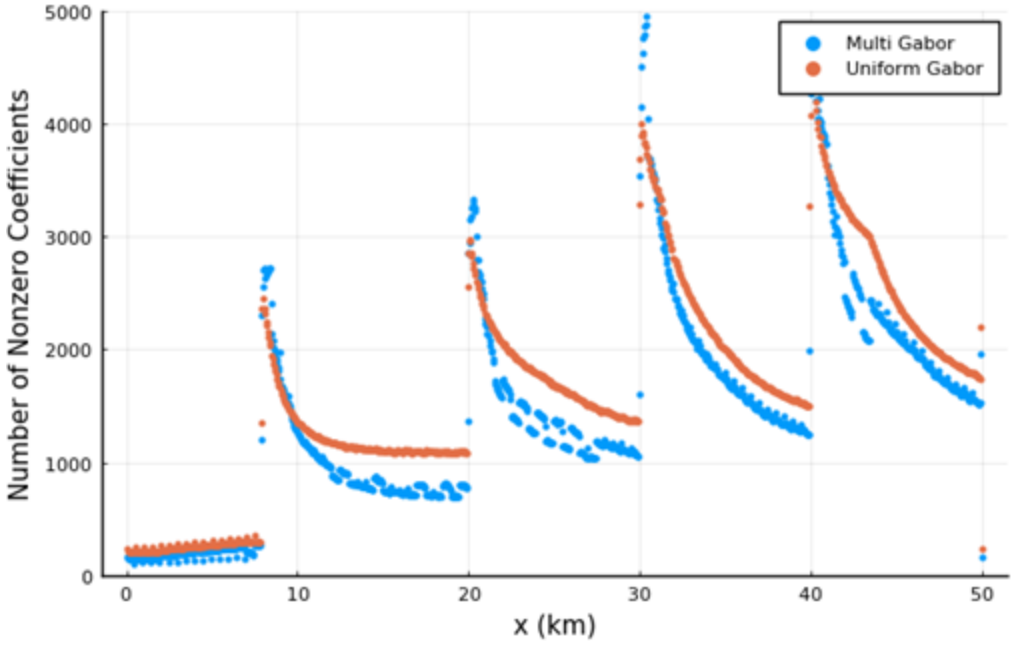


Fig. 2.12. Plot of the number of stored nonzero Gabor coefficient bins at each position in range for the Gaussian beam excitation over an ocean duct with knife edges case for (blue) Multi-Gabor and (orange) uniform-resolution Gabor.

Fig. 2.9 shows a Gaussian beam propagating through a trilinear duct; radiation can be seen deflecting upwards due to the Earth curvature, as well as deflecting downwards as some energy gets trapped inside the duct. Each knife edge creates a diffraction pattern. From Fig. 2.11 and Fig. 2.12, it can be seen that the multi-Gabor framework uses less memory at runtime than the uniform-resolution Gabor framework, while maintaining the same accuracy. The savings in memory tend to grow over time as the diffraction patterns following knife edges interact with the ocean duct.

Chapter 3. Hybrid Split-Step Fourier – Finite Difference Solver for Long-Range Propagation over Rural Terrain

3.1 Introduction

In the study of long-range electromagnetic wave propagation, there is significant interest in the modeling of complex terrain. Detailed terrain description is particularly relevant in the simulation of links in rural environments. Such links are characterized by features such as hills, buildings, water, mountains, vegetation, often all with varying degrees of roughness. There exist many methods for approximating terrain reflection, diffraction, and scattering, as well as for rigorously solving the wave equation, but there is no one-size-fits-all solution for large-scale problems.

This chapter introduces a hybrid Split-Step Fourier – Finite Difference solver to accurately march fields over terrain. Split-Step Fourier (SSF) will be used to march fields through atmosphere above the terrain, and a rigorous Finite Difference (FD) solver will compute the scattering of fields from the terrain itself. The SSF and FD solvers are implemented in the top and bottom sections respectively of a partitioned domain, with a splitting and recombination occurring at each spatial step to produce a full field profile. The FD solver solves the Helmholtz equation with an iterative solver, accelerated by a sweeping physics-based preconditioner.

3.2 Construction of Split-Step Fourier and Finite Difference Methods

This section describes the preliminaries for marching fields through space with both SSF and FD. The SSF solver is exactly as outlined in previous chapters. The second solver is a finite difference method, which rigorously solves the wave equation for arbitrarily complex atmosphere and terrain. The excitation of the finite difference setup can be constructed in such a way to make the finite difference propagator march a field profile from slice to slice, much like SSF. The FD method is accelerated by a sweeping preconditioner, described later in this section.

A. Finite Difference Solver

A finite-difference method accelerated with a sweeping preconditioner will be used to solve the Helmholtz equation. In context, it will be used to march an initial profile $\psi_0(z)$ forward by Δx.

The fields will be decomposed into TE and TM modes, where TE modes are represented by the y-directed component E_y and TM fields are represented by H_y. For TE modes, the Helmholtz equation is

$$\nabla^2 E_y + k_0{}^2 n^2 E_y = f \,, \tag{3.1}$$

where f is a field excitation, n is the refractive index of the medium, and k_0 is the free space wavenumber.

Consider a rectangular FD domain with directions x (left-right) and z (up-down). Let the space be discretized into rectangular coordinates, where h is the step size in x and z. Let h be much smaller than Δx from the SSF domain and also subject to the constraint that $\Delta x / h$ is an integer. In the FD domain, an element in space (z, x) will be denoted by the coordinates (ζ, χ), such that the point (ζ, χ) corresponds to the location $(z = \zeta h, x = \chi h)$ in physical space. The central difference discretization of the TE Helmholtz equation is

$$\begin{aligned} &\frac{1}{h^2}\left(\mathbf{E}_y[\zeta-1,\chi]+\mathbf{E}_y[\zeta+1,\chi]+\mathbf{E}_y[\zeta,\chi-1]+\mathbf{E}_y[\zeta,\chi+1]-4\mathbf{E}_y[\zeta,\chi]\right). \\ &+\mathbf{n}^2[\zeta,\chi]k_0{}^2\mathbf{E}_y[\zeta,\chi]=\mathbf{F}[\zeta,\chi] \end{aligned} \tag{3.2}$$

The boundaries of the rectangular domain will be regarded as radiation surfaces. In order to achieve this, a Perfectly-Matched Layer of width $\eta = bh$ is introduced around all four edges of the domain. Equation (3.2) is rewritten as

$$\begin{aligned} &\frac{1}{h^2}\left(s_l\mathbf{E}_y[\zeta-1,\chi]+s_r\mathbf{E}_y[\zeta+1,\chi]+s_d\mathbf{E}_y[\zeta,\chi-1]+s_u\mathbf{E}_y[\zeta,\chi+1]\right) \\ &-\frac{s_l+s_r+s_d+s_u}{h^2}\mathbf{E}_y[\zeta,\chi]+\frac{\mathbf{n}^2[\zeta,\chi]k_0{}^2}{s_1(\zeta,\chi)s_2(\zeta,\chi)}\mathbf{E}_y[\zeta,\chi]=\mathbf{F}(\zeta,\chi) \end{aligned}, \tag{3.3}$$

where the PML terms are given by

$$\begin{aligned} s_l &= \left(\frac{s_1(\zeta-\frac{1}{2},\chi,\eta,C)}{s_2(\zeta-\frac{1}{2},\chi,\eta,C)}\right), s_r = \left(\frac{s_1(\zeta+\frac{1}{2},\chi,\eta,C)}{s_2(\zeta+\frac{1}{2},\chi,\eta,C)}\right), \\ s_u &= \left(\frac{s_2(\zeta,\chi+\frac{1}{2},\eta,C)}{s_1(\zeta,\chi+\frac{1}{2},\eta,C)}\right), s_d = \left(\frac{s_2(\zeta,\chi-\frac{1}{2},\eta,C)}{s_1(\zeta,\chi-\frac{1}{2},\eta,C)}\right) \end{aligned}, \tag{3.4}$$

and the functions s_1, s_2 are given by

$$s_1(\zeta,\chi,\eta,C)=\frac{1}{1+i\dfrac{\sigma_1(\zeta h,\eta,C)}{k}}, s_2(\zeta,\chi,\eta,C)=\frac{1}{1+i\dfrac{\sigma_2(\chi h,\eta,C)}{k}}. \tag{3.5}$$

The PML σ_1, σ_2 is defined by

$$\sigma_1(z,\eta,C)=\frac{C}{\eta}\left(\frac{z-\eta}{\eta}\right)^2, \sigma_2(x,\eta,C)=\frac{C}{\eta}\left(\frac{x-\eta}{\eta}\right)^2, \tag{3.6}$$

where C is a positive constant; C is tuned empirically in order to minimize reflections from the PML.

For TM modes, the Helmholtz equation is given by

$$\nabla\left(\frac{1}{n^2}\nabla\bullet H_y\right)+k_0{}^2H_y=f. \tag{3.7}$$

There is a crucial difference between the TM and TE equations: Unlike the TE case, the gradient operator in (3.7) is acting on the refractive index n. In the presence of large-scale fluctuations in the refractive index, the discretization process must be modified. A process inspired by the Finite Element Method [66] is used to discretize the TM Helmholtz equation for H_y.

Let the one-dimensional wave equation be

$$\frac{\partial}{\partial x}\left(\frac{1}{n^2}\frac{\partial H}{\partial x}\right)+k_0{}^2H=f. \tag{3.8}$$

Let H and f be discretized according to the Galerkin method [66], with triangular basis functions $N_i(x)$ of width $2h$, so that

$$H(x)=\sum_{j=0}^{N_x}H_jN_j(x), \tag{3.9}$$

$$f(x)=\sum_{j=0}^{N_x}f_jN_j(x). \tag{3.10}$$

The equation (3.8) is then integrated to give

$$\int N_i(x)\frac{\partial}{\partial x}\left(\frac{1}{n^2}\frac{\partial\left(H_j N_j(x)\right)}{\partial x}\right)dx = \int N_i(x)\,f dx - \int N_i(x)k_0{}^2 H_j N_j(x)dx\,. \tag{3.11}$$

By integrating by parts and noting that $N_i(x)$ vanishes at the boundaries, the left-hand side of (3.11) can be evaluated as

$$\int N_i(x)\frac{\partial}{\partial x}\left(\frac{1}{n^2}\frac{\partial\left(H_j N_j(x)\right)}{\partial x}\right)dx = -H_j\int\frac{1}{n^2(x)}\frac{\partial N_i(x)}{\partial x}\frac{\partial N_j(x)}{\partial x}dx\,, \tag{3.12}$$

The system then has the matrix representation

$$\sum_{j=1}^{N} K_{ij}H_j = b_i\,, \tag{3.13}$$

where the elements away from the boundaries are

$$K_{ij} = \int\left(\frac{1}{n^2(x)}\frac{\partial}{\partial x}N_i(x)\frac{\partial}{\partial x}N_j(x) - N_i(x)k_0{}^2 N_j(x)\right)dx\,, \tag{3.14}$$

$$b_i = -\int N_i(x)\,f(x)dx\,. \tag{3.15}$$

The integral (3.14) is evaluated over the width of one basis function at an arbitrary point x, and is split into two parts

$$I_{ij} = -\int_{x-h}^{x+h}\frac{1}{n^2(x)}\frac{\partial N_i(x)}{\partial x}\frac{\partial N_j(x)}{\partial x}dx \tag{3.16}$$

and

$$J_{ij} = -\int_{x-h}^{x+h} N_i(x)k_0{}^2 dx\,. \tag{3.17}$$

These evaluate to

$$I_{i,i} = \frac{1}{h}\left(\frac{1}{n^2(x_{i-1})} + \frac{1}{n^2(x_{i+1})}\right), \tag{3.18}$$

$$I_{i,i-1} = \frac{-1}{h}\left(\frac{1}{n^2(x_{i-1})}\right), I_{i,i+1} = \frac{-1}{h}\left(\frac{1}{n^2(x_{i+1})}\right), \tag{3.19}$$

$$J_{i,i} = -k_0{}^2\frac{2h}{3}, J_{i,i-1} = J_{i,i+1} = -k_0{}^2\frac{h}{6}, \tag{3.20}$$

$$b_i = -f_i\frac{h}{2} - f_{i+1}\frac{h}{2}. \tag{3.21}$$

One line of the matrix equation represented by (3.13) is given by

$$\begin{aligned}&\frac{1}{h}\left(\frac{1}{n^2(x_{i-1})}+\frac{1}{n^2(x_{i+1})}\right)H_i + \frac{-1}{h}\left(\frac{1}{n^2(x_{i-1})}\right)H_{i-1} + \frac{-1}{h}\left(\frac{1}{n^2(x_{i+1})}\right)H_{i+1}\\ &-k_0{}^2\frac{2h}{3}H_i - k_0{}^2\frac{h}{6}\left(H_{i-1}+H_{i+1}\right) = -f_i\frac{h}{2} - f_{i+1}\frac{h}{2}\end{aligned}. \tag{3.22}$$

The non-laplacian terms can be regarded as averages and grouped together to form

$$\begin{aligned}&\frac{1}{h}\left(\frac{1}{n^2(x_{i-1})}+\frac{1}{n^2(x_{i+1})}\right)H_i + \frac{-1}{h}\left(\frac{1}{n^2(x_{i-1})}\right)H_{i-1}\\ &+\frac{-1}{h}\left(\frac{1}{n^2(x_{i+1})}\right)H_{i+1} - k_0{}^2\frac{2h}{3}H_i - k_0{}^2\frac{h}{3}\left(H_i\right) = -f_i h\end{aligned}. \tag{3.23}$$

The expression then simplifies to

$$\frac{-1}{h^2}\left(\frac{1}{n^2(x_{i-1})}+\frac{1}{n^2(x_{i+1})}\right)H_i + \frac{1}{h^2}\left(\frac{1}{n^2(x_{i-1})}\right)H_{i-1} + \frac{1}{h^2}\left(\frac{1}{n^2(x_{i+1})}\right)H_{i+1} + k_0{}^2H_i = f_i. \tag{3.24}$$

In two dimensions, the same procedure can be followed to produce the final expression that is used in the FD solver,

$$\begin{aligned}&\frac{1}{h^2}\left(H_{y(i-1,j)}/n^2{}_{(i-1,j)} + H_{y(i+1,j)}/n^2{}_{(i+1,j)} + H_{y(i,j-1)}/n^2{}_{(i,j-1)} + H_{y(i,j+1)}/n^2{}_{(i,j+1)}\right)\\ &-\frac{1}{h^2}\left(1/n^2{}_{(i-1,j)} + 1/n^2{}_{(i+1,j)} + 1/n^2{}_{(i,j-1)} + 1/n^2{}_{(i,j+1)}\right)H_{y(i,j)}\\ &+k_0{}^2H_{y(i,j)} = f_{(i,j)}\end{aligned}. \tag{3.25}$$

Equation (3.25) is discretized to form the TM central difference equation

$$\frac{1}{h^2}\left(\begin{array}{l}\dfrac{\mathbf{H}_y[\zeta-1,\chi]}{\mathbf{n}^2[\zeta-1,\chi]}+\dfrac{\mathbf{H}_y[\zeta+1,\chi]}{\mathbf{n}^2[\zeta+1,\chi]}+\dfrac{\mathbf{H}_y[\zeta,\chi-1]}{\mathbf{n}^2[\zeta,\chi-1]}+\dfrac{\mathbf{H}_y[\zeta,\chi+1]}{\mathbf{n}^2[\zeta,\chi+1]} \\ -(\dfrac{1}{\mathbf{n}^2[\zeta-1,\chi]}+\dfrac{1}{\mathbf{n}^2[\zeta+1,\chi]}+\dfrac{1}{\mathbf{n}^2[\zeta,\chi-1]}+\dfrac{1}{\mathbf{n}^2[\zeta,\chi+1]})\mathbf{H}_y[\zeta,\chi]\end{array}\right). \tag{3.26}$$
$$+k_0{}^2\mathbf{H}_y[\zeta,\chi]=\mathbf{F}(\zeta,\chi)$$

With the PML as defined in (3.6), the TM central difference equation becomes

$$\frac{1}{h^2}\left(\begin{array}{l}s_l\dfrac{\mathbf{H}_y[\zeta-1,\chi]}{\mathbf{n}^2[\zeta-1,\chi]}+s_r\dfrac{\mathbf{H}_y[\zeta+1,\chi]}{\mathbf{n}^2[\zeta+1,\chi]}+s_d\dfrac{\mathbf{H}_y[\zeta,\chi-1]}{\mathbf{n}^2[\zeta,\chi-1]}+s_u\dfrac{\mathbf{H}_y[\zeta,\chi+1]}{\mathbf{n}^2[\zeta,\chi+1]} \\ -(\dfrac{s_l}{\mathbf{n}^2[\zeta-1,\chi]}+\dfrac{s_r}{\mathbf{n}^2[\zeta+1,\chi]}+\dfrac{s_d}{\mathbf{n}^2[\zeta,\chi-1]}+\dfrac{s_u}{\mathbf{n}^2[\zeta,\chi+1]})\mathbf{H}_y[\zeta,\chi]\end{array}\right). \tag{3.27}$$
$$+\frac{k_0{}^2}{s_1(\zeta,\chi)s_2(\zeta,\chi)}\mathbf{H}_y[\zeta,\chi]=\mathbf{F}(\zeta,\chi)$$

In context, the FD solver will be used to march a field profile $\psi_i(x,z)$ from left-to-right; the $+x$ wall of the FD domain will be denoted as the right-hand side of the domain, and the $-x$ wall of the FD domain will be denoted as the left-hand side of the domain. Since FD is not inherently a marching algorithm, work is needed in the construction of the excitation in order to cause it to mimic one.

The solver will utilize the framework of the Total Field/Scattered Field (TF/SF) formulation for FD problems [67]. TF/SF partitions the domain into a "scattered field" (SF) region about the edges of the domain, and a "total field" (TF) region consisting of the interior of the domain. The FD excitation $\mathbf{F}$ is supplied along the interface between the TF and SF regions. The Helmholtz equation is then solved for all space.

In the context of the solver, TF/SF is used to march a field profile. An initial wavefront $\psi_i(x,z)$ propagating in the $+x$ direction is defined along the left-hand side of the domain, with the aim of computing the total fields on the right-hand side of the domain, a distance of Δx away from the location of the incident fields. Section III will describe the construction of f and the extraction of a propagated field profile in more detail; the remainder of this section is concerned with solving the Helmholtz equation.

While the Helmholtz equation can be solved directly with an iterative scheme, the convergence rate is extremely slow. Thus, a sweeping preconditioner, first introduced in [68], will be utilized in order to accelerate the convergence rate.

The preconditioner will be introduced in four parts. First, the matrix representation of the exact inverse of the Helmholtz operator will be presented. Second, the boundary conditions of the Helmholtz equation will be slightly modified in order to motivate a physical interpretation of the matrix inversion. Third, a sweeping PML will be introduced in order to reduce an expensive matrix inversion to a sequence of approximate, inexpensive Green's function solutions. Fourth, the PML method will be generalized to the true boundary conditions of the Helmholtz equation.

Consider the matrix representation of the FD problem

$$\mathbf{A}\boldsymbol{\psi} = \mathbf{f}\,, \tag{3.28}$$

where $\mathbf{A}$ is the finite-difference Helmholtz operator, and $\boldsymbol{\psi}$ and $\mathbf{f}$ are column-major vectorized forms of the fields and excitation respectively.

The block-matrix representation of (3.28) is

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{21} & & \\ \mathbf{A}_{12} & \mathbf{A}_{22} & \ddots & \\ & \ddots & \ddots & \mathbf{A}_{n_x(n_x-1)} \\ & & \mathbf{A}_{(n_x-1)n_x} & \mathbf{A}_{n_x n_x} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}_1 \\ \boldsymbol{\psi}_2 \\ \vdots \\ \boldsymbol{\psi}_{n_x} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{n_x} \end{bmatrix}, \tag{3.29}$$

where $\boldsymbol{\psi}_m = \begin{bmatrix} \boldsymbol{\psi}[1,m] \\ \boldsymbol{\psi}[2,m] \\ \vdots \\ \boldsymbol{\psi}[n_z,m] \end{bmatrix}^T, \mathbf{f}_m = \begin{bmatrix} \mathbf{f}[1,m] \\ \mathbf{f}[2,m] \\ \vdots \\ \mathbf{f}[n_z,m] \end{bmatrix}^T$ are subsets of the vectors $\boldsymbol{\psi}$ and $\mathbf{f}$, and the matrices $\mathbf{A}_{ij}$ are $n_z \times n_z$ submatrices of $\mathbf{A}$.

The matrix A is factorized as

$$\mathbf{A}\boldsymbol{\psi} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}\boldsymbol{\psi}\,. \tag{3.30}$$

The matrices $\mathbf{D}$, $\mathbf{L}$, and $\mathbf{L}^{\mathrm{T}}$ are given by

$$\mathbf{D} = \begin{bmatrix} \mathbf{S}_1 & & & \\ & \mathbf{S}_2 & \ddots & \\ & & & \mathbf{S}_{n_x} \end{bmatrix}, \tag{3.31}$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{I} & 0 & & & \\ \mathbf{L}_{12} & \mathbf{I} & 0 & & \\ & \mathbf{L}_{23} & \mathbf{I} & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & \mathbf{L}_{n_x(n_x-1)} & \mathbf{I} \end{bmatrix}, \mathbf{L}^{\mathrm{T}} = \begin{bmatrix} \mathbf{I} & \mathbf{L}_{21} & & & \\ 0 & \mathbf{I} & \mathbf{L}_{32} & & \\ & 0 & \mathbf{I} & \ddots & \\ & & \ddots & \ddots & \mathbf{L}_{(n_x-1)n_x} \\ & & & 0 & \mathbf{I} \end{bmatrix}. \tag{3.32}$$

The submatrices are given by $\mathbf{S}_1 = \mathbf{A}_{11}$, and $\mathbf{S}_j = \mathbf{A}_{jj} - \mathbf{A}_{j(j-1)}\mathbf{S}^{-1}{}_{(j-1)}\mathbf{A}_{(j-1)j}$ for $j = 2,...,n_x$,

and $\mathbf{L}_{(j+1)j} = \mathbf{L}^{\mathrm{T}}{}_{j(j+1)} = \mathbf{A}_{(j+1)j}\mathbf{S}^{-1}{}_{j}$.

This factorization can be inverted to produce a solution for $\boldsymbol{\psi}$ in terms of the excitation $\mathbf{f}$

$$\mathbf{L}\mathbf{D}^{-1}\mathbf{L}^{\mathrm{T}}\begin{bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\\ \vdots\\ \mathbf{f}_{n_x}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\psi}_1\\ \boldsymbol{\psi}_2\\ \vdots\\ \boldsymbol{\psi}_{n_x}\end{bmatrix}, \tag{3.33}$$

where

$$\mathbf{D}^{-1} = \begin{bmatrix} S^{-1}{}_1 & & & \\ & S^{-1}{}_2 & \ddots & \\ & & & S^{-1}{}_{n_x}\end{bmatrix} \tag{3.34}$$

and

$$\mathbf{S}_1^{-1} = \mathbf{A}^{-1}{}_{11}, \mathbf{S}_j^{-1} = \mathbf{A}_{jj} - \mathbf{A}_{j(j-1)}\mathbf{S}^{-1}{}_{j-1}\mathbf{A}_{(j-1)j}. \tag{3.35}$$

The matrix product (3.33) has an algorithmic representation given by Algorithm 1.

Algorithm 3.1. Inverse LDLT formula for evaluating (3.33)

1. For $m = 1...n$
   a. $\boldsymbol{\psi}_m = \mathbf{f}_m$
2. For $m = 1...n-1$
   a. $\boldsymbol{\psi}_{m+1} = \boldsymbol{\psi}_{m+1} - \mathbf{A}_{(m+1)m}(\mathbf{S}^{-1}{}_m\boldsymbol{\psi}_m)$
3. For $m = 1...n$
   a. $\boldsymbol{\psi}_m = \mathbf{S}^{-1}{}_m\boldsymbol{\psi}_m$
4. For $m = n-1...1$
   a. $\boldsymbol{\psi}_m = \boldsymbol{\psi}_m - \mathbf{S}^{-1}{}_m(\mathbf{A}_{m(m+1)}\boldsymbol{\psi}_m)$

While the LDLT inversion is exact, it is still not efficient enough; in the inverted matrix D^{-1}, each element S_j^{-1} requires the computation of the inverses of all $S_1 \ldots S_{j-1}$, where each successive S_j^{-1} increases in size.

Instead of inverting a sequence of matrices of increasing size, an approximation will be made that reformulates the application of D^{-1} in terms of inverses of matrices of uniform size. This approximation utilizes the notion of half-space Green's functions, which are introduced below.

Consider a restriction of the problem $A\psi = f$ to the subdomain $j = 1...m$, given by

$$\begin{bmatrix} A_{11} & A_{21} & & \\ A_{12} & A_{22} & \ddots & \\ & \ddots & \ddots & A_{m(m-1)} \\ & & A_{(m-1)m} & A_{mm} \end{bmatrix}^{-1} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_m \end{bmatrix}. \tag{3.36}$$

As part of this restriction, the boundary conditions of the FD equation will also be modified. Rather than using a PML on all four sides, a PML will only be used on 3 sides; the side corresponding to the boundary at the m th entry ($+x$ wall) of the restricted domain will be replaced with a perfectly conducting surface. The derivation will be carried out for this modified problem, and then a method compatible with a PML on all four sides will be introduced, as required by the broader propagation problem.

Suppose that all $f_j : j < b$ is enforced to be 0. The restricted problem becomes

$$\begin{bmatrix} A_{11} & A_{21} & & \\ A_{12} & A_{22} & \ddots & \\ & \ddots & \ddots & A_{m(m-1)} \\ & & A_{(m-1)m} & A_{mm} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ f_m \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_m \end{bmatrix}. \tag{3.37}$$

The LDLT factorization of the restriction of A^{-1} in (3.37) is

$$\begin{bmatrix} I & 0 & & & \\ L_{12} & I & 0 & & \\ & L_{23} & I & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & L_{m(m-1)} & I \end{bmatrix} \begin{bmatrix} S^{-1}{}_1 & & & \\ & S^{-1}{}_2 & & \\ & & \ddots & \\ & & & S^{-1}{}_m \end{bmatrix} \begin{bmatrix} I & L_{21} & & & \\ 0 & I & L_{32} & & \\ & 0 & I & \ddots & \\ & & \ddots & \ddots & L_{(m-1)m} \\ & & & 0 & I \end{bmatrix}. \tag{3.38}$$

Due to the condition $f_j = 0$ for $j < b$, the equation for the element ψ_m on the right-hand side of (3.38) reduces to a simple application of $S^{-1}{}_m$,

$$\psi_m = S^{-1}{}_m f_m. \tag{3.39}$$

Equation (3.39) is the crux of the new cost-efficient method. Relating the application of $S^{-1}{}_m$ to the solution of (3.38) motivates an important physical interpretation; the system (3.38) can be interpreted as applying a Green's function on a domain restricted to the first m columns to a point source, and extracting the last value of the resulting vector.

More precisely, the use of the physical interpretation of equation (3.38) to solve equation (3.39) can be expressed as follows: let an operation $\psi_m = T_m f_m$ denote the process of solving the Helmholtz equation on a domain restricted to $j = 1...m$ on an excitation $\begin{bmatrix} 0 \\ 0 \\ \vdots \\ f_m \end{bmatrix}$, and extracting the *m*th value from the resultant ψ vector. The operation $\psi_m = T_m f_m$ is used in place of the operation $\psi_m = S^{-1}{}_m f_m$.

There is now no longer a need to exactly invert a sequence of S_j matrices. However, more work is still necessary in order to realize a true performance improvement, as the sub-problems still increase in size with increasing m. A sweeping PML will now be introduced in order to restrict each sub-problem to a uniform, small size, independent of m.

Consider a system restricted on the values in x: $m - b + 1 \le j \le m$. Let a PML of width b absorb any outgoing radiation in the left-sided direction. Provided b is sufficiently large, solving the sub-problem with a restricted f vector and extracting the *m*th value of the solution vector will be nearly identical to solving the entire system from $j = 1...m$.

Graphically, this process can be interpreted as sweeping a domain from left to right. At each slice, a local problem is solved: A field localized to the right-most column is allowed to locally spread, with a conducting wall along the right-most side of the domain and a PML encompassing all space to the left of the excitation. The right-most column of the solved field is stored, and is used in the process of constructing the excitation for the next slice. The PML can be thought of as "sweeping" from left to right as it chases the slices being operated on.

An operator $\mathbf{H}_m$ can be defined to describe the operation to obtain $\boldsymbol{\psi}_m$ from $\mathbf{f}_m$ while employing a sweeping PML. A modified Helmholtz equation with the sweeping PML is used; the PML parameters are given by

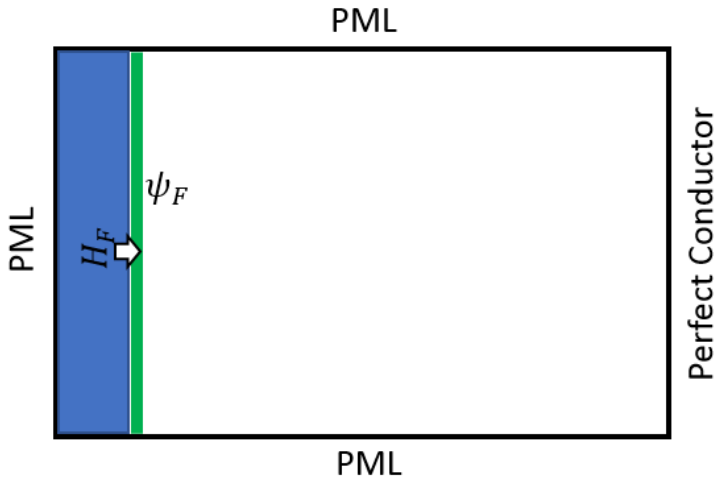
$$s_l = \left(\frac{s_1(\zeta - \frac{1}{2}, \chi, bh, C_L)}{s_2(\zeta - \frac{1}{2}, \chi, bh, C_L)} \right), s_r = 1,$$
$$s_u = \left(\frac{s_2(\zeta, \chi + \frac{1}{2}, \eta, C)}{s_1(\zeta, \chi + \frac{1}{2}, \eta, C)} \right), s_d = \left(\frac{s_2(\zeta, \chi - \frac{1}{2}, \eta, C)}{s_1(\zeta, \chi - \frac{1}{2}, \eta, C)} \right), \tag{3.40}$$

where C_L is tuned empirically in order to avoid reflections. The modified local Helmholtz operator is defined by the matrix $\mathbf{A}^L$, and the matrices $\mathbf{A}^L{}_{ij}$ are defined as $n_z \times n_z$ submatrices of $\mathbf{A}^L$.

$\boldsymbol{\psi}_m = \mathbf{H}_m \mathbf{f}_m$ is computed by formulating the vector $\begin{bmatrix} 0 & 0 & 0 & \mathbf{f}_m{}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ of length $n_z b$, then solving the equation

$$\begin{bmatrix} \mathbf{A}^L{}_{(m-b+1)(m-b+1)} & \mathbf{A}^L{}_{(m-b+2)(m-b+1)} & & \\ \mathbf{A}^L{}_{(m-b+1)(m-b+2)} & \mathbf{A}^L{}_{(m-b+1)(m-b+1)} & \ddots & \\ & \ddots & \ddots & \mathbf{A}^L{}_{(m)(m-1)} \\ & & \mathbf{A}^L{}_{(m-1)(m)} & \mathbf{A}^L{}_{(m)(m)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \mathbf{f}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}_1 \\ \boldsymbol{\psi}_2 \\ \vdots \\ \boldsymbol{\psi}_m \end{bmatrix} \tag{3.41}$$

exactly, and then extracting ψ_m from the solution. The matrix inversion is most efficiently performed by permuting all matrices and vectors from column-major to row-major ordering in order to compress the bandwidth of the matrix, and then solving with LU. This process is visualized in Fig. 3.1.



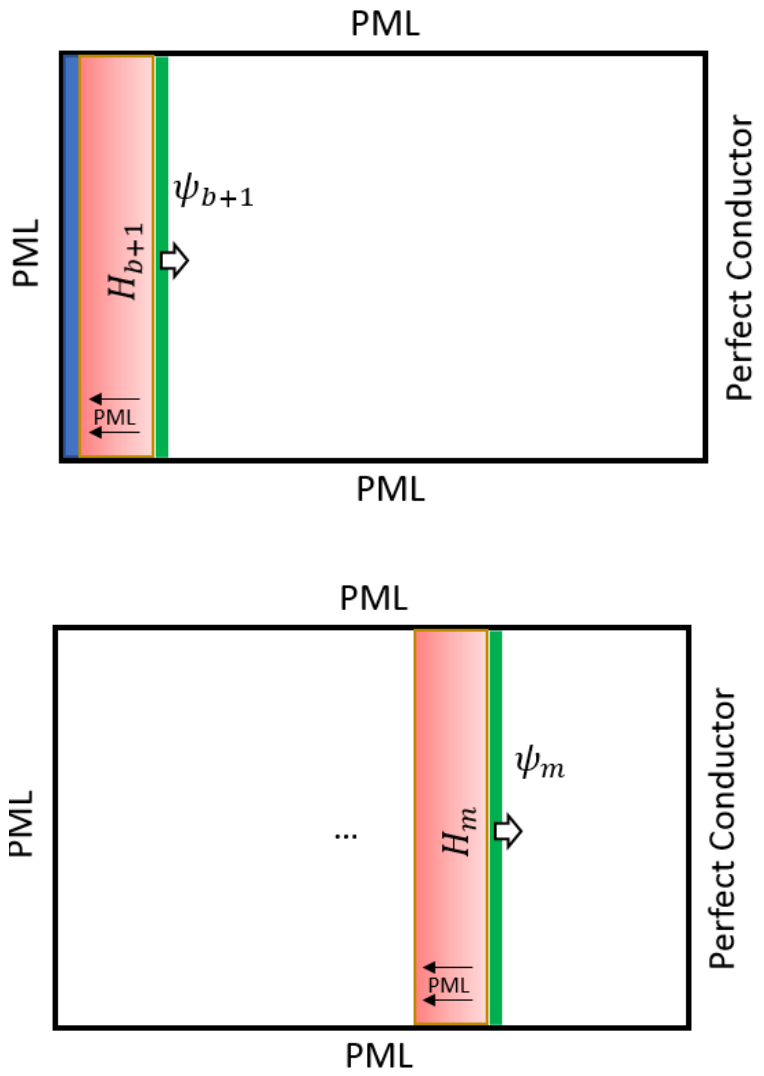


Fig. 3.1. Visualization of the sweeping preconditioner. An initial slice is provided along the left side of the domain (top) given by $\boldsymbol{\psi}_{\mathrm{F}}$, and is marched forward (middle, bottom) for $b+1 \le m \le n_x$ by solving a localized Helmholtz equation $\mathbf{H}_m$ to produce $\boldsymbol{\psi}_m$, with a PML along the left side of the local solution region.

The above procedure can be encompassed into a preconditioner linear operator, given by

$$\mathbf{P} = \mathbf{L}\begin{bmatrix} \mathbf{H}_{\mathbf{F}} & & & \\ & \mathbf{H}_{b+1} & & \\ & & \ddots & \\ & & & \mathbf{H}_{n_x} \end{bmatrix}\mathbf{L}^{\mathrm{T}}. \tag{3.42}$$

The operation $\boldsymbol{\psi} = \mathbf{P}\mathbf{f}$ is described by the following algorithm 3.2.

Algorithm 3.2. Procedure for applying the preconditioner operator to evaluate $\boldsymbol{\psi} = \mathbf{Pf}$

1. $\boldsymbol{\psi} = \mathbf{f}$
2. $\boldsymbol{\psi}_{b+1} = \boldsymbol{\psi}_{b+1} - \mathbf{A}_{b+1,\mathbf{F}} \left(\mathbf{A}_{\mathbf{F},\mathbf{F}} \setminus \boldsymbol{\psi}_{\mathbf{F}} \right)$
3. For $m = b+1 ... n_x - 1$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z(b-1)+1 ... n_z b} = \boldsymbol{\psi}_m$
    c. $\boldsymbol{\psi}_{m+1} = \boldsymbol{\psi}_{m+1} - \mathbf{A}_{m+1,m} \mathbf{H}_m \tilde{\boldsymbol{\psi}}$
4. For $m = b+1 ... n_x$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z(b-1)+1 ... n_z b} = \boldsymbol{\psi}_m$
    c. $\boldsymbol{\psi}_m = \mathbf{H}_m \tilde{\boldsymbol{\psi}}$
5. For $m = n_x - 1 ... b+1$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z(b-1)+1 ... n_z b} = \mathbf{A}_{m,m+1} \boldsymbol{\psi}_{m+1}$
    c. $\boldsymbol{\psi}_m = \boldsymbol{\psi}_m - H_m \tilde{\boldsymbol{\psi}}$
6. $\boldsymbol{\psi}_{\mathbf{F}} = \boldsymbol{\psi}_{\mathbf{F}} - \mathbf{A}_{\mathbf{F},\mathbf{F}} \setminus \left(\mathbf{A}_{\mathbf{F},b+1} \boldsymbol{\psi}_{b+1} \right)$

The parameter $\mathbf{F}$ denotes the indices $1...b$.

With the problem solved for a scenario in which the right-side wall of the domain is a PEC, the procedure for a problem where all four side walls are radiation surfaces will now be developed. The domain will be partitioned down the middle into a left side and a right side. Rather than one PML sweeping from left to right, two PMLs will sweep from the side walls towards the center.

The system is re-interpreted as a sequence of Green's functions as was done for the one-sided case, except that the Green's functions H_m are constructed and applied in opposing manners depending on whether m belongs to the left side or the right side of the domain. For $m = 1...(n_x+1)/2 - 1$, a perfect conductor is positioned on the right-hand wall; for $m = (n_x+1)/2 + 1...n_x$, a perfect conductor is positioned on the left-hand wall. The center element $m = (n_x+1)/2$ is treated as a special case, where a Green's function is solved with a PML on all four sides of the local domain; the operator $\mathbf{H}_{(n_x+1)/2}$ excites and extracts from the middle of the

domain, rather than the left or right side. The procedure for the two-front PML method is given by algorithm 3.3.

Algorithm 3.3. Two-front sweeping preconditioner application procedure

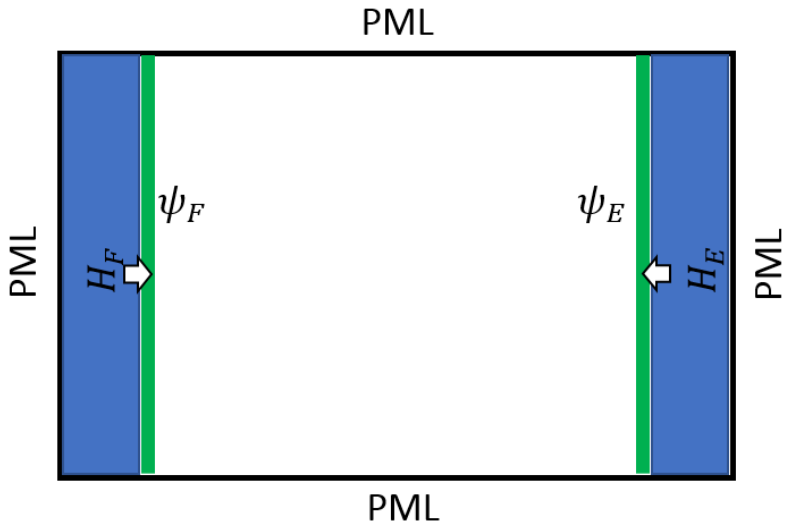
1. $\boldsymbol{\psi} = \mathbf{f}$
2. $\boldsymbol{\psi}_{b+1} = \boldsymbol{\psi}_{b+1} - \mathbf{A}_{b+1,\mathbf{F}} \left(\mathbf{A}_{\mathbf{F},\mathbf{F}} \setminus \boldsymbol{\psi}_{\mathbf{F}} \right)$
3. $\boldsymbol{\psi}_{n_x - b} = \boldsymbol{\psi}_{n_x - b} - \mathbf{A}_{n_x - b,\mathbf{E}} \left(\mathbf{A}_{\mathbf{E},\mathbf{E}} \setminus \boldsymbol{\psi}_{\mathbf{E}} \right)$
4. For $m = b+1 \ldots \left(n_x + 1 \right) / 2 - 1$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z (b-1)+1 \ldots n_z b} = \boldsymbol{\psi}_m$
    c. $\boldsymbol{\psi}_{m+1} = \boldsymbol{\psi}_{m+1} - \mathbf{A}_{m+1,m} \mathbf{H}_m \tilde{\boldsymbol{\psi}}$
5. For $m = n_x - b + 1 \ldots \left(n_x + 1 \right) / 2 + 1$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z (b-1)+1 \ldots n_z b} = \boldsymbol{\psi}_m$
    c. $\boldsymbol{\psi}_{m-1} = \boldsymbol{\psi}_{m-1} - \mathbf{A}_{m,m-1} \mathbf{H}_m \tilde{\boldsymbol{\psi}}$
6. For $m = b+1 \ldots n_x$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z (b-1)+1 \ldots n_z b} = \boldsymbol{\psi}_m$
    c. $\boldsymbol{\psi}_m = \mathbf{H}_m \tilde{\boldsymbol{\psi}}$
7. For $m = \left(n_x + 1 \right) / 2 + 1 \ldots n_x - b$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z (b-1)+1 \ldots n_z b} = \mathbf{A}_{m-1,m} \boldsymbol{\psi}_{m-1}$
    c. $\boldsymbol{\psi}_m = \boldsymbol{\psi}_m - \mathbf{H}_m \tilde{\boldsymbol{\psi}}$
8. For $m = \left(n_x + 1 \right) / 2 - 1 \ldots b + 1$
    a. $\tilde{\boldsymbol{\psi}} = zeros(n_z b)$
    b. $\tilde{\boldsymbol{\psi}}_{n_z (b-1)+1 \ldots n_z b} = \mathbf{A}_{m,m+1} \boldsymbol{\psi}_{m+1}$
    c. $\boldsymbol{\psi}_m = \boldsymbol{\psi}_m - \mathbf{H}_m \tilde{\boldsymbol{\psi}}$
9. $\boldsymbol{\psi}_{\mathbf{F}} = \boldsymbol{\psi}_{\mathbf{F}} - \boldsymbol{\psi}_{\mathbf{F},\mathbf{F}} \setminus \left(\mathbf{A}_{\mathbf{F},b+1} \boldsymbol{\psi}_{b+1} \right)$
10. $\boldsymbol{\psi}_{\mathbf{E}} = \boldsymbol{\psi}_{\mathbf{E}} - \mathbf{A}_{\mathbf{E},\mathbf{E}} \setminus \left(\mathbf{A}_{\mathbf{E},n_x - b} \boldsymbol{\psi}_{n_x - b} \right)$

The parameter $\mathbf{F}$ denotes the indices $1...b$, and $\mathbf{E}$ denotes the indices $n_x...n_x - b + 1$. The sweeping PML interpretation of algorithm 3 is visualized in Fig. 3.2.

Algorithm 3 represents an operator $\mathbf{P}$ such that $\boldsymbol{\psi} \approx \mathbf{P}\mathbf{f}$. $\mathbf{P}$ approximately inverts $\mathbf{A}$, and is used to precondition the Helmholtz system

$$\mathbf{P}\mathbf{A}\boldsymbol{\psi} = \mathbf{P}\mathbf{f}. \tag{3.43}$$

The system (3.43) can be solved for $\boldsymbol{\psi}$ iteratively, with methods such as GMRES. Convergence is empirically observed to be very fast.



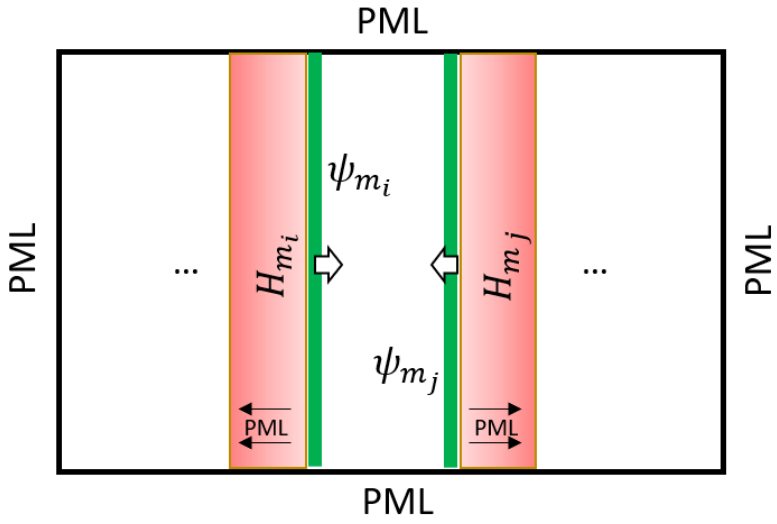


Fig. 3.2. Visualization of two-front PML. Two field slices are defined along the left and right edges of the domain (top), and the local Helmholtz problems are solved from both sides iteratively sweeping towards the center (bottom).

3.3 Hybridization of Split-Step Fourier and Finite Difference

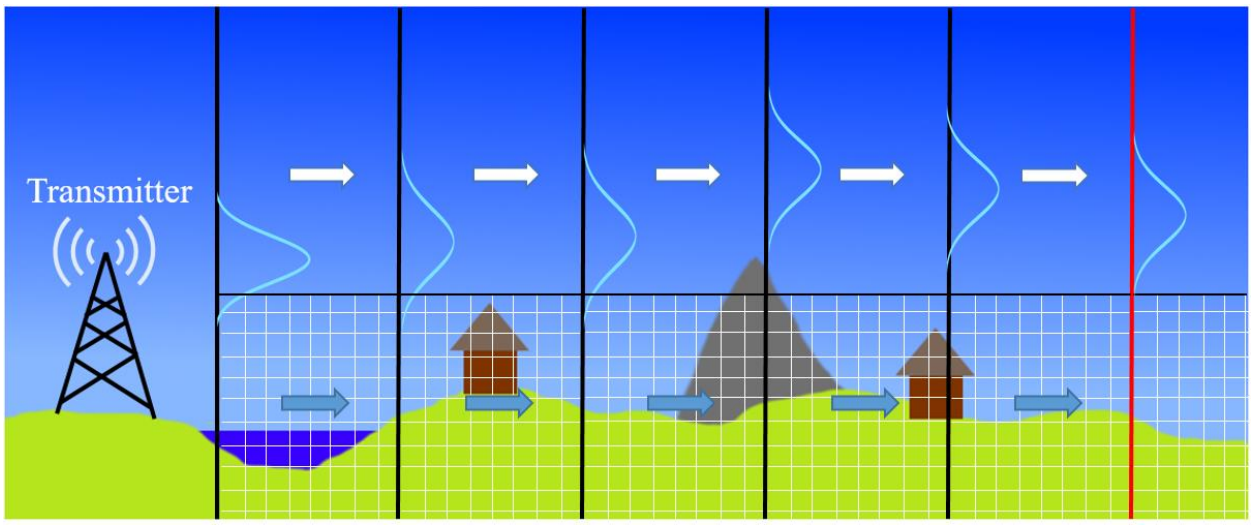


Fig. 3.3. Illustration of how fields are marched in the hybrid scheme; fields in the atmosphere are marched with SSF, while fields near the ground are marched with FD. Fields near the interface are smoothly blended together to form a complete profile.

This solver combines the Split-Step Fourier and Finite Difference methods in order to build a robust and efficient framework to march a wavefront from one vertical slice to the next. SSF will be used to march fields through the atmosphere, and FD will be used to march fields over terrain and past obstacles. The FD component of the solver is used only near the terrain surface. The FD

domain is constrained to a rectangular region near the terrain interface; this small region is chosen to encapsulate detailed features such as terrain height changes, buildings, and vegetation. For all other fields outside of the small FD region about the surface, the SSF propagator will be used. Since the region outside an appropriately chosen FD domain is quasi-homogeneous atmosphere, SSF is an appropriate, efficient choice. The partitioning of the domain is shown in Fig. 3.3.

The process of marching fields through space is described in three operations:

1. Fields at some slice x_0 are split into two slightly overlapping regions. At the boundaries of the regions, the fields are tapered off with smooth window functions.
2. Both fields are marched forward by Δx in their respective regions; the fields in the SSF region are marched according to the Split-Step Fourier method, and the fields in the FD region are marched with the preconditioned FD solver.
3. The resultant fields from the two solvers are smoothly blended together to construct the full field profile at $x_0 + \Delta x$.

The remainder of this section is a description of the full implementation detail of the hybrid solver. The general outline listed above serves as the guidelines for the solver, but several aspects are modified in practice in the interest of performance and accuracy. Most notably, the FD solver and SSF solvers use different resolutions, and the FD solver requires two field slices as input while the SSF solver only requires one field slice as input. The flow diagram for the marching procedure is shown in Fig. 3.4.

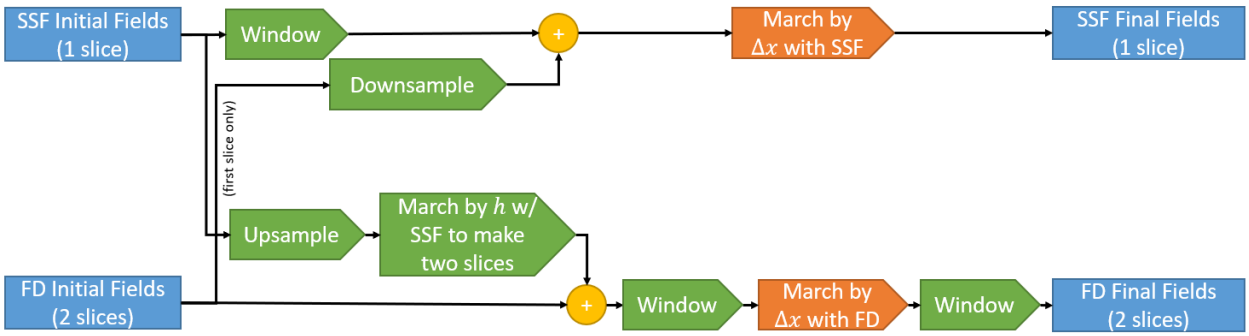


Fig. 3.4. Flow diagram for marching fields with the hybrid scheme.

A. Split Fields into Two Regions

Given a field profile at some vertical slice $\boldsymbol{\psi}(x_0)[z]$, the vertical domain $z_{\min} \le z \le z_{\max}$ is divided into two overlapping intervals: The FD region $z_{\min} \le z \le z_{FD}$ and the SSF region $z_{SSF} \le z \le z_{\max}$. The FD region encompasses all terrain features and a small amount of atmosphere above such features; the SSF region contains the rest of the atmosphere above the terrain. The two regions slightly overlap, so $z_{FD} > z_{SSF}$.

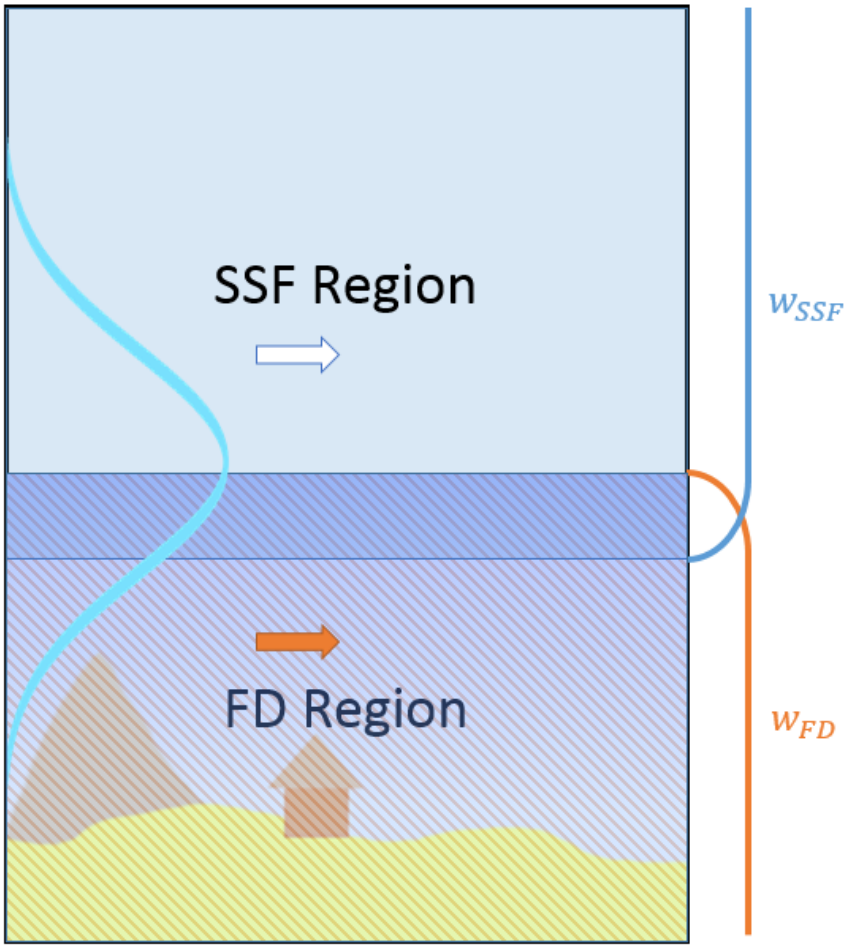


Fig. 3.5. Visualization of the SSF and FD regions, showing the overlap between the regions and the corresponding window functions.

Window functions will be used to smoothly separate and blend fields that exist in the overlap region. Two window functions $\mathbf{w}_{\mathrm{FD}}[\mathbf{z}]$ and $\mathbf{w}_{\mathrm{SSF}}[\mathbf{z}]$ are defined for all z, with the condition that $w_{\mathrm{FD}}(z) = 0$ for $z > z_{\mathrm{FD}}$, $w_{\mathrm{SSF}}(z) = 0$ for $z < z_{\mathrm{SSF}}$, both windows sum together to unity at every z, and the overlap between the two windows is contained in the region $z_{\mathrm{SSF}} \le z \le z_{\mathrm{FD}}$. The FD and

SSF fields are defined as the result of applying the window functions $\mathbf{w}_{\mathrm{FD}}[\mathbf{z}]$ and $\mathbf{w}_{\mathrm{SSF}}[\mathbf{z}]$ to the field profile: $\boldsymbol{\psi}_{\mathrm{FD}}[x_0, \mathbf{z}] = \mathbf{w}_{\mathrm{FD}}[z] \odot \boldsymbol{\psi}[x_0, \mathbf{z}]$ for $z_{\min} \le z \le z_{\mathrm{FD}}$, and $\boldsymbol{\psi}_{\mathrm{SSF}}[x_0, \mathbf{z}] = \mathbf{w}_{\mathrm{SSF}}[z] \odot \boldsymbol{\psi}[x_0, \mathbf{z}]$ for $z_{\mathrm{SSF}} \le z \le z_{\max}$. The SSF and FD region partitions are shown in Fig. 3.5. In the following step, ψ_{FD} and ψ_{SSF} will be treated independently in their respective domains of support.

B. March Fields Through Upper Region with SSF

The fields $\boldsymbol{\psi}_{\mathrm{SSF}}[x_0, \mathbf{z}]$ are marched forwards to $\boldsymbol{\psi}_{\mathrm{SSF}}[x_0 + \Delta x, \mathbf{z}]$ with the Split-Step Fourier method as outlined in chapter 1. The boundary z_{SSF} is chosen such that negligible energy escapes below the SSF region. An absorbing layer is used to simulate a radiation boundary at the top of the SSF region.

C. March Fields Through Lower Region with FD

The fields $\boldsymbol{\psi}_{\mathrm{FD}}[x_0, \mathbf{z}]$ is marched forwards by Δx via a high-resolution Finite Difference solver. The incident fields $\boldsymbol{\psi}_{\mathrm{FD}}[x_0, \mathbf{z}]$ are interpolated to a higher-resolution domain $\boldsymbol{\psi}_{\mathrm{FD}}[x_0, \boldsymbol{\zeta}]$ due to the higher-resolution requirements of the FD problem. In principle, this step could be avoided if the SSF z domain were sufficiently dense such that $\Delta z = \Delta\zeta$, however this would come at a significant cost of performance of the SSF component.

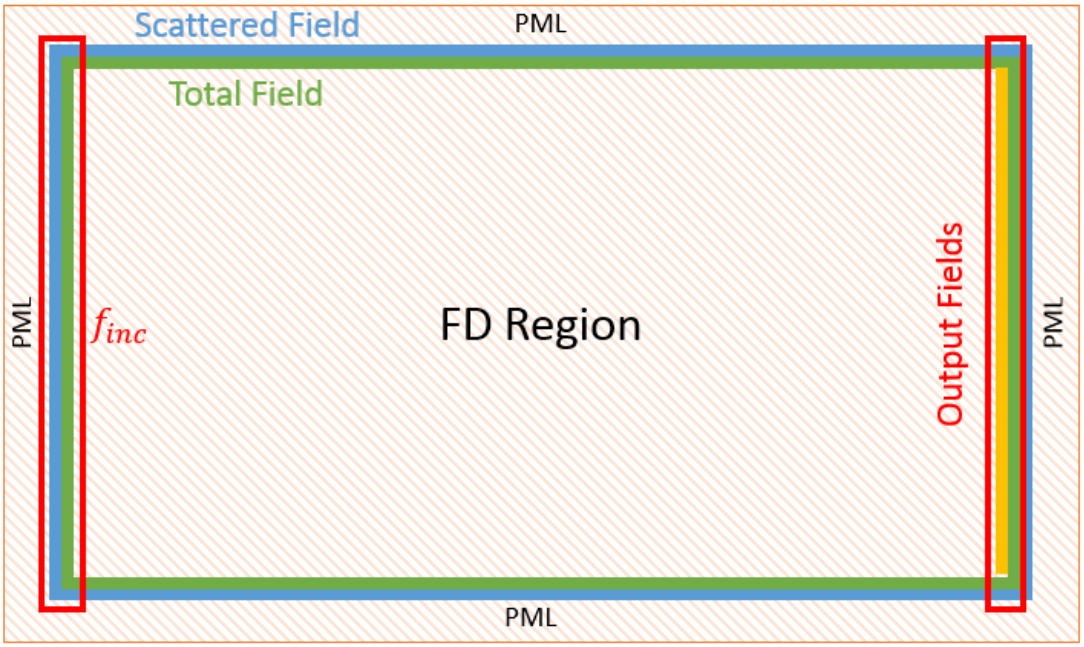


Fig. 3.6. The formulation of the FD region, including PML placement, TF/SF region definitions, and locations of incident fields and output fields. Note that two input and output field slices are used due to the TF/SF formulation using two concentric contours.

The TF/SF framework is used to express the incident fields in a manner that is compatible with a FD problem, as well as to extract the FD solution to produce a marched field. The TF/SF interface, along which an excitation is supplied, consists of two concentric rectangles. The incident fields are supplied at the left-hand side of the TF/SF interface, and the output fields are measured along a slice just before the right-hand side of the TF/SF interface after the full solution is computed. When concerned with only one-directional propagation, where there are nonreflecting boundaries on the right-hand side, and when a narrowly windowed incident field ensures no propagating fields reach the top/bottom boundaries, it is valid to only provide incident fields on the left-hand side of the domain; the top, bottom, and right sides of the TF/SF interface can be initialized as zero without loss of accuracy.

Since the TF/SF field definition requires incident fields to be supplied in two concentric rectangles, the FD solver must use two field slices for the input and output. For propagating fields that stay below the SSF-FD interface across a step Δx, implementation is straightforward, as the two output slices from one FD region are recycled as input to the next FD region. However, there is slight additional complexity when interacting with fields that move between the FD and SSF

regions, since the SSF region only has one slice for its input/output. Fields incident from the SSF region onto the FD region are marched forward by one step h with a localized Split-Step Fourier solver in free space. For fields incident from the FD region onto the SSF region, only the slice corresponding exactly to the location $x_0 + \Delta x$ is extracted. The TF/SF problem setup using the two pairs of field slices is shown in Fig. 3.6.

The expression for the FD excitation $\mathbf{F}$ in terms of the incident fields $\mathbf{F}_{\text{inc}}$ is given by

$$\begin{aligned} \mathbf{F}[\zeta, \xi_{\text{lhs}} + 1] &= \frac{-\mathbf{F}_{\text{inc}}[\zeta, \xi_{\text{lhs}}]}{h^2 \epsilon_{r\,\text{TE}}(\zeta, \xi_{\text{lhs}})} \\ \mathbf{F}[\zeta, \xi_{\text{lhs}}] &= \frac{\mathbf{F}_{\text{inc}}[\zeta, \xi_{\text{lhs}} + 1]}{h^2 \epsilon_{r\,\text{TE}}(\zeta, \xi_{\text{lhs}} + 1)} \end{aligned}, \tag{3.44}$$

where the function $\epsilon_{r\,\text{TE}}(\zeta, \xi_0)$ is set to unity for the TM case and to the relative permittivity at (ζ, ξ_0) for the TE case. ξ_{lhs} and $\xi_{\text{lhs}} + 1$ correspond to the two indices in ξ along the left-hand-side boundary between the TF/SF region.

After solving the FD system to produce $\mathbf{\Psi}$, the two output field slices $\mathbf{F}_{\text{out}}$ are extracted as

$$\begin{aligned} \mathbf{F}_{\text{out}}[\zeta, \xi_{\text{rhs}}] &= \mathbf{\Psi}[\zeta, \xi_{\text{rhs}}] \\ \mathbf{F}_{\text{out}}[\zeta, \xi_{\text{rhs}} + 1] &= \mathbf{\Psi}[\zeta, \xi_{\text{rhs}} + 1] \end{aligned}, \tag{3.45}$$

where ξ_{rhs} and $\xi_{\text{rhs}} + 1$ are a physical length Δx to the right of ξ_{lhs} and $\xi_{\text{lhs}} + 1$ respectively.

Because long-range field propagation is typically narrow-angle, most of the resultant field solution will remain confined to a small cone bounded by z_{min} and z_{FD}. However, some objects may scatter fields at very steep angles. PMLs on the top and bottom of the FD region absorb such fields. As the broader problem is long-range, such steep-angle reflections would never reach the end of the range and can thus be safely ignored.

Once $\boldsymbol{\psi}_{\text{FD}}[x_0, \zeta]$ is known, the field is subsampled along ζ to produce $\boldsymbol{\psi}_{\text{FD}}[x_0 + \Delta x, \zeta]$.

D. Smoothly Blend Fields in Upper and Lower Regions

Now that $\boldsymbol{\psi}_{\text{FD}}[x_0 + \Delta x, z]$ and $\boldsymbol{\psi}_{\text{SSF}}[x_0 + \Delta x, z]$ are known for $z_{\text{min}} \le z \le z_{FD}$ and $z_{SSF} \le z \le z_{\text{max}}$ respectively, they are summed together over all $z_{\text{min}} \le z_{\text{max}}$ to produce the field $\psi(x_0 + \Delta x, z)$.

Upsampling and downsampling fields for the FD region when moving between slices can diminish accuracy as information is destroyed or approximated. This can be partially mitigated by directly using the output FD fields from the previous slice as input fields to the FD region in the

next slice. With this modification, upsampling only needs to be performed on fields that move from the SSF region downwards into the FD region, and downsampling only needs to be performed on fields that move from the FD region upwards into the SSF region. There is also freedom in whether the window functions are applied before or after marching by Δx.

Once $\psi(x_0 + \Delta x, z)$ is determined, steps 1-3 can be repeated to yield $\psi(x_0 + m\Delta x, z)$ for $m = 1, 2, 3...$ to construct the field solution for the entire computational domain.

E. Additional Enhancements

The solver can be augmented to accommodate backward-forward propagation, in the style of the two-way knife-edge propagation scheme described in [69]. As fields are marched sequentially through space, the back-propagating fields are saved at each slice. As the FD solver solves the full wave equation, both forward-propagating and backward-propagating fields are already stored in the FD region, so back-propagating fields can be extracted easily. By design, there are no back-scatterers in the SSF region, so no additional processing is needed in the SSF solver. Thus, back-scattered fields may be stored at each slice, and once the solver has completed one forward sweep from x_0 to $x_{\max}$, the scheme can be operated in the reverse direction, marching back the stored backscattered fields. Backward-forward sweeping may be repeated in this manner until convergence. This promotes the method to a full wave equation scheme.

3.4 Numerical Results

In this section, the hybrid solver will first be numerically validated for accuracy and stability by comparing reflection coefficients and plate scattering patterns to theory, as well as by analyzing sensitivity to perturbations in numerical parameters. The solver will subsequently be used to predict path loss over long ranges in rural environments.

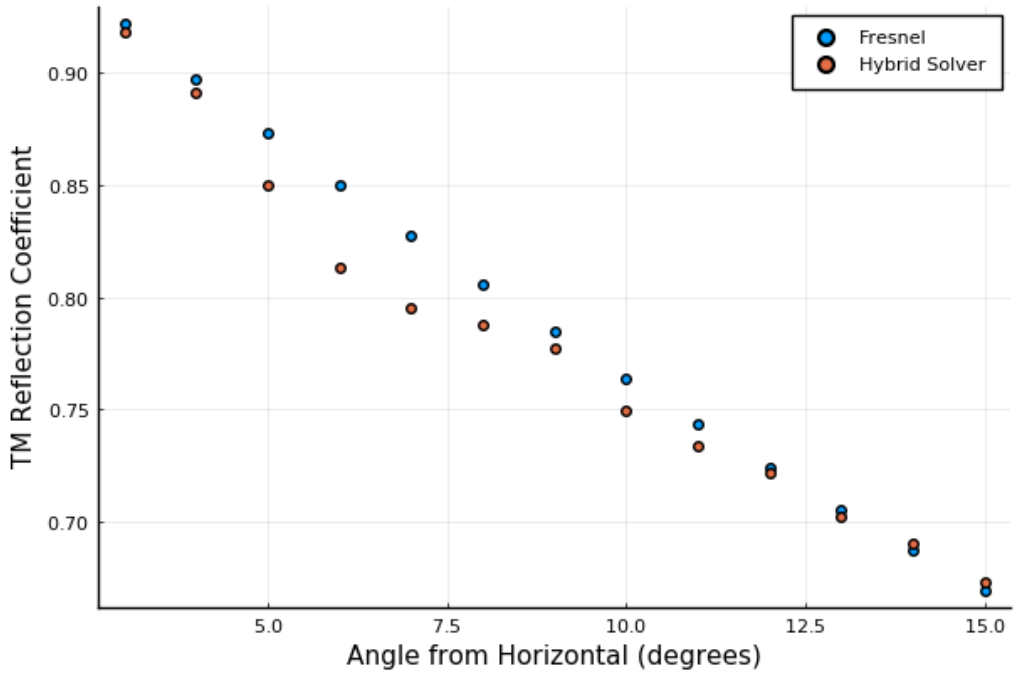
Several approximations are made in the interest of improving performance. The iterative solver used to solve FD systems is terminated after ten to twenty iterations. The Green's function matrices are also only constructed for every two to five indices; this, in effect, decreases the accuracy of the refractive index profile, but improves performance by not computing as many Green's function matrices.

A. Validation

The first numerical study concerns the reflection coefficient. The reflection coefficient is computed by launching a Gaussian beam at a flat surface and measuring the amplitude of the reflected field.

In order to account for the amplitude decay of a Gaussian beam, the reflected field is normalized against a Gaussian beam traveling an equivalent distance through free space. The reflection coefficient accuracy is studied for both the TM and TE cases.

The reflection coefficient study is performed at a frequency of 1 GHz. The domain is discretized with $\Delta z = 0.501\lambda$, $N_z = 65536$, and $\Delta x = 10$ m. The FD region is discretized with $h = 0.01$ m and $N_\zeta = 2675$ with 1875 points in the domain above the surface. The domain length in x is set to be long enough for the incident beam to reflect entirely from the terrain. The ground has permittivity $\epsilon_r = 2 - i$.



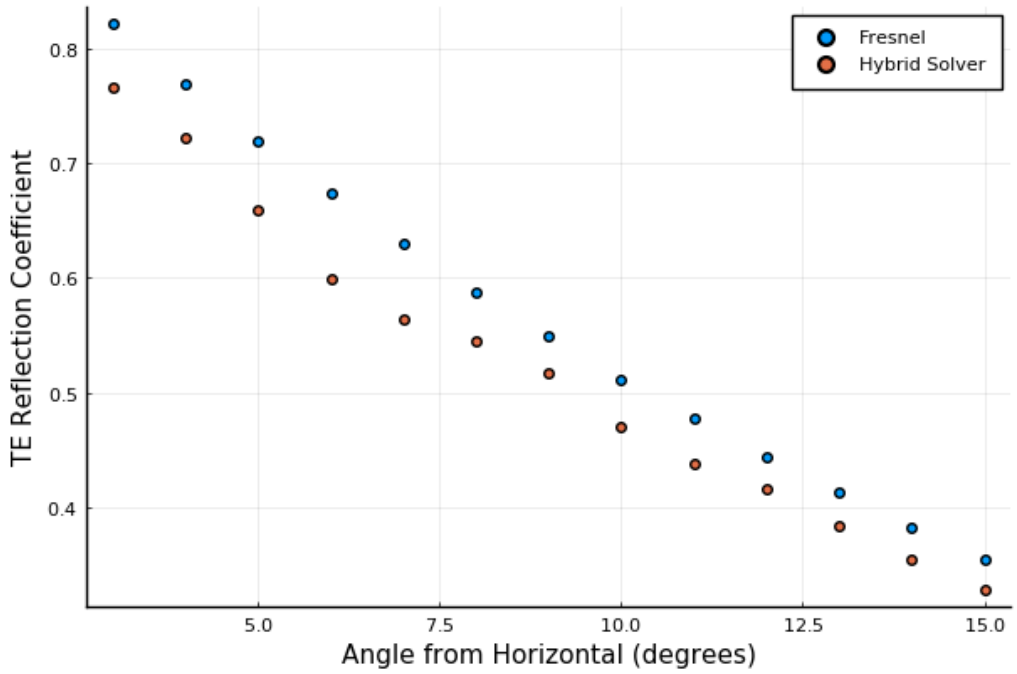


Fig. 3.7. Comparison of the theoretical Fresnel reflection coefficient and the experimental reflection coefficient produced by the solver, for TM (top) and TE (bottom) polarizations.

Fig. 3.7 shows very close agreement between the hybrid solver and the theoretical Fresnel reflection coefficient over a wide range of angles. The TM model, despite its increased complexity due to the FEM-inspired FD equation, is more accurate than the TE model; this experiment, in essence, validates the FEM framework for the TM discretization.

Numerical stability of the solver is demonstrated by modulating the discretization for a simple terrain scattering problem. A 100 MHz vertically-polarized Gaussian beam is launched at a hilly terrain characterized by $z_{\text{terrain}} = 50 + 10\sin(x/50)$, measured in meters. The SSF domain is discretized by $\Delta z = 0.501\lambda$, $N_z = 512$. The numerical parameters for the FD domain number of points above the surface, the SSF step size Δx, and the FD discretization h. The field strength from one example run is shown in

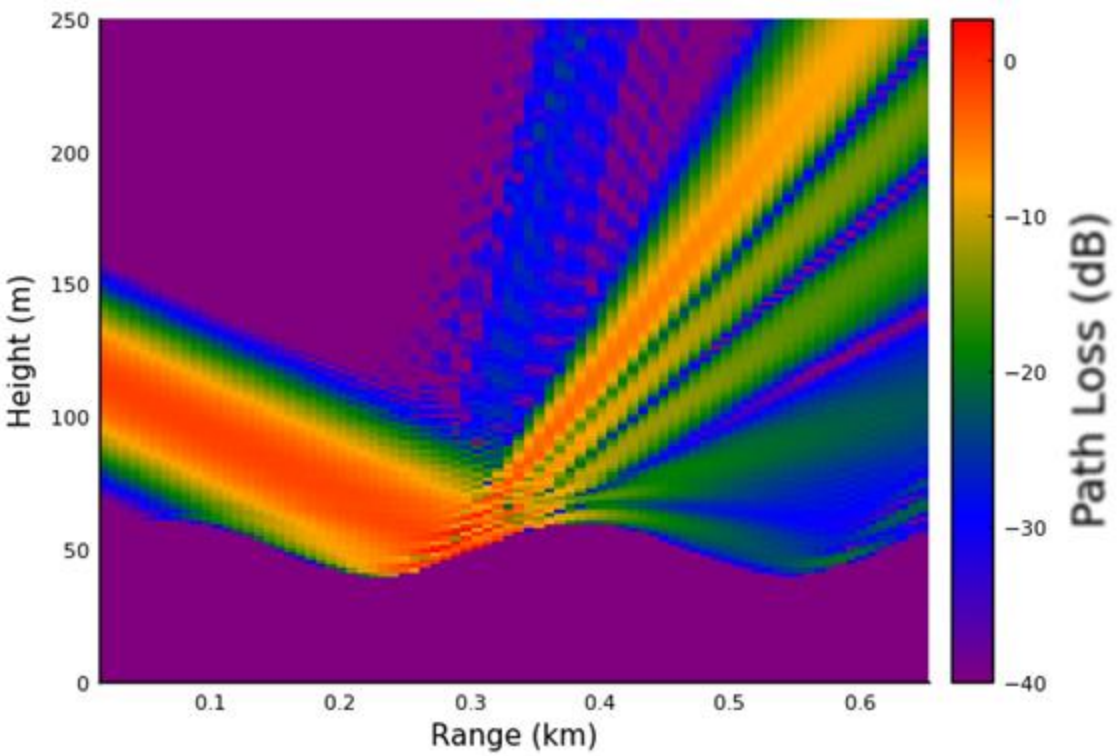


Fig. 3.8. Heatmap of path loss for one realization of scattering from a sinusoidal surface.

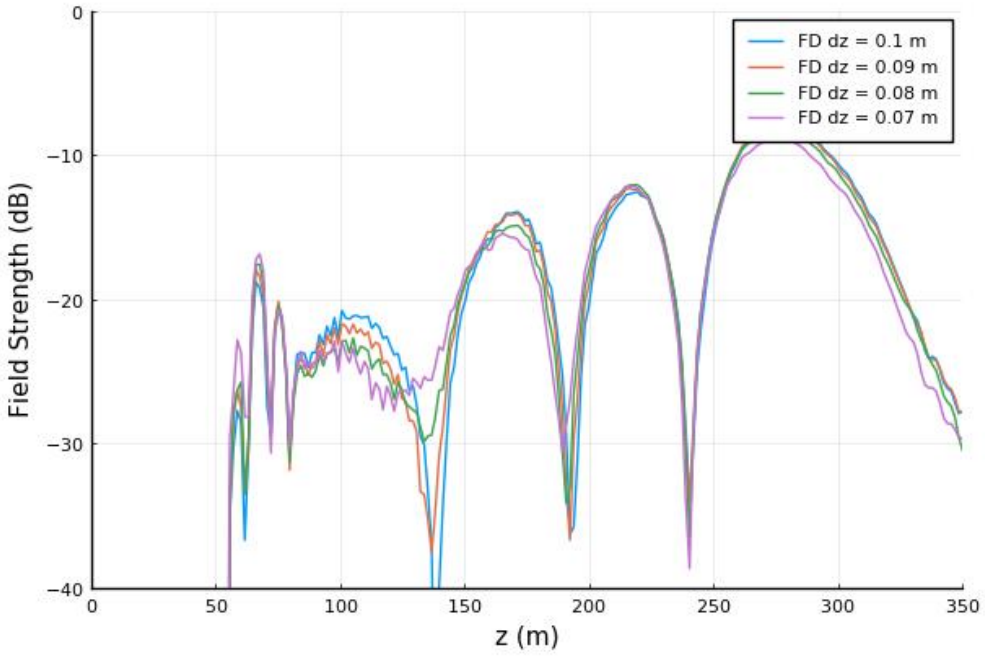


Fig. 3.9. Field strength along the last vertical slice in x for FD domain height above terrain = 175, FD step size $dz = h$ modulated, and $\Delta x = 10$ m.

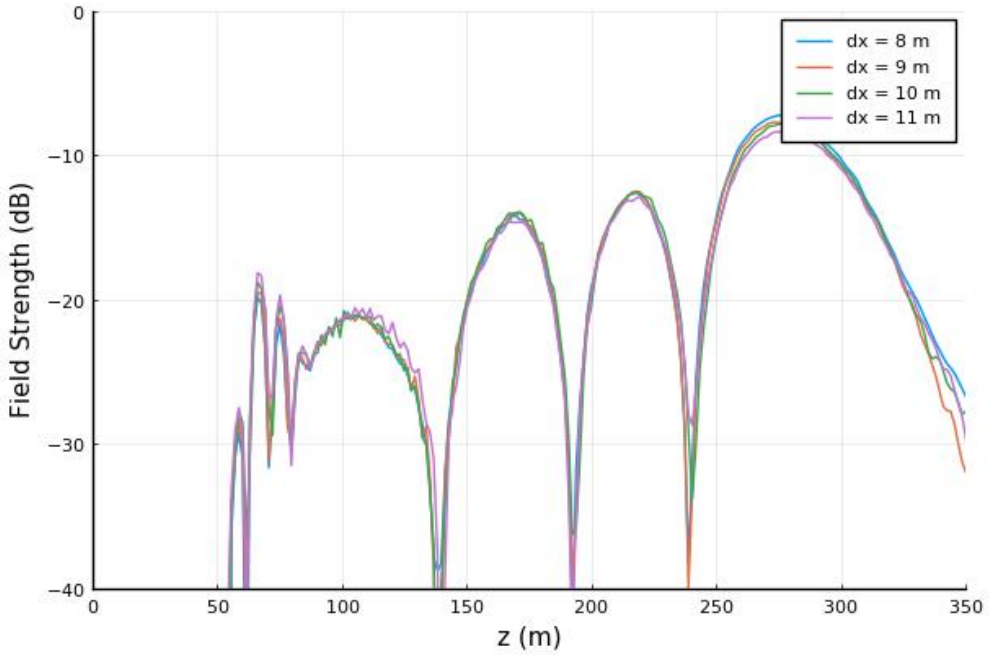


Fig. 3.10. Field strength along the last vertical slice in x for FD domain height above terrain = 175, FD step size $dz = h = 0.1$ m, and Δx modulated.

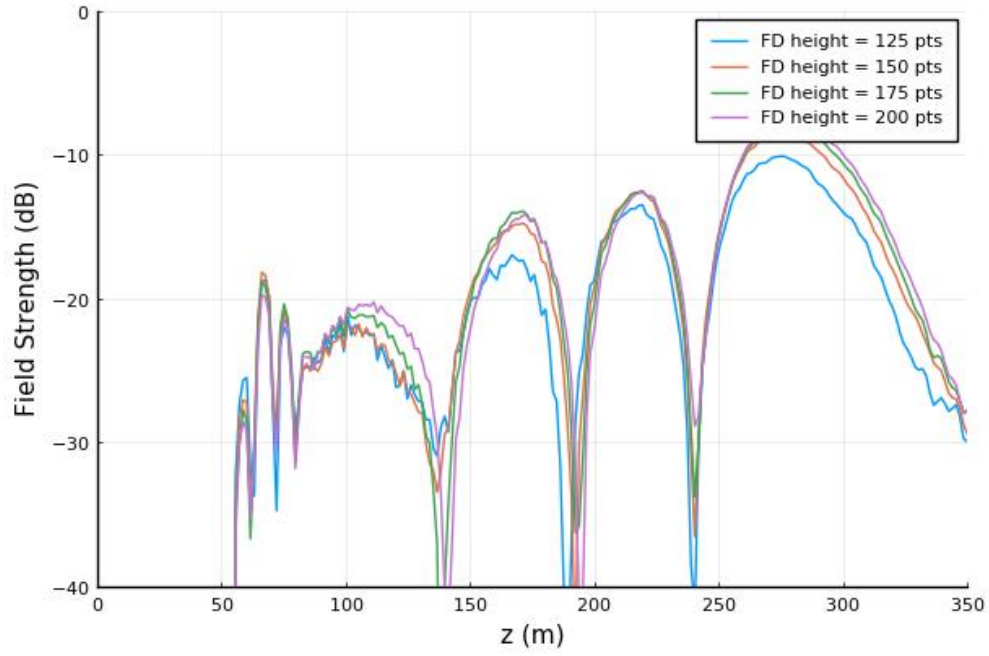


Fig. 3.11. Field strength along the last vertical slice in x for FD domain height above terrain modulated, FD step size $dz = h = 0.1$ m, and $\Delta x = 10$ m.

Numerical stability is observed in Fig. 3.9, Fig. 3.10, and Fig. 3.11. Under small perturbations of the discretization of the FD domain, the field slice remains negligibly changed. Convergence over FD height can be observed; in Fig. 3.11, increasing domain heights approach a consistent profile.

B. Propagation Studies

The first scenario studied is short-range propagation over a hilly, forested terrain. A 1 GHz transmitter emits fields propagating over 2 kilometers. The terrain has tree cover from 750 meters to 1 km, and the forest is modeled as an absorbing block with uniform permittivity. The transmitter has a half-power beamwidth of 6 degrees. It is mounted 2 meters from the ground and is aimed at the peak of the hill at 1 km.

The domain is discretized with $\Delta z = 0.501\lambda$, $N_z = 131072$, and $\Delta x = 10$ meters. The FD region is discretized with $h = \lambda/30$, and with 1875 points above the terrain surface and 800 points below the terrain surface.

The forest is modeled as a sequence of trees. A tree is represented by a rectangular canopy on top of a rectangular trunk. The permittivities of the trunk and canopy models were tuned such that the decay rate of fields through the forest matches that of existing empirical models [70] so that $\epsilon_{\text{trunk}} = 25 - 0.037i$, $\epsilon_{\text{canopy}} = 25 - 0.001i$. The ground is described by $\epsilon_{\text{ground}} = 2 - i$. Inside forest, the FD region discretization was rescaled by a factor of 4, such that $h = \lambda/120$ and the vertical domain size is 10700 points, chosen in order to keep the step size sufficiently small relative to the wavelength inside the forest.

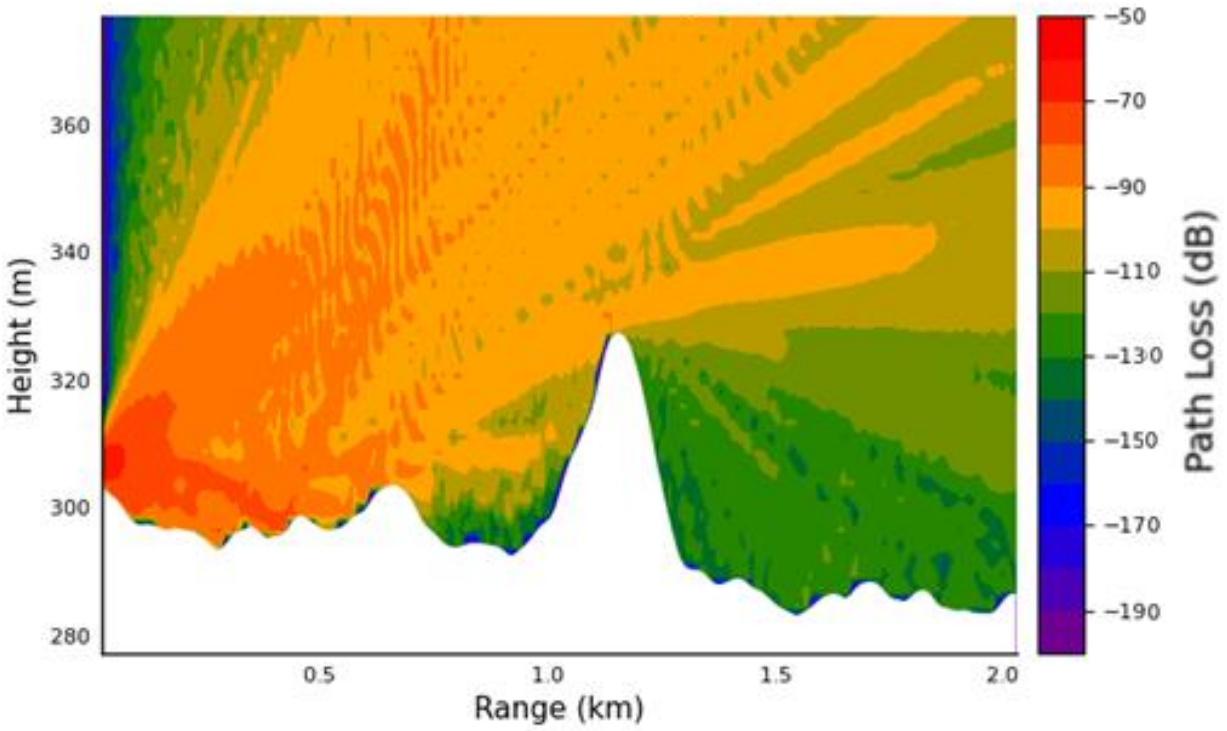


Fig. 3.12. Heatmap of field strength, in decibels, for the hilly forest terrain simulation.

Fig. 3.12 shows a beam spreading and scattering off of the hills from 0 to 1 kilometers. Fields diffract past the mountain at 1 kilometer, and a shadow region can be observed from 1.5 to 2 kilometers. Field attenuation can be observed at the forest from 750 to 1000 meters. The FD solver thus accurately accommodates scattering features and absorbing features. The simulation completed in 6 days on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated.

The second scenario is propagation over many kilometers of mountainous terrain. The terrain profile is extracted from geographic data from Eastern Washington. A 1 GHz vertically polarized transmitter 2 m above the ground oriented towards the peak at 15 km emits a Gaussian beam with a 6 degree HPBW. The ground is described by $\epsilon_{\text{ground}} = 2 - i$. The domain is discretized with $\Delta z = 0.501\lambda$, $N_z = 131072$, and $\Delta x = 10$ meters. The FD region is discretized with $h = \lambda / 30$, and with 1875 points above the terrain surface and 800 points below the terrain surface.

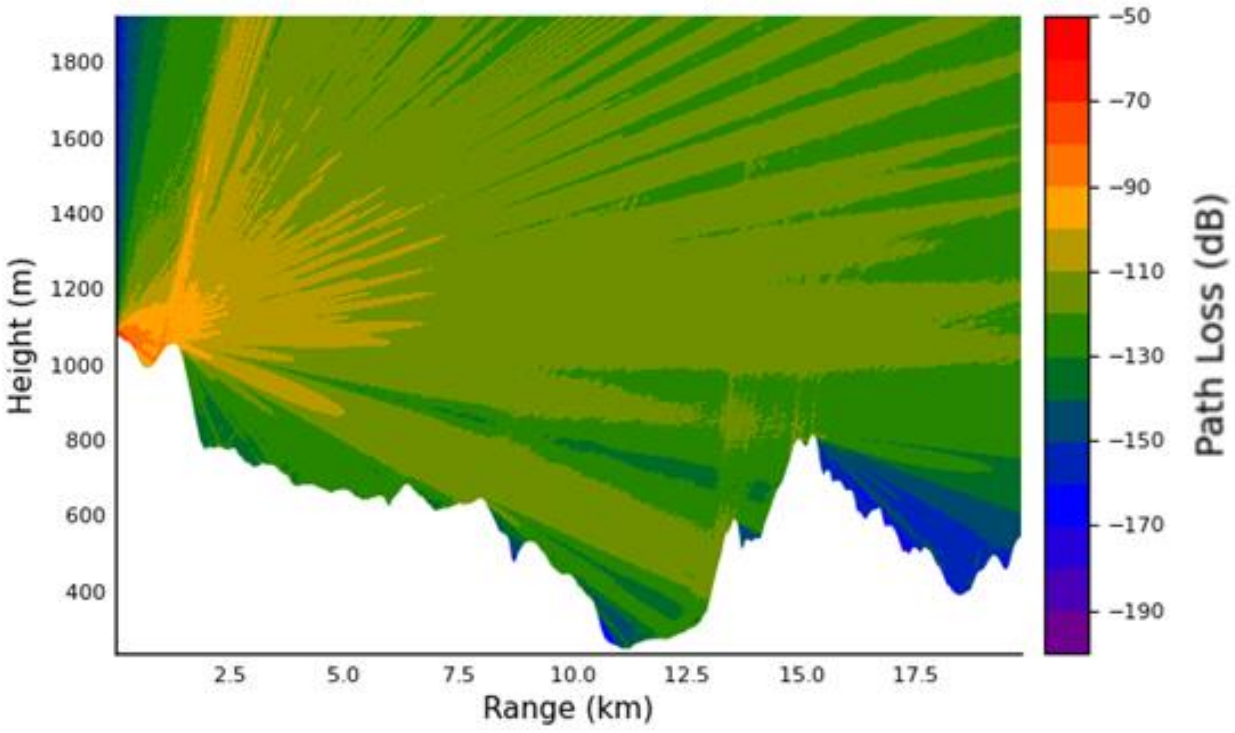


Fig. 3.13. Heatmap of field strength, in decibels, for the long-range Eastern Washington case

In Fig. 3.13, Fields spread outwards from the transmitter and two significant points of reflection and scattering are observed: One at the hill around 2 km, and another on the hillside at 13 km; fields are reflected towards the sky. Minor reflections can also be seen from the beam sidelobe grazing a peak at 7.5 kilometers. Deep shadow regions are observed past the mountain peaks, particularly between 15 and 20 kilometers. The simulation took 13 days to execute.

Chapter 4. Sparse Gabor Frame-Based Propagation in 3D

4.1 Introduction

This chapter generalizes the Gabor Frame-based propagator in chapter 1 from 2D to 3D. In 3D, the Split-Step Fourier (SSF) method marches fields from rectangular slice to rectangular slice through space, in an analogous manner to the 2D SSF method which marches a wavefront through successive 1D slices. The Gabor Transform framework introduced in chapter 1 will be generalized to 2D, and the SSF propagation operator will be represented in terms of a sparse Gabor-domain operator. Similarly, propagating fields will be described as sparse Gabor coefficient sets.

Phase screens will be implemented in the Gabor domain, and the Radiation Boundary Condition (RBC) method from chapter 1 will be implemented along the walls of the 3D domain. RBCs will prove useful in this chapter as they absorb radiation from horizontally spreading fields, a problem which has complicated previous implementations of 3D solvers.

4.2 3D Propagation with Split-Step Fourier

This section discusses propagation in three dimensions with the conventional Split-Step Fourier method. The 3D notation will be introduced, the geometry will be defined, and the 3D analogs of 2D propagation concepts will be discussed.

Consider a transmitter mounted above the ground in three-dimensional space. The transmitter emits fields that spread in 3D, interact with scatterers, and refract through the atmosphere. Space is modeled as a rectangular prism with a rectangular coordinate system (z, y, x). The transmitter produces a field excitation that is supplied along the $z-y$ plane at the $-x$ side wall of the domain, and fields are assumed to propagate in the $+x$ direction.

Consider some total Hertz potential vector quantity $\mathbf{\Pi}(z, y, x)$ that represents a field vector in 3D space. The Hertz potential can be fully described by decomposing into scalar components $\Pi_{ze}(z, y, x)$ and $\Pi_{zm}(z, y, x)$, corresponding to TE_z or TM_z modes respectively [5]. Let $\psi(z, y, x)$

represent one of the two Hertz scalar potentials $\mathbf{\Pi}_{ze}(z,y,x)$ or $\mathbf{\Pi}_{zm}(z,y,x)$, depending on the polarization being studied. $\psi(z,y,x)$ obeys the 3D Helmholtz equation, given by

$$\frac{\partial^2\psi(z,y,x)}{\partial x^2}+\frac{\partial^2\psi(z,y,x)}{\partial y^2}+\frac{\partial^2\psi(z,y,x)}{\partial z^2}+k_0{}^2n^2(z,y,x)\psi(z,y,x)=0, \tag{4.1}$$

where k_0 is the free space wavenumber, and $n(z,y,x)$ is the refractive index.

The domain is discretized by the vectors $\mathbf{z},\mathbf{y},\mathbf{x}$, with $0\le z\le N_z\Delta z$, $-\frac{N_y}{2}\Delta y+1\le y\le\frac{N_y}{2}\Delta y$, and $0\le x\le N_x\Delta x$ for all $z\in\mathbf{z}, y\in\mathbf{y}, x\in\mathbf{x}$, where the step size of each vector is $\Delta z,\Delta y,\Delta x$. Each point in space is represented by some triplet $z\in\mathbf{z}, y\in\mathbf{y}, x\in\mathbf{x}$.

The field quantity is discretized by $\mathbf{\Psi}[z,y,x]$. Let a field slice $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x]$ be a matrix with elements $\mathbf{\Psi}[z_i,y_j,x]$ for all $z_i\in\mathbf{z}, y_j\in\mathbf{y}$. Suppose a field slice $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0]$ is known at $x=x_0$; for example, if the fields are supplied by a transmitter. To express $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0+\Delta x]$ in terms of $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0]$, a Fourier-domain propagator will march fields through free space, and a spatial-domain phase screen will be used to handle the atmospheric propagation. If waves are assumed to propagate in only the $+x$ direction, then $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0+\Delta x]$ is given by

$$\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0+\Delta x]=\mathrm{IFFT}_{k_y,k_z}\{\mathbf{\Psi}[\mathbf{k_z},\mathbf{k_y},x_0]\exp(i\mathbf{K_x}[\mathbf{k_z},\mathbf{k_y}]\Delta x)\}\exp(ik\Delta x(\mathbf{n}[\mathbf{z}]-1)), \tag{4.2}$$

where $\mathbf{\Psi}[\mathbf{k_z},\mathbf{k_y},x_0]=\mathrm{FFT}_{z,y}\{\mathbf{\Psi}[\mathbf{z},\mathbf{y},x]\}$, $\mathbf{k_z},\mathbf{k_y}$ are the Fourier domain variables corresponding to $\mathbf{z},\mathbf{y}$, and the wavenumber matrix is given by $\mathbf{K_x}[k_z,k_y]=\sqrt{k^2-k_y{}^2-k_z{}^2}$. $\mathbf{n}[\mathbf{z}]$ is the refractive index, which is assumed to only vary in $\mathbf{z}$. The subscripts of the FFT and IFFT operators denote which variables the Fourier Transforms are computed over.

Equation (4.2) is used to repeatedly march $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x]$ from slice to slice for all $x\in\mathbf{x}$. The coordinate system, as well as the layout of the vertical slices, are visualized in Fig. 4.1.

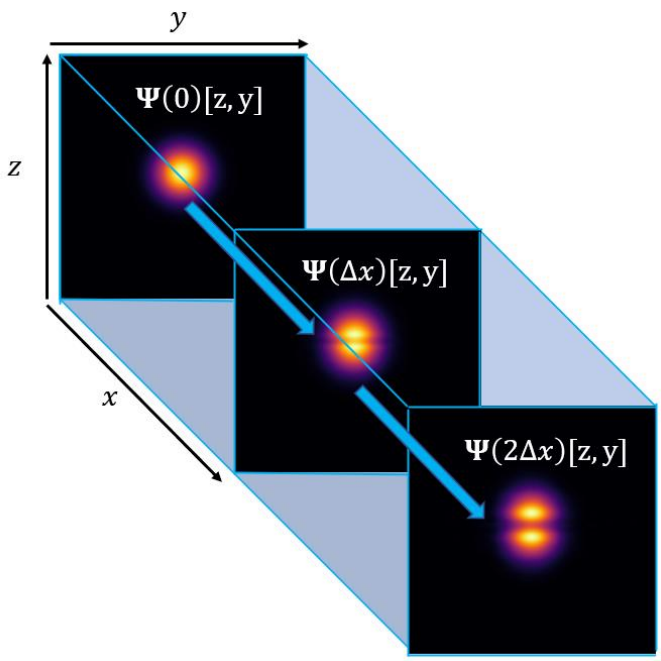


Fig. 4.1. Coordinate system for Split-Step Fourier in 3D. An initial wavefront defined along the $z-y$ plane, and is sequentially marched from slice to slice by Δx along the $+x$ direction. The vertical slices show a wavefront evolving through space.

4.3 2D Gabor Analysis

The 2D Gabor frame is an extension of the 1D Gabor frame. While in 1D, a vector is expressed as weighted sum of shifted and modulated window functions, in 2D a matrix is expressed as a weighted sum of shifted and modulated window matrices. There are now two shifting and modulation parameters, corresponding to shifts and modulations in z and y.

A 2D Gabor window matrix can be constructed by computing the outer product of two 1D Gabor windows [65] as

$$\mathbf{G}_{n1,n2,m1,m2}[\mathbf{z},\mathbf{y}] = \mathbf{g}_{n1,m1}[\mathbf{z}]\mathbf{g}_{n2,m2}[\mathbf{y}]^{\mathrm{T}}. \tag{4.3}$$

The spatial shift indices in z and y are $n_1 \in \mathbf{n}_1 = [0,...,N_1-1]$ and $n_2 \in \mathbf{n}_2 = [0,...,N_2-1]$ respectively, the frequency modulations z and y are $m_1 \in \mathbf{m}_1 = [0,...,M_1-1]$ and $m_2 \in \mathbf{m}_2 = [0,...,M_2-1]$ respectively, and the window widths in z and y are set to $N_{p1} = M_1$ and $N_{p2} = M_2$ respectively. Two examples of 2D Gabor window functions are shown in Fig. 4.2.

The dual window functions, an example of which is shown in Fig. 4.3, are constructed as outer products of 2D dual window functions, given by

$$\tilde{\mathbf{G}}_{n1,n2,m1,m2}[\mathbf{z},\mathbf{y}] = \tilde{\mathbf{g}}_{n1,m1}[\mathbf{z}]\tilde{\mathbf{g}}_{n2,m2}[\mathbf{y}]^T . \tag{4.4}$$

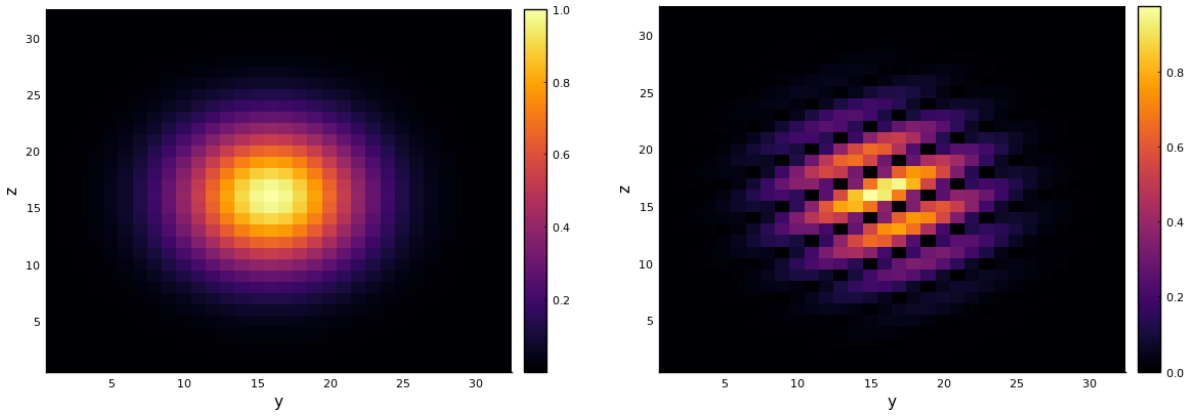


Fig. 4.2. Visualization of absolute value of real part of 2D Gabor window functions $\mathbf{G}_{n1,n2,m1,m2}[z,y]$, with $N_{p1} = N_{p2} = 32$, and with $(n_1,n_2,m_1,m_2) = (0,0,0,0)$ (left) and $(n_1,n_2,m_1,m_2) = (0,0,4,2)$ (right).

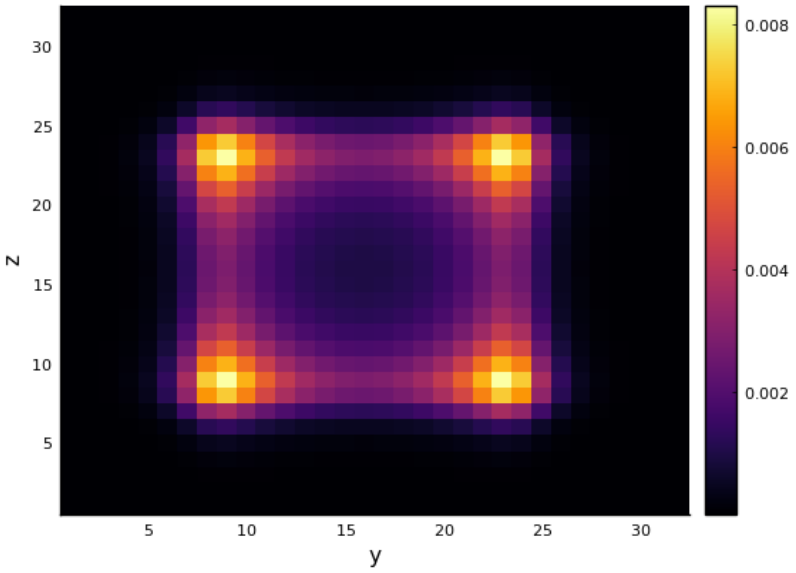


Fig. 4.3. 3D dual window function $\tilde{\mathbf{G}}_{n1,n2,m1,m2}[z,y]$, with $N_{p1} = N_{p2} = 32$ and $(n_1,n_2,m_1,m_2) = (0,0,0,0)$.

The matrix is written in terms of Gabor window functions as

$$\boldsymbol{\Psi}[\mathbf{z},\mathbf{y}] = \sum_{n_1,n_2}^{N_1,N_2} \sum_{m_1,m_2}^{M_1,M_2} \phi_{n1,n2,m1,m2} \mathbf{G}_{n1,n2,m1,m2}[\mathbf{z},\mathbf{y}]. \tag{4.5}$$

Equation (4.5) is called an Inverse 2D Gabor Transform, where each $\phi_{n1,n2,m1,m2}$ is a Gabor coefficient. A Gabor coefficient can be computed by performing an element-wise multiplication between $\tilde{\mathbf{G}}_{n1,n2,m1,m2}[\mathbf{z},\mathbf{y}]$ and $\boldsymbol{\Psi}[\mathbf{z},\mathbf{y}]$, and then summing together all elements of the resulting matrix. This operation can be written in terms of a vector-matrix-vector multiplication as

$$\phi_{n1,n2,m1,m2} = \tilde{\mathbf{g}}_{n1,m1}[\mathbf{z}]^{\mathrm{T}} \boldsymbol{\Psi}[\mathbf{z},\mathbf{y}] \tilde{\mathbf{g}}_{n2,m2}[\mathbf{y}]. \tag{4.6}$$

The collection of all Gabor coefficients $\phi_{n1,n2,m1,m2}$ is denoted as a Gabor coefficient set $\boldsymbol{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2]$.

In future sections, Gabor Transforms between a spatial domain matrix and a Gabor coefficient set will be given by the notation

$$\boldsymbol{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2] = \mathrm{GT}_{2\mathrm{D}}\{\boldsymbol{\Psi}[\mathbf{z},\mathbf{y}]\}, \tag{4.7}$$

$$\boldsymbol{\Psi}[\mathbf{z},\mathbf{y}] = \mathrm{IGT}_{2\mathrm{D}}\{\boldsymbol{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2]\}. \tag{4.8}$$

The computation of Gabor coefficients can be sped up by using Fourier Transforms. This is useful as not only a means of making some derivations more straightforward, but also because implemented Gabor Transform code can take advantage of optimized FFT libraries.

Let the Gabor coefficient set $\boldsymbol{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2]$ be rewritten as an "outer" spatial data structure $\boldsymbol{\Phi}^{space}[\mathbf{n}_1,\mathbf{n}_2]$ and an "inner" frequency data structure $\boldsymbol{\Phi}^{freq}{}_{n_1,n_2}[\mathbf{m}_1,\mathbf{m}_2]$, such that $\boldsymbol{\Phi}[n_1,n_2,\mathbf{m}_1,\mathbf{m}_2] = \boldsymbol{\Phi}^{space}[n_1,n_2] = \boldsymbol{\Phi}^{freq}{}_{n_1,n_2}[\mathbf{m}_1,\mathbf{m}_2]$. Each element of $\boldsymbol{\Phi}^{space}[\mathbf{n}_1,\mathbf{n}_2]$ can be assembled by multiplying unmodulated 2D Gabor windows with localized Fourier Transforms of the function $\boldsymbol{\Psi}[\mathbf{z}_{n_1,n_2},\mathbf{y}_{n_1,n_2}]$. The Fourier Transform is performed over the spatial support of each Gabor window function, given by $(\mathbf{z}_{n_1,n_2},\mathbf{y}_{n_1,n_2}) = \{z \in \mathbf{z}, y \in \mathbf{y} : \tilde{\mathbf{G}}_{n_1,n_2,m_1,m_2}[z,y] \neq 0\}$, which is of size $N_{p_1} \times N_{p_2}$. The computation in terms of Fourier Transforms is hence

$$\boldsymbol{\Phi}^{freq}{}_{n_1,n_2}[\mathbf{m}_1,\mathbf{m}_2] = \mathrm{FFT}_{z,y}\{\boldsymbol{\Psi}[\mathbf{z}_{n_1,n_2},\mathbf{y}_{n_1,n_2}] \odot \tilde{\mathbf{G}}_{n_1,n_2,0,0}[\mathbf{z}_{n_1,n_2},\mathbf{y}_{n_1,n_2}]\}, \tag{4.9}$$

where the symbol $\odot$ denotes element-wise multiplication of matrices. Likewise, the procedure for reconstructing $\mathbf{\Psi}[\mathbf{z},\mathbf{y}]$ is given by

$$\mathbf{\Psi}[\mathbf{z},\mathbf{y}] = \sum_{n_1,n_2}^{N_1,N_2} \mathbf{s}_{n_1,n_2}[\mathbf{z},\mathbf{y}]\mathbf{G}_{n1,n2,0,0}[\mathbf{z},\mathbf{y}] \odot \mathrm{IFFT}_{m_1,m_2}\{\mathbf{\Phi}^{freq}{}_{n_1,n_2}[\mathbf{m}_1,\mathbf{m}_2]\}, \tag{4.10}$$

where the indicator function $\mathbf{s}_{n_1,n_2}[z,y] = \begin{cases} 1, & z \in \mathbf{z}_{n_1,n_2}, y \in \mathbf{y}_{n_1,n_2} \\ 0, & \text{otherwise} \end{cases}$.

4.4 3D Propagation in the Gabor Domain

In this section, the Gabor framework for propagating fields in 3D is introduced. Fields will be decomposed into beamlets with precomputed propagation characteristics, and the beamlets will be marched in the Gabor domain.

The marching operator can be represented by a Gabor propagator in a similar way to prior sections on 2D. The field slice $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0]$ is represented in the Gabor domain by

$$\mathbf{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2,x_0] = \mathrm{GT}_{2\mathrm{D}}\{\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0]\}. \tag{4.11}$$

The notation $\phi_{n_1,n_2,m_1,m_2}[x_0]$ will sometimes be used to denote an individual element of the matrix $\mathbf{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2,x_0]$.

For 3D propagation, a beamlet $\mathbf{B}_{s_1,s_2,f_1,f_2}[\mathbf{z},\mathbf{y},x]$ is defined as the Split-Step Fourier solution to a problem with an initial excitation at $x = x_0$ of a 2D Gabor dual window function,

$$\mathbf{B}_{s_1,s_2,f_1,f_2}[\mathbf{z},\mathbf{y},x_0] = \tilde{\mathbf{G}}_{s_1,s_2,f_1,f_2}[\mathbf{z},\mathbf{y}]. \tag{4.12}$$

The propagation matrix set, similar to the 2D case, is assembled by marching (4.12) forward by Δx to form $\mathbf{B}_{s_1,s_2,f_1,f_2}[\mathbf{z},\mathbf{y},x_0+\Delta x]$ for all beamlet spatial shifts and frequency modulations s_1, s_2, f_1, f_2

$$\mathbf{P}[s_1,s_2,f_1,f_2,\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2] = \mathrm{GT}_{2\mathrm{D}}\{\mathrm{IFFT}_{k_y,k_z}\{\mathrm{FFT}_{y,z}\{\tilde{\mathbf{G}}_{s_1,s_2,f_1,f_2}[\mathbf{z},\mathbf{y}]\}e^{i\mathbf{K}_x\Delta x}\}\}. \tag{4.13}$$

The (s_1,s_2,f_1,f_2) element of the propagation matrix set is a Gabor coefficient set for a marched beamlet, where the beamlet has spatial and frequency modulations (s_1,s_2,f_1,f_2).

In order to propagate a wavefront $\boldsymbol{\Psi}[\mathbf{z},\mathbf{y},x_0]$, its Gabor coefficient matrix is obtained as per (4.11) ; the coefficient matrix of the field slice is be written as $\boldsymbol{\Phi}[\mathbf{s}_1,\mathbf{s}_2,\mathbf{f}_1,\mathbf{f}_2,x_0]$. For each element in the matrix $\boldsymbol{\Phi}[\mathbf{s}_1,\mathbf{s}_2,\mathbf{f}_1,\mathbf{f}_2,x_0]$, the propagation matrix set submatrix at (s_1,s_2,f_1,f_2) is added to the resultant fields, weighted by $\phi_{s_1,s_2,f_1,f_2}[x_0]$. This operation is represented mathematically as

$$\boldsymbol{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2,x_0+\Delta x]=\sum_{s_1,s_2,f_1,f_2}\mathbf{P}[s_1,s_2,f_1,f_2,\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2]\phi_{s_1,s_2,f_1,f_2}[x_0]. \qquad (4.14)$$

While the operator (4.14) is exact, it is extremely costly in memory. The propagation matrix $\mathbf{P}$ is an 8-dimensional structure; it is a 4-dimensional composition of Gabor representations of marched fields. In addition to memory costs, computing marched beamlets for every single index of s_1,s_2,f_1,f_2 is prohibitively slow; it requires a total of $N_1\times N_2\times N_{p_1}\times N_{p_2}$ marches. In order to implement a practical Gabor-domain 3D solver, efforts must be made to eliminate redundancies to reduce the number of stored dimensions of the propagation matrix.

The two dimensions representing spatial shifts can be easily eliminated. Provided the convention $\mathbf{g}_{n,m}[\mathbf{z}]=\mathbf{g}[\mathbf{z}-na]e^{i(mb)(\mathbf{z}-na)}$ is used, the s_1,s_2 elements of $\mathbf{P}$ can be simplified to frequency shifts, given by

$$\mathbf{P}[s_1,s_2,f_1,f_2,n_1,n_2,m_1,m_2]=\mathbf{P}[0,0,f_1,f_2,n_1-s_1,n_2-s_2,m_1,m_2]. \qquad (4.15)$$

With (4.15), fields need to be marched only $N_{p_1}\times N_{p_2}$ times in order to populate the propagation matrix. In order to eliminate one more dimension, some modification is needed to the SSF propagation operator (4.2).

The steps that follow reduce the computational cost of assembling $\mathbf{P}$ in two ways. The first is to facilitate shift-invariance in f_1 or f_2. The second, which aids the first, is to split the propagator into independent y and z marching steps, rather than mixed y,z marching. This means that not only can the transverse Gabor modulation term can be decoupled from the propagation matrix in each independent step, thus achieving shift-invariance in one frequency dimensions, but also the same propagator can be recycled between the y and z directions, thus requiring less computational overhead in building the propagation matrix set.

Recall that $\exp(-i\mathbf{K}_{\mathbf{x}}[k_z, k_y]\Delta x) = \exp(-i\sqrt{k^2 - {k_y}^2 - {k_z}^2}\Delta x)$. This term can not be split into a product of a k_y term and a k_z term. However, if the parabolic approximation is invoked, such a splitting becomes possible. The square root operator is approximated by

$$\exp(-i\sqrt{k^2 - {k_y}^2 - {k_z}^2}\Delta x) = \exp(-ik\sqrt{1 - \frac{{k_y}^2}{k^2} - \frac{{k_z}^2}{k^2}}\Delta x) \approx \exp(-ik(1 - \frac{{k_y}^2}{2k^2} - \frac{{k_z}^2}{2k^2})\Delta x)\,. \quad (4.16)$$

The right-hand side of (4.16) can be rewritten as

$$\exp(-ik(1 - \frac{{k_y}^2}{2k^2} - \frac{{k_z}^2}{2k^2})\Delta x) = \exp(-ik(\frac{1}{2} - \frac{{k_y}^2}{2k^2})\Delta x)\exp(-ik(\frac{1}{2} - \frac{{k_z}^2}{2k^2})\Delta x)\,. \quad (4.17)$$

Let the partial propagator $\boldsymbol{\rho}[k_i]$ be defined by

$$\boldsymbol{\rho}[k_i] = \exp(-ik(\frac{1}{2} - \frac{k_i^2}{2k^2})\Delta x)\,, \quad (4.18)$$

so that in free space, the SSF propagator (4.2) can be approximated as

$$\boldsymbol{\Psi}[\mathbf{z}, \mathbf{y}, x_0 + \Delta x] \approx \mathrm{IFFT}_{k_y, k_z}\{\mathrm{FFT}_{y,z}\{\boldsymbol{\Psi}[\mathbf{z}, \mathbf{y}, x_0]\boldsymbol{\rho}[\mathbf{k}_z]\boldsymbol{\rho}[\mathbf{k}_y]\}\}\,. \quad (4.19)$$

Note that with the k_y and k_z terms now being independent, the Fourier transforms in(4.19) can be split into Fourier transforms in separate directions z, k_z and y, k_y. The split propagator is given by

$$\begin{aligned}\boldsymbol{\Psi}_{\mathrm{aux}}[\mathbf{z}, \mathbf{y}, x_0] &= \mathrm{IFFT}_{k_z}\{\mathrm{FFT}_z\{\boldsymbol{\Psi}[\mathbf{z}, \mathbf{y}, x_0]\}\boldsymbol{\rho}[\mathbf{k}_z]\} \\ \boldsymbol{\Psi}[\mathbf{z}, \mathbf{y}, x_0 + \Delta x] &\approx \mathrm{IFFT}_{k_y}\{\mathrm{FFT}_y\{\boldsymbol{\Psi}_{\mathrm{aux}}[\mathbf{z}, \mathbf{y}, x_0]\}\boldsymbol{\rho}[\mathbf{k}_y]\}\end{aligned}\,. \quad (4.20)$$

If the domain is square such that $\mathbf{y} = \mathbf{z}$ and thus $\mathbf{k}_z = \mathbf{k}_y$, then the two propagation operators in (4.20) are identical. If a propagation matrix is built to represent the partial propagator $\boldsymbol{\rho}[k_i]$, then the propagation matrix set can be recycled for both steps in (4.20).

The utilization of the propagation matrix set for the partial propagator $\boldsymbol{\rho}[k_i]$ follows. Consider the propagation matrix $\mathbf{P}_z$ associated with the partial marching in the z direction in (4.20), given by

$$\mathbf{P}_z[s_1, s_2, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2] = \mathrm{GT}_{2\mathrm{D}}\{\mathrm{IFFT}_{k_z}\{\mathrm{FFT}_z\{\mathbf{g}_{s_1,f_1}[\mathbf{y}]\mathbf{g}_{s_2,f_2}[\mathbf{z}]\}\boldsymbol{\rho}[\mathbf{k}_z]\}\}. \quad (4.21)$$

Recalling the simplification (4.15), it is only necessary to write (4.21) for $s_1 = s_2 = 0$, so that

$$\mathbf{P}_z[0, 0, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2] = \mathrm{GT}_{2\mathrm{D}}\{\mathrm{IFFT}_{k_z}\{\mathrm{FFT}_z\{\mathbf{g}_{s_1,f_1}[\mathbf{y}]\mathbf{g}_{s_2,f_2}[\mathbf{z}]\}\boldsymbol{\rho}[\mathbf{k}_z]\}\}. \quad (4.22)$$

The Gabor window function can be expanded as $\mathbf{g}_{0,f_2}[y] = \mathbf{g}_{0,0}[y]\exp(i\frac{2\pi f_2 y}{N_{p_z}})$. Substituting into (4.22) yields

$$\begin{aligned}&\mathbf{P}_z[0, 0, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2] = \\ &\mathrm{GT}_{2\mathrm{D}}\{\mathrm{IFFT}_{k_z}\{\mathrm{FFT}_z\{\mathbf{g}_{0,0}[\mathbf{y}]\mathbf{g}_{s_2,f_2}[\mathbf{z}]\exp(i\frac{2\pi f_2 \mathbf{y}}{N_{p_2}})\}\boldsymbol{\rho}[\mathbf{k}_z]\}\}.\end{aligned} \quad (4.23)$$

The exponential operator can be pulled out of the Fourier transform, as it has no z dependence, so that

$$\begin{aligned}&\mathbf{P}_z[0, 0, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2] = \\ &\mathrm{GT}_{2\mathrm{D}}\{\exp(i\frac{2\pi f_2 \mathbf{y}}{N_{p_2}})\mathrm{IFFT}_{k_z}\{\mathrm{FFT}_z\{\mathbf{g}_{0,0}[\mathbf{y}]\mathbf{g}_{s_2,f_2}[\mathbf{z}]\}\boldsymbol{\rho}[\mathbf{k}_z]\}\}\end{aligned} \quad (4.24)$$

By using (4.24) in conjunction with properties of Gabor transforms,

$$\mathbf{P}_z[0, 0, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2] = \mathbf{P}_z[0, 0, f_1, 0, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2 - f_2] \quad (4.25)$$

Equation (4.25) shows that it is only necessary to precompute the propagation matrix $\mathbf{P}_z[0, 0, f_1, 0, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2]$, and all other elements $\mathbf{P}_z[s_1, s_2, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2]$ can be expressed as shifts of it at runtime.

The y propagation set $\mathbf{P}_y[s_1, s_2, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2]$ for the partial propagator $\boldsymbol{\rho}[\mathbf{k}_y]$ can be derived in the same manner. If the domain is square, it is unnecessary to precompute the y propagation set, as it can be expressed as a transpose of the z propagation matrix set as

$$\mathbf{P}_y[s_1, s_2, f_1, f_2, \mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2] = \mathbf{P}_z[s_2, s_1, f_1, f_2, \mathbf{n}_2, \mathbf{n}_1, \mathbf{m}_1, \mathbf{m}_2]. \quad (4.26)$$

It is therefore only necessary to perform N_{p_1} different marches during precomputation, and the other elements of the propagation matrix can be determined from shifts and transposes at runtime.

4.5 Sparsification in 3D

Gabor representations of 3D field slices can be sparsified to a very high degree. Sparsification is achieved by deleting coefficients in the Gabor coefficient set $\mathbf{\Phi}[\mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x]$ that fall below a certain threshold.

In implementation, it is most practical to only sparsify the "inner" frequency data structures $\mathbf{\Phi}^{freq}{}_{n_1,n_2,x}[\mathbf{m}_1, \mathbf{m}_2]$ of the Gabor coefficient set, so that the "outer" spatial data structure $\mathbf{\Phi}^{space}[\mathbf{n}_1, \mathbf{n}_2, x]$ is stored as a dense matrix.

Sparsification via simple thresholding is performed through the operation

$$\phi_{n_1,n_2,m_1,m_2}[x] = \begin{Bmatrix} \phi_{n_1,n_2,m_1,m_2}[x], & \left|\phi_{n_1,n_2,m_1,m_2}[x]\right| \geq \tau \left\|\mathbf{\Phi}[\mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x]\right\|_1 \\ 0, & \left|\phi_{n_1,n_2,m_1,m_2}[x]\right| < \tau \left\|\mathbf{\Phi}[\mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x]\right\|_1 \end{Bmatrix}. \tag{4.27}$$

The norm $\left\|\mathbf{\Phi}[\mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x]\right\|_1$ is obtained by summing together all elements, for all $(n_1, n_2, m_1, m_2) \in (\mathbf{n}_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2)$.

For optimally small storage, the sparsification operation (4.27) should be applied after any propagation operation; if the simple Gabor propagator (4.14) is used, fields should be sparsified once for every step in x. If the split propagation method (4.20) is used, an sparsification step is be required, albeit with a weaker tolerance, in between the z and y propagation steps of the split propagator. This is such that the second step of the split propagator will still operate over a sufficiently sparse matrix.

4.6 Radiation Boundary Conditions in 3D

The implementation of RBCs in 3D mimics the 2D implementation. Fields are marched on an auxiliary domain that is arbitrarily large, and excited Gabor coefficients outside of the domain are discarded. Let $\phi^{\text{aux}}{}_{n_1,n_2,m_1,m_2}[x_0 + \Delta x]$ be the result of marching $\phi_{n_1,n_2,m_1,m_2}[x_0]$ forward by one step on an arbitrarily large domain. The formula for truncating Gabor coefficients outside of the domain is

$$\phi_{n_1,n_2,m_1,m_2}[x_0 + \Delta x] = \begin{Bmatrix} \phi^{\text{aux}}{}_{n_1,n_2,m_1,m_2}[x_0 + \Delta x], & 1 \leq n_1 \leq N-1, 1 \leq n_2 \leq N-1 \\ 0, & \text{otherwise} \end{Bmatrix}. \tag{4.28}$$

While the typical use-case of this is upper atmosphere truncation on the upper $x-z$ wall, in 3D the RBCs can be implemented on the side $x-y$ walls as well, preventing sideways-moving radiation from reflecting back into the domain. This means that the width of the domain in y only needs to be enough to encapsulate significant environmental features and a thin absorbing layer to compensate for the relative weakness of Gabor RBCs at shallow incidence.

4.7 Gabor-Domain Phase Screens in 3D

The method for mixed-domain phase screens in 2D can be naturally carried forward to 3D. If the refractive index profile is piecewise linear, then if there are N_s unique slopes, then N_s+1 propagation matrix sets are generated; the first N_s matrix sets are marched under a refractive index profile with one of the unique slopes. The extra N_s+1th matrix comes from the splitting of propagation directions. If the split propagation method is used, some redundancy is lost as the phase screen only operates on k_z wavenumbers; k_y wavenumbers march through effectively free space. Therefore, if the phase screening terms are to be built into the propagation matrix set, an extra propagation matrix set must be generated for the free space case and used to march k_y wavenumbers. The operation of building the phase screening terms into the propagation matrix set, for a linear atmosphere with slope α, is given by

$$\begin{aligned}&\mathbf{P}_{\alpha z}[0,0,f_1,f_2,\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2]\\&=\mathrm{GT}_{2\mathrm{D}}\{\exp(ik\Delta x(\alpha\mathbf{z}-1))\mathrm{IFFT}_{kz}\{\mathrm{FFT}_z\{\mathbf{g}_{0,f_1}[\mathbf{z}]\mathbf{g}_{0,0}[\mathbf{y}]\}\boldsymbol{\rho}[\mathbf{k}_z]\}\}.\end{aligned}\tag{4.29}$$

In the 2D case, corners in the refractive index profile were handled by local Inverse/Forward Gabor Transforms; an inverse transform would be computed about a small set of spatial bins, the phase screen would be applied in the spatial domain, and a local forward transform would be computed. In 3D, this same process applies; all spatial indices $n_1,n_2\in\mathcal{N}_{inv}$ corresponding to corners in the refractive index profile are locally inversely transformed to the spatial domain and processed spatially. Much like 2D, a performance improvement is realized by, rather than computing a local Inverse/Forward Gabor Transform over all spatial bins, computing IFFTs/FFTs individually over the frequency content at each spatial bin. This is executed by constructing a diagonal operator $\mathbf{D}_{n_1,n_2}$ defined such that

$$\mathbf{D}\boldsymbol{\psi}[\mathbf{z}_{n_1},\mathbf{y}_{n_2}]=e^{-ik_0\Delta x(\alpha(n)\bullet(\mathbf{z}_n-na\Delta z)+\alpha(n)\bullet(n-1)-\mathbf{n}[\mathbf{z}_n])}\boldsymbol{\psi}[\mathbf{z}_{n_1},\mathbf{y}_{n_2}],\tag{4.30}$$

which operates over the vectors

$$\begin{aligned}\mathbf{z}_n &= [(n-1)a\Delta z, (n-1)a\Delta z + \Delta z, \ldots, na\Delta z - \Delta z] \subset \mathbf{z} \\ \mathbf{y}_n &= [(n-1)a\Delta z, (n-1)a\Delta z + \Delta z, \ldots, na\Delta z - \Delta z] \subset \mathbf{y}\end{aligned}. \tag{4.31}$$

Then for all n_1, n_2,

$$\mathbf{\Phi}^{freq}{}_{n_1,n_2,x_0}[\mathbf{m}_1,\mathbf{m}_2] = \begin{Bmatrix} \mathrm{FFT}^{n_1,n_2}{}_z\{\mathbf{D}_{n_1,n_2}\mathrm{IFFT}^{n_1,n_2}{}_{m_1}\{\mathbf{\Phi}^{freq}{}_{n_1,n_2,x_0}[\mathbf{m}_1,\mathbf{m}_2]\}, & n_1,n_2 \in \mathcal{N}_{jnv} \\ \mathbf{\Phi}^{freq}{}_{n_1,n_2,x_0}[\mathbf{m}_1,\mathbf{m}_2], & n_1,n_2 \notin \mathcal{N}_{inv} \end{Bmatrix}, \tag{4.32}$$

where

$$\mathrm{FFT}^{n_1,n_2}{}_z\{\mathbf{F}[\mathbf{z},\mathbf{y}]\}[m_1,m_2] = \mathrm{FFT}_z\{\mathbf{F}[\mathbf{z}_{n_1},\mathbf{y}_{n_2}]\} \tag{4.33}$$

and

$$\mathrm{IFFT}^{n_1,n_2}{}_{m_1}\{\hat{\mathbf{F}}[\mathbf{m}_1,\mathbf{m}_2]\}[z,y] = \begin{Bmatrix} \mathrm{IFFT}_{m_1}\{\hat{\mathbf{F}}[\mathbf{m}_1,\mathbf{m}_2]\}[z,y], & z \in \mathbf{z}_{n_1}, y \in \mathbf{y}_{n_2} \\ 0, & \text{otherwise} \end{Bmatrix}. \tag{4.34}$$

4.8 Numerical Results

The 3D solver is first validated for a simple beam propagating through refractive atmosphere and for a beam propagating through an ocean duct. The effects of sparsification in 3D are then studied, followed by a characterization of 3d Radiation Boundary Conditions.

A. Validation

The first test is a simple beam propagation scenario, synthesized to demonstrate the best-case utility of the solver for simple wave profiles in linear atmospheres. A Gaussian beam of width 25 meters, centered at $(z, y) = (936, 0)$ meters at 1 GHz, is launched through an atmosphere defined as an ionosphere with Earth curvature correction. The discretization is $\Delta y = \Delta z = 3\lambda/4$, $\Delta x = 300$ meters, $N_z = N_y = 8192$, $N_{p_x} = N_{p_y} = 1024$. Gabor RBCs are active, paired with an absorbing layer realized as an error function on a pedestal of width 368λ, tuned such that radiation incident at 0.5 degrees from horizontal is attenuated by 200 dB. The results are visualized in Fig. 4.4, Fig. 4.5, and Fig. 4.6.

Since the atmospheric profile is linear, the phase screen can be applied entirely in the Gabor domain, leading to a significant improvement in performance.

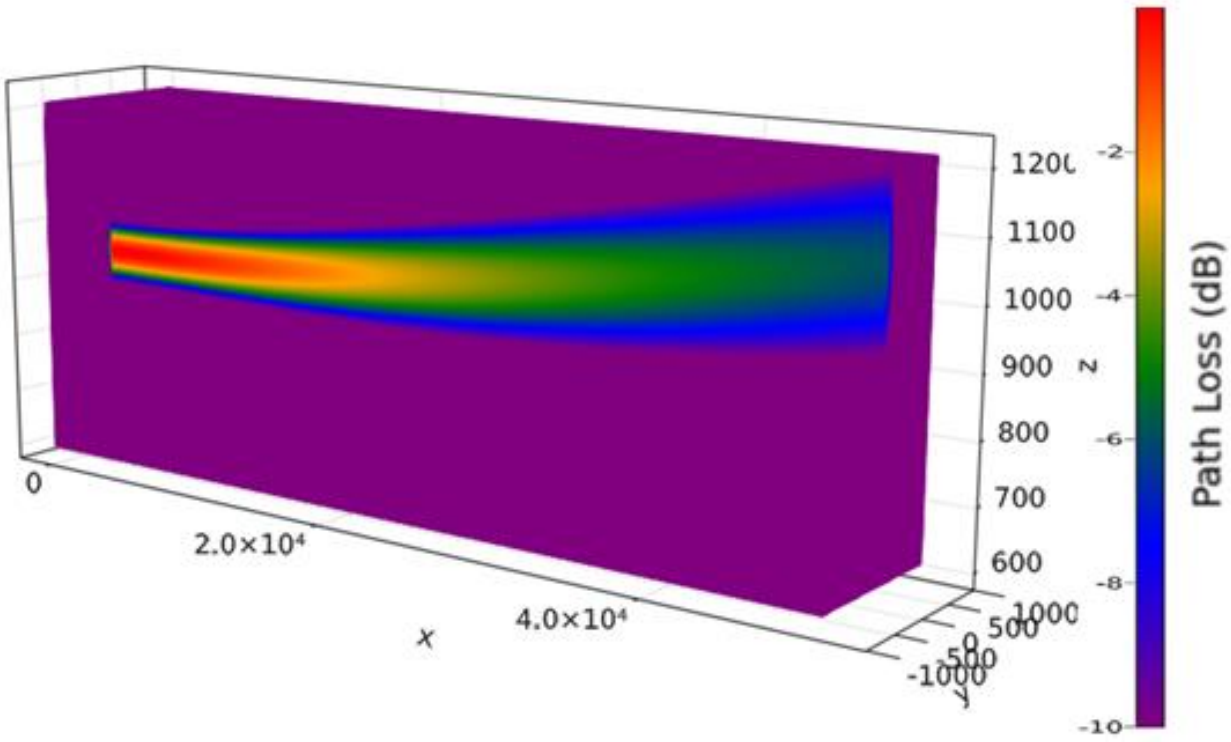


Fig. 4.4. Heatmap of the field strength (in dB) of a Gaussian beam propagating through a simple linear atmosphere.

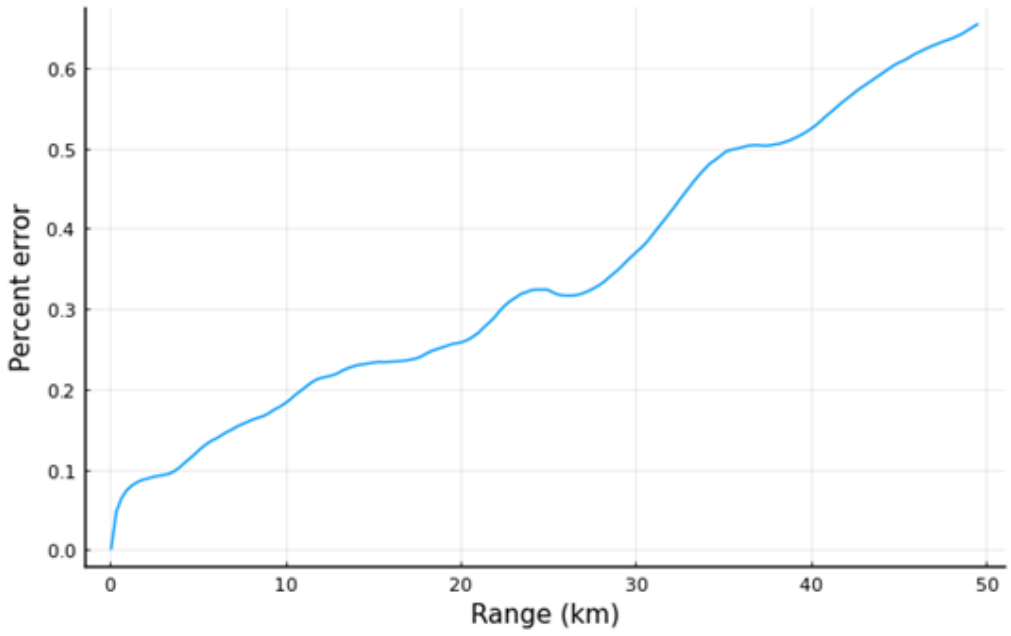


Fig. 4.5. Percent error of each slice in x for a Gaussian beam propagating through a simple linear atmosphere.

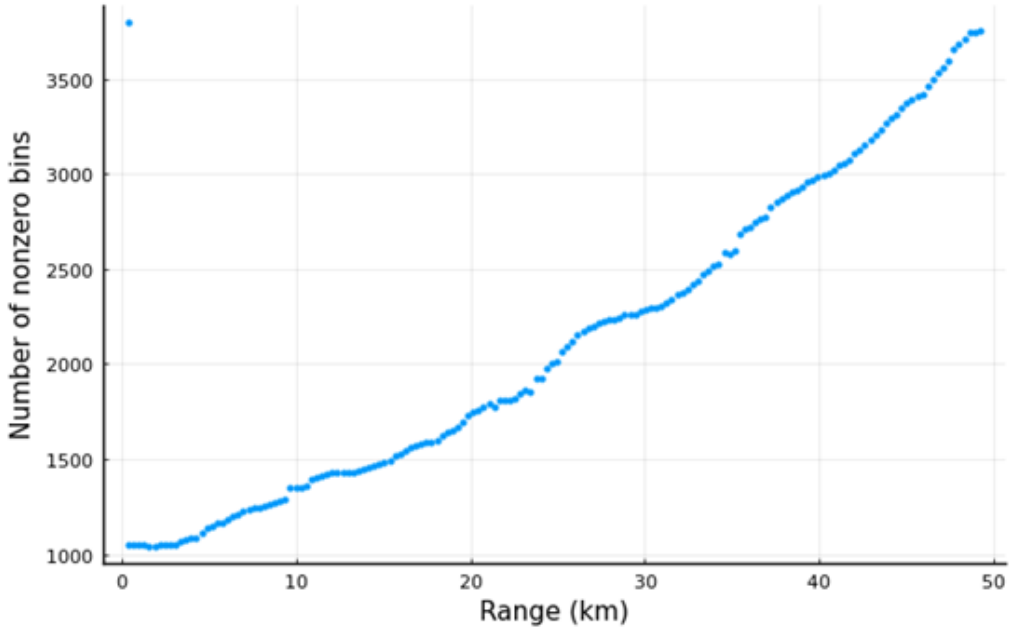


Fig. 4.6. Number of nonzero field coefficients at each slice in x, for a Gaussian beam propagating through a simple linear atmosphere.

Solver Type	Average Step Time (ms)	Peak Memory Usage (# points)
Split-Step Fourier	14400	67108864
Gabor	6545	3801

Table 4.1. Comparison of CPU time and number of stored points for the simulation in Fig. 4.4 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated.

Fig. 4.4 shows that the Gaussian beam smoothly deflects upwards as it propagates through space, as expected from a linear atmospheric profile. Fig. 4.5 and Fig. 4.6 show that the 3D Gabor solver represents propagating fields with fewer than 4000 stored coefficients and less than 1% error for a 8192×8192 domain; this amounts to a 99.99% sparsification. As shown in Table 4.1, Gabor solver marches fields and applies the phase screen at 2.2 times the speed of the Split-Step Fourier solver.

B. Long-Range Propagation

The next scenario is a more realistic case-study: A Gaussian beam propagating through an ocean duct. A Gaussian beam of width 25 meters, centered at $(z, y) = (926, 0)$ meters at 1 GHz, is launched through an atmosphere defined as an ionosphere with Earth curvature correction. The discretization is $\Delta y = \Delta z = 3\lambda / 4$, $\Delta x = 300$ meters, $N_z = N_y = 8192$, $N_{p_x} = N_{p_y} = 1024$. Gabor

RBCs are active, paired with an absorbing layer realized as an error function on a pedestal of width 368λ, tuned such that radiation incident at 0.5 degrees from horizontal is attenuated by 200 dB. The results are visualized in Fig. 4.7, Fig. 4.8, and Fig. 4.9.

The ocean duct is a trilinear profile with two unique slopes. The boundary condition of the ground is implemented as an image of the domain above the ground with reflection coefficient -1, thereby increasing the number of unique slopes in the computational problem to four. There are a total of five "corners" in the refractive index profile that must be locally processed in the spatial domain.

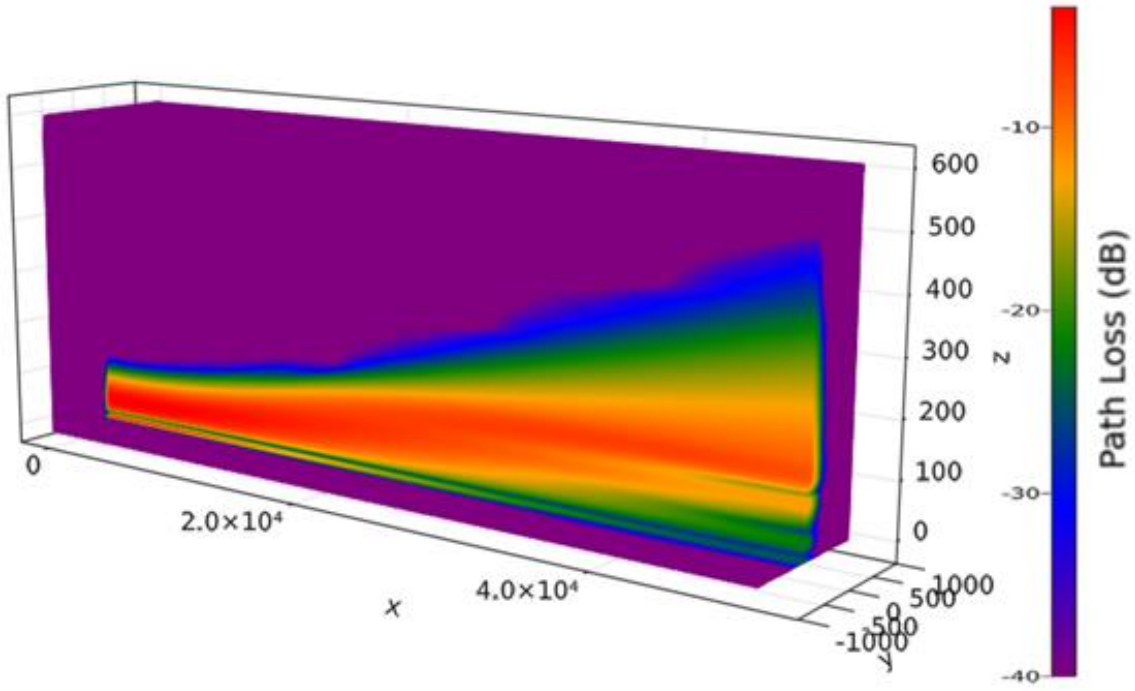


Fig. 4.7. Heatmap of the field strength (in dB) of a Gaussian beam propagating through a trilinear duct.

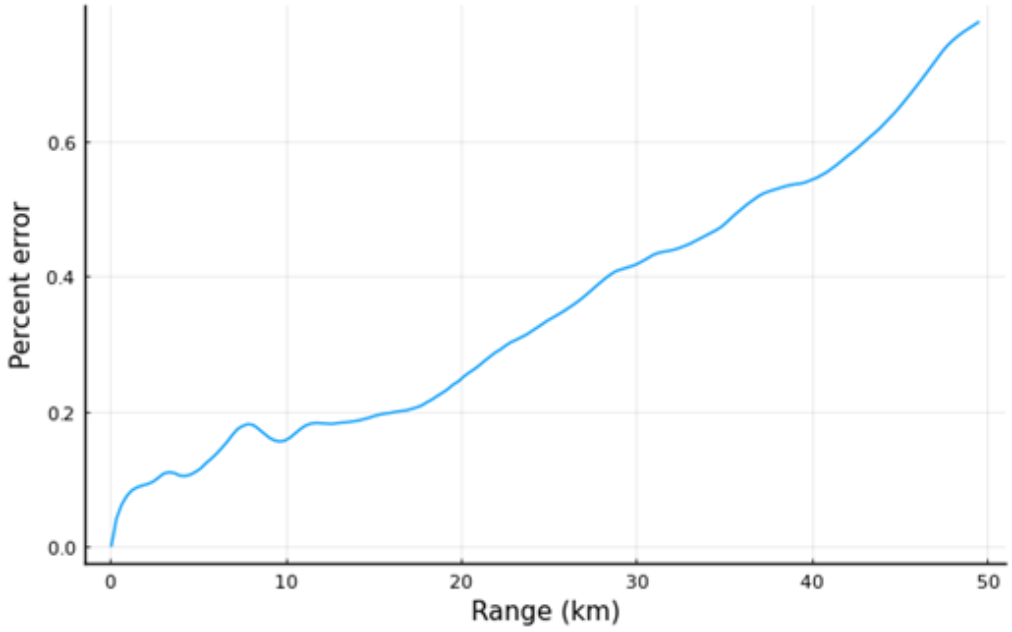


Fig. 4.8. Percent error of each slice in x for a Gaussian beam propagating through a trilinear duct.

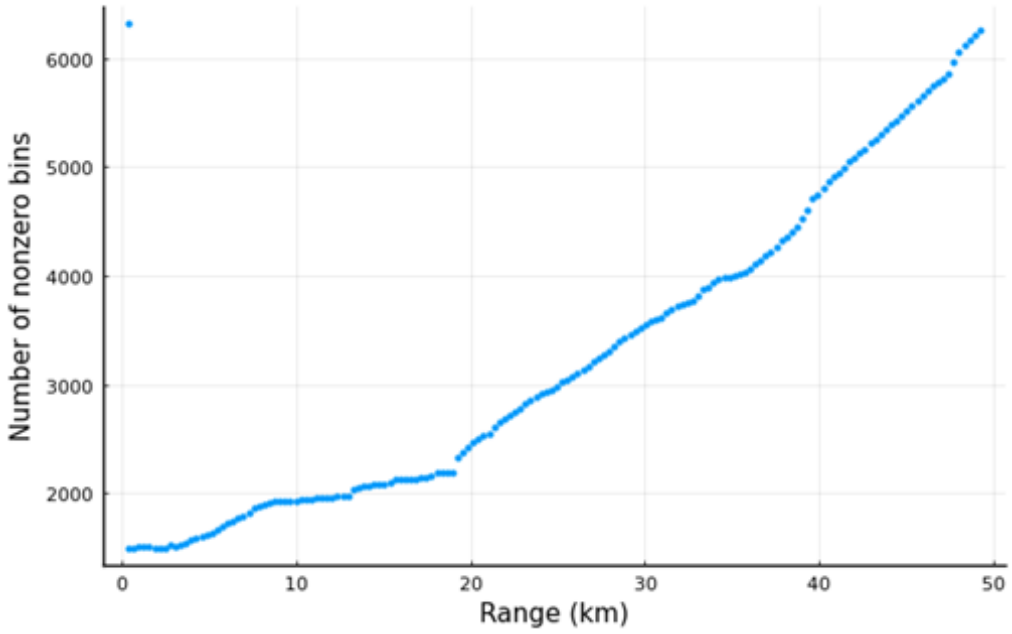


Fig. 4.9. Number of nonzero field coefficients at each slice in x, for a Gaussian beam propagating through a trilinear duct.

Solver Type	Average Step Time (ms)	Peak Memory Usage (# points)
Split-Step Fourier	12410	67108864
Gabor	10340	6572

Table 4.2. Comparison of CPU time and number of stored points for the simulation in Fig. 4.7 on a 3.0 GHz Intel Xeon Gold 6154 with one processor allocated.

Fig. 4.7 shows a Gaussian beam deflecting slightly upwards before leveling out, in accordance with energy getting trapped inside the duct. It can be seen in Fig. 4.8 and Fig. 4.9 that this scenario is represented efficiently within the Gabor framework; fewer than 6000 coefficients are stored with less than 1% error, which on a 8192×8192 domain means that the scenario has 99.99% sparsity. As shown in Table 4.2, the Gabor method performs 20% faster than Split-Step Fourier; the reduced performance relative to the simple linear atmosphere case is attributed to the additional processing needed for corners in the atmosphere.

C. Radiation Boundary Conditions

The 3D implementation of RBCs is characterized in the next experiment. The efficacy of the RBCs is measured through the reflection coefficient; a Gaussian beam is launched through free space at a side wall of the domain, and the reflected amplitude is compared with the incident amplitude.

Gaussian beams are launched at the $+y$ side wall of the domain, at which there is an RBC. The beams have a frequency of 1 GHz, have a beam waist of 10 m, and are centered at $z=-512\lambda, y=-512\lambda$. The domain is discretized with $\Delta z=\Delta y=\lambda$, $\Delta x=50$ m, $N_z=N_y=1024$, with Gabor window widths $N_{p_x}=N_{p_y}=128$.

Two cases are evaluated. The first case is strictly a Gabor-domain RBC. In the second case, two boundary conditions are used simultaneously: Gabor RBCs, and an absorbing layer. The absorbing layer is applied locally in the spatial domain. The layer is defined as an error function on a pedestal of width 204λ, tuned such that radiation incident at 0.5 degrees from horizontal is attenuated by 200 dB. The setup is visualized in Fig. 4.10 and Fig. 4.11, and the results are visualized in Fig. 4.12.

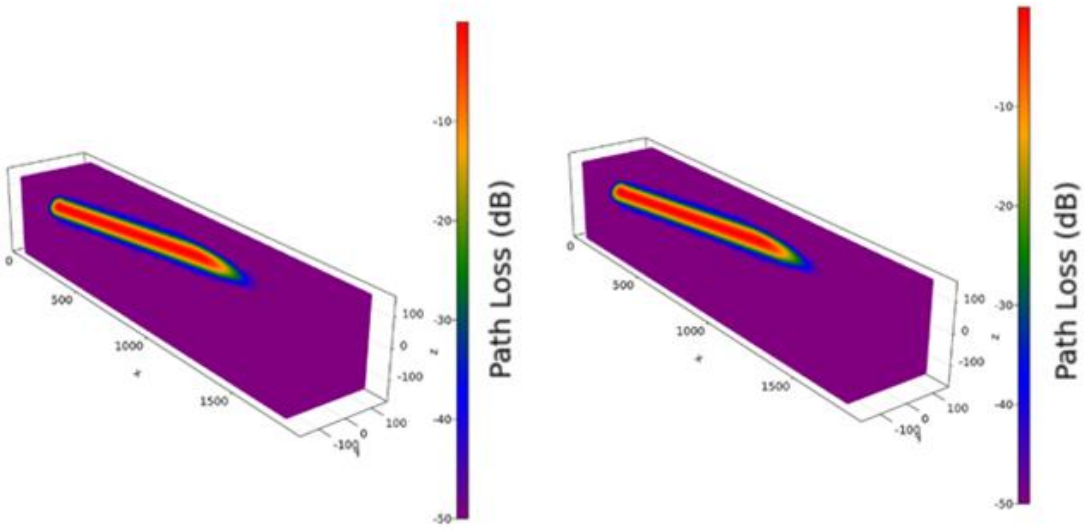


Fig. 4.10. Heatmap showing field strength for one realization of the experiment to characterize Gabor RBCs (left) and Gabor RBCs plus an absorbing layer (right), for a steep angle of incidence on the radiation surface.

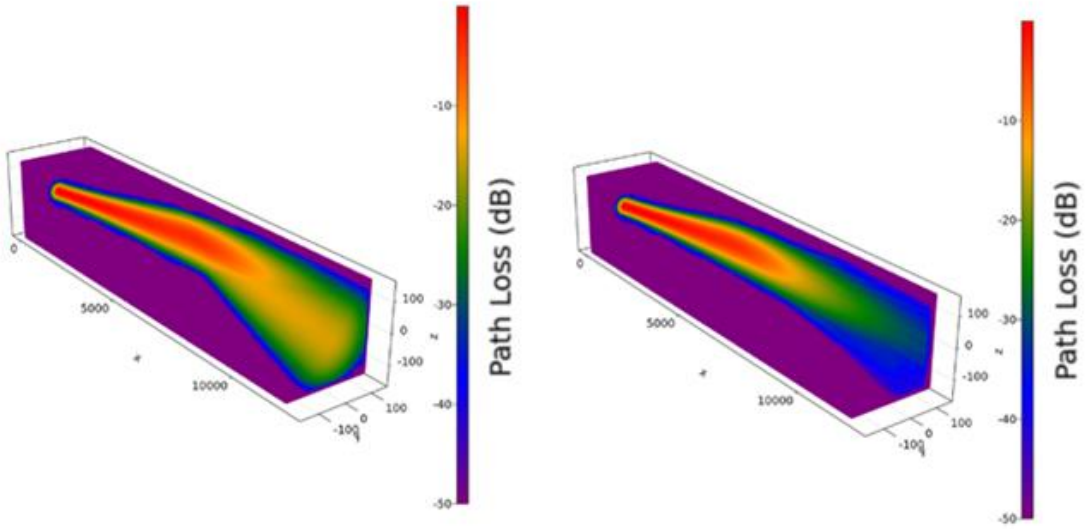


Fig. 4.11. Heatmap showing field strength for one realization of the experiment to characterize Gabor RBCs (left) and Gabor RBCs plus an absorbing layer (right), for a shallow angle of incidence on the radiation surface.

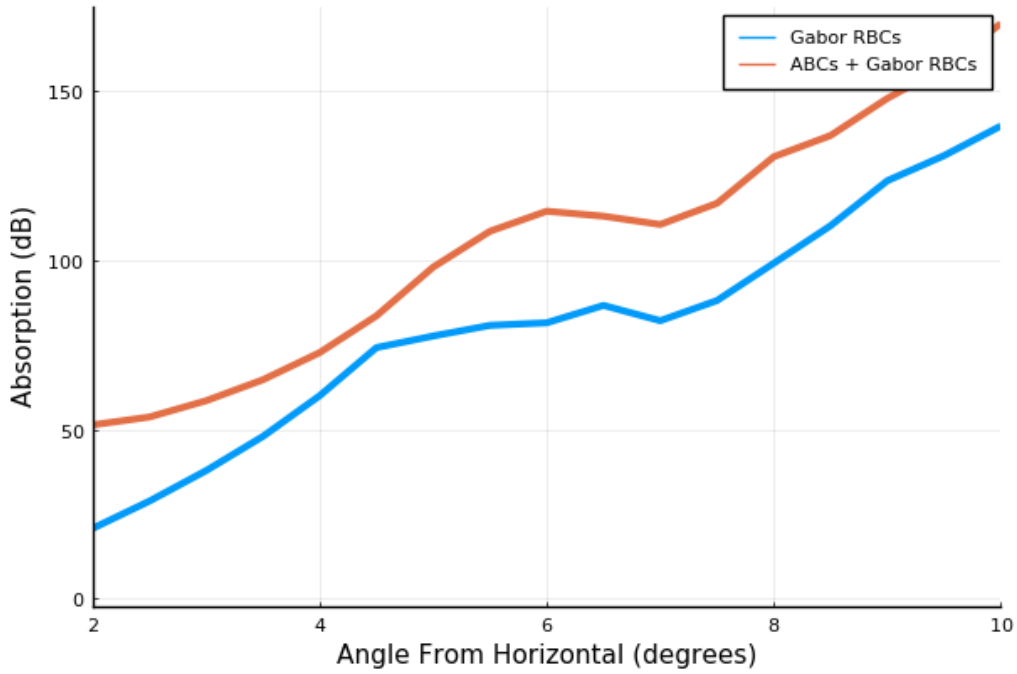


Fig. 4.12. Plot of the attenuation of two different boundary conditions versus incident angle: Gabor RBCs (blue), and Gabor RBCs with an added absorbing layer (orange).

The RBCs in 3D work perform similarly to the 2D case. Incident radiation at shallow angles is reflected strongly, but at steeper angles almost all incident radiation is absorbed. Since absorbing layers attenuate well for shallow angles, a thin absorbing layer helps compensate for the shortcomings of an RBC; a scheme using both Gabor RBCs and an absorbing layer achieves wide-angle attenuation.

D. Sparsity Effects

The last numerical experiment is a study of the effects of sparsification on accuracy. A Gaussian beam is marched through free space and is thresholded at different levels, and the associated error trends are compared. The thresholding is applied to both the propagation matrix set and the stored fields. The error is computed by comparing the 2-norm of the sparsified field at each location in x with that of a field with no sparsification.

The sparsified Gaussian beam has a frequency of 300 MHz and with beam waist 50 m, centered in the middle of the domain. The domain is discretized with $\Delta z = \Delta y = \lambda$, $\Delta x = 19$ m, and the Gabor window widths are $N_{p_z} = N_{p_y} = 64$. Gabor RBCs are active. One realization of the problem setup is shown in Fig. 4.13, and the results are shown in Fig. 4.14.

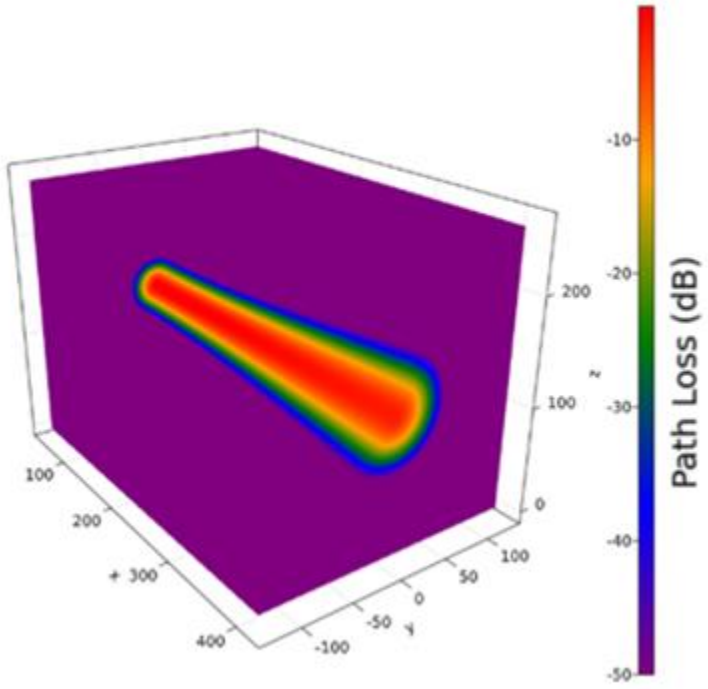


Fig. 4.13. Heatmap of the field strength of one realization of the sparsification study.

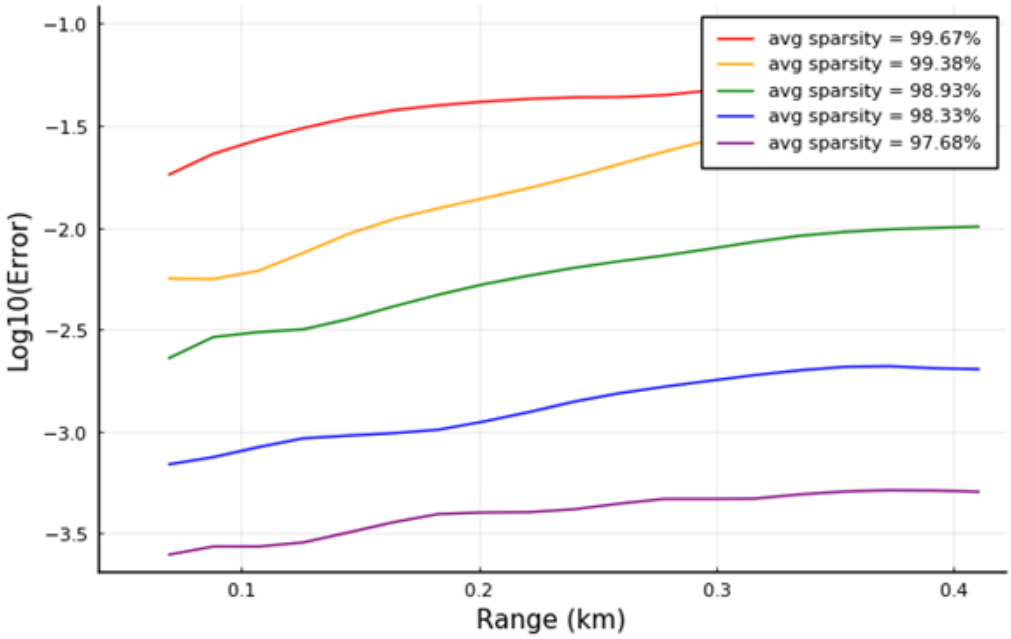


Fig. 4.14. Error induced by sparsification of a Gaussian beam at each vertical slice. Each line represents a different level of sparsification.

Fig. 4.14 shows that increasing sparsification increases error. It can be seen that sparsification levels above 95% are extremely easy to achieve while maintaining low error. The error increases with distance, so care must be taken in problem definitions such that the error remains low over long ranges.

Chapter 5. Terrain Modeling in 3D

5.1 Introduction

Terrain modeling is critical for a robust 3D solver. There is strong interest in the description of propagation over terrain, and while there is a robust array of 2D solvers, they are significantly limited in their accuracy. 2D models neglect the effects of objects oriented such that scattered fields are perpendicular to the vertical and range direction. These effects can by accurately described by using a 3D terrain model; however, while 3D terrain models exist for Parabolic Wave Equation solvers, their scalability is suboptimal due to the resources required to describe structured fields in the atmosphere above the terrain.

This chapter adds a terrain model to the 3D solver developed in chapter 4. The Gabor frame-based solver will be integrated with an Impedance Boundary Condition (IBC). In a manner inspired by the hybrid SSF-FD solver in chapter 3, the computational domain is partitioned into an upper region and a lower region. Fields in the upper region are marched with the sparse Gabor solver, and fields in the lower region are marched with a SSF solver augmented with IBCs. Local Inverse Gabor Transforms are used to elegantly hybridize the solvers.

5.2 Impedance Boundary Conditions

This section describes how fields can be marched under an Impedance Boundary Condition using the 3D Split-Step Fourier framework. IBCs enable the reflection of fields from hard surfaces that have a specified permittivity. This section largely recounts the work of [54], which implemented IBCs for propagation over forest.

Consider a 3D field profile $\mathbf{\Psi}[\mathbf{z}, \mathbf{y}, x_0]$ over terrain that has permittivity $\epsilon_{\text{terrain}}$. The terrain elevation at each point (y, x_0) is given by $z_{\text{terrain}} = \mathbf{h}[y, x_0]$. At the terrain interface, $\mathbf{\Psi}[\mathbf{z}, \mathbf{y}, x_0]$ obeys the Impedance Boundary Condition

$$\begin{aligned} &\frac{\partial \psi_{\mathrm{TE}}}{\partial z} + \frac{ik_0}{\sqrt{\epsilon_{\mathrm{terrain}}}}\psi_{\mathrm{TE}} = 0 \\ &\frac{\partial \psi_{\mathrm{TM}}}{\partial z} + ik_0\sqrt{\epsilon_{\mathrm{terrain}}}\psi_{\mathrm{TM}} = 0 \end{aligned} . \tag{5.1}$$

The propagation equation through weakly inhomogeneous space, as explored in the previous chapter, is given by

$$\boldsymbol{\Psi}[\mathbf{z},\mathbf{y},x_0+\Delta x] = \mathrm{IFFT}_{k_y,k_z}\{\boldsymbol{\Psi}[\mathbf{k_z},\mathbf{k_y},x_0]\exp(i\mathbf{K_x}[\mathbf{k_z},\mathbf{k_y}]\Delta x)\}\exp(ik\Delta x(\mathbf{n}[\mathbf{z}]-1)) . \tag{5.2}$$

In order to march fields over terrain, equation (5.2) will be modified to accommodate IBCs. For continuous scalar field quantities ψ_{TE} and ψ_{TM} for TE and TM polarizations respectively, the IBCs are

$$\begin{aligned} &\frac{\partial \psi_{\mathrm{TE}}}{\partial z} + \frac{ik_0}{\sqrt{\epsilon_{\mathrm{terrain}}}}\psi_{\mathrm{TE}} = 0 \\ &\frac{\partial \psi_{\mathrm{TM}}}{\partial z} + ik_0\sqrt{\epsilon_{\mathrm{terrain}}}\psi_{\mathrm{TM}} = 0 \end{aligned} . \tag{5.3}$$

First, the Fourier Transform of the field profile $\boldsymbol{\Psi}[\mathbf{z},\mathbf{y},x_0]$ subject to IBCs is computed as

$$\boldsymbol{\Psi}[k_z,k_y,x_0] = \int_{z_{\mathrm{terrain}}}^{\infty}\int_{-\infty}^{\infty}\boldsymbol{\Psi}[z,y,x_0]e^{-ik_y y}(e^{-ik_z z}+\Gamma(k_z)e^{ik_z z})dydz , \tag{5.4}$$

where $\Gamma(k_z)$ is the terrain reflection coefficient as a function of the incident wavenumber k_z. The reflection coefficient in TE and TM modes is given by

$$\Gamma_{\mathrm{TE}}(k_z) = \frac{\frac{k_{1z}}{k_1\sqrt{\epsilon_r}} - \frac{k_z}{k_0}}{\frac{k_{1z}}{k_1\sqrt{\epsilon_r}} + \frac{k_z}{k_0}}, \qquad \Gamma_{\mathrm{TM}}(k_z) = \frac{\frac{k_1}{k_{1z}\sqrt{\epsilon_r}} - \frac{k_0}{k_z}}{\frac{k_1}{k_{1z}\sqrt{\epsilon_r}} + \frac{k_0}{k_z}}, \tag{5.5}$$

where k_{1z} is the z component of the wavenumber inside the terrain, and k_1 is the wavenumber below the terrain interface.

Equation (5.4) can be written in terms of a Fast Fourier Transform by splitting the field profile into even and odd components with respect to z_{terrain}, denoted by $\boldsymbol{\Psi}_e[z, y, x_0]$ and $\boldsymbol{\Psi}_o[z, y, x_0]$ respectively. By using the property

$$\begin{aligned}\int_0^{\infty}(e^{-ik_z z}+\Gamma(k_z)e^{ik_z z})\boldsymbol{\psi}[z,y,x]dz &= \int_{-\infty}^{\infty} e^{-ik_z z}\frac{1+\Gamma(k_z)}{2}\boldsymbol{\psi}_e[z,y,x]dz \\ &+\int_{-\infty}^{\infty} e^{-ik_z z}\frac{1-\Gamma(k_z)}{2}\boldsymbol{\psi}_o[z,y,x]dz\end{aligned}, \tag{5.6}$$

(5.4) can be rewritten as

$$\boldsymbol{\Psi}[k_z,k_y,x_0]=\frac{1+\Gamma(k_z)}{2}\boldsymbol{\Psi}_e[k_z,k_y,x_0]+\frac{1-\Gamma(k_z)}{2}\boldsymbol{\Psi}_o[k_z,k_y,x_0], \tag{5.7}$$

where $\boldsymbol{\Psi}_e[k_z,k_y,x_0]$ and $\boldsymbol{\Psi}_o[k_z,k_y,x_0]$ are the Fast Fourier Transforms of $\boldsymbol{\Psi}_e[z,y,x_0]$ and $\boldsymbol{\Psi}_o[z,y,x_0]$ respectively. After evaluating (5.7), the resultant $\boldsymbol{\Psi}[k_z,k_y,x_0]$ can be substituted into the SSF propagator (5.2). Thus, in order to march fields subject to an IBC, (5.7) is invoked in the SSF marching step in place of the usual FFT computations of spatial-domain field slices.

5.3 Hybridization of Gabor Propagation and IBCs

The 3D Gabor propagation scheme outlined in the previous chapter can be elegantly hybridized with the IBC-augmented SSF method by using local Inverse Gabor Transforms. Fields in most of the domain are marched as usual with the Gabor framework, however fields about the surface are locally transformed to the spatial domain and marched with the propagator derived in section 5.2. This can be visualized as a partitioning of the domain into a “Gabor region” and an “IBC region”, shown in Fig. 5.1.

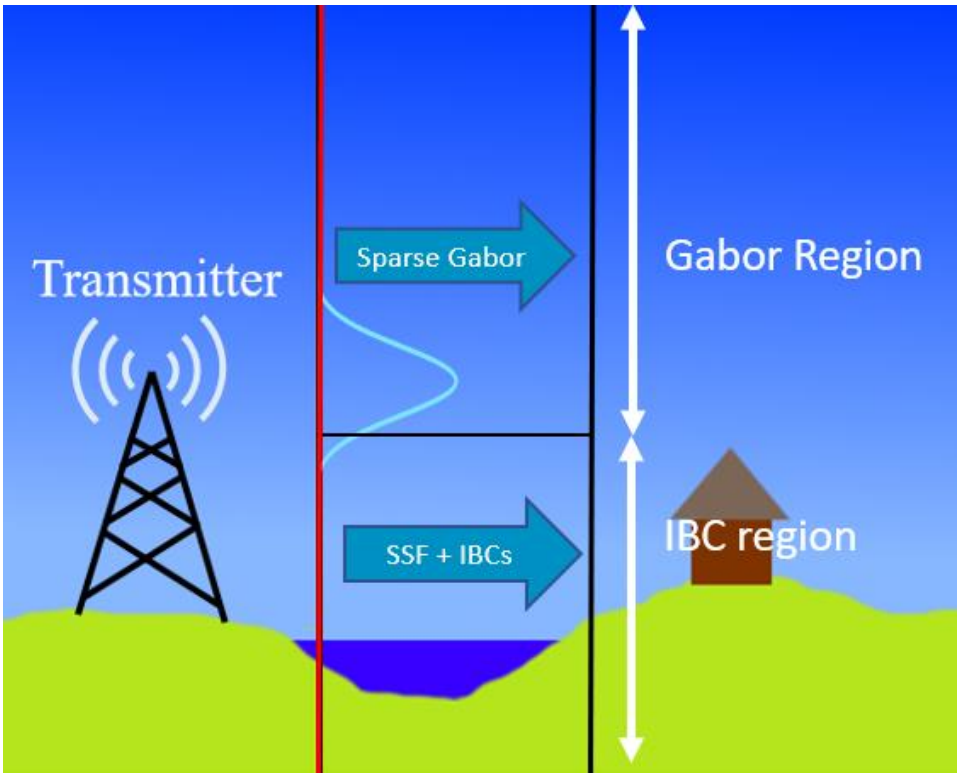


Fig. 5.1. Partitioning of the domain into the Gabor region, where fields are marched with a 3D Gabor propagator, and the IBC region, where fields are marched with 3D Split-Step Fourier augmented with Impedance Boundary Conditions.

Consider a field profile $\mathbf{\Psi}[\mathbf{z},\mathbf{y},x_0]$ with the corresponding Gabor coefficient set $\mathbf{\Phi}[\mathbf{n}_1,\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2,x_0]$ that is to be marched one step forward from $x=x_0$ to $x=x_0+\Delta x$ subject to IBCs. Consider a set of Gabor indices $\mathbf{n}_{\text{terrain}}\in\mathbf{n}_1$ about the surface of the terrain, such that the spatial support of $\mathbf{n}_{\text{terrain}}$ encloses the terrain profile over all $\mathbf{y}$ at $x=x_0$. In other words, the connected set $\mathbf{n}_{\text{terrain}}$ is chosen such that $\sum_{n_1\in\mathbf{n}_{terrain}}\mathbf{s}_{n_1}[\max_y \mathbf{h}[y,x_0]]>0$ and $\sum_{n_1\in\mathbf{n}_{terrain}}\mathbf{s}_{n_1}[\min_y \mathbf{h}[y,x_0]]>0$, where the indicator function $\mathbf{s}_{n_1}[z]$ is defined by (0.27).

The subset of the Gabor plane for $n_1\notin\mathbf{n}_{\text{terrain}}$, represented by $\mathbf{\Phi}[\mathbf{n}^c{}_{\text{terrain}},\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2,x_0]$, is marched as per (4.20) to produce $\mathbf{\Phi}_{\text{Gabor}}[\mathbf{n}^c{}_{\text{terrain}},\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2,x_0+\Delta x]$. For the section $n_1\in\mathbf{n}_{\text{terrain}}$, a local Inverse Gabor Transform is computed of the restricted function $\mathbf{\Phi}[\mathbf{n}_{\text{terrain}},\mathbf{n}_2,\mathbf{m}_1,\mathbf{m}_2,x_0]$ to produce a local spatial domain signal $\mathbf{\Psi}_{\text{IBC}}[\mathbf{z}_{\text{IBC}},\mathbf{y},x_0]$, which is to be marched forward to produce $\mathbf{\Psi}_{\text{IBC}}[\mathbf{z}_{\text{IBC}},\mathbf{y},x_0+\Delta x]$ with an IBC propagator defined in section 5.2.

The vector $\mathbf{z}_{\text{IBC}}$ is slightly bigger than the mere spatial analog of $\mathbf{n}_{\text{terrain}}$; $\mathbf{z}_{\text{IBC}}$ is made large enough that the propagated local fields $\mathbf{\Psi}_{\text{IBC}}[\mathbf{z}_{\text{IBC}},\mathbf{y},x_0+\Delta x]$ will not reach the domain boundaries. An enlarged subset $\mathbf{n}_{\text{IBC}}$ is defined such that $\mathbf{n}_{\text{terrain}}\subset\mathbf{n}_{\text{IBC}}\subset\mathbf{n}_1$, and $\mathbf{z}_{\text{IBC}}$ is

correspondingly defined to span the spatial analog of $\mathbf{n}_{\text{IBC}}$; in other words, $\mathbf{z}_{\text{IBC}} = \{z : \exists n_i \in \mathbf{n}_{\text{IBC}} : \mathbf{s}_{n_i}[z] > 0\}$. Typically, it suffices to let $\mathbf{n}_{\text{IBC}}$ be the union of $\mathbf{n}_{\text{terrain}}$ and its immediate top and bottom neighbors within the vector $\mathbf{n}$.

Once $\boldsymbol{\Psi}_{\text{IBC}}[\mathbf{z}_{\text{IBC}}, \mathbf{y}, x_0 + \Delta x]$ is obtained, a local Gabor Transform is computed over $\mathbf{z}_{\text{IBC}}$ to produce $\boldsymbol{\Phi}_{\text{IBC}}[\mathbf{n}_{\text{IBC}}, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x_0 + \Delta x]$. The total marched Gabor-domain fields are then described as the sum between the IBC fields from the IBC region, and the Gabor fields from the Gabor region, represented as

$$\begin{aligned}\boldsymbol{\Phi}[n_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x_0 + \Delta x] = &\begin{cases}\boldsymbol{\Phi}_{\text{IBC}}[n_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x_0 + \Delta x], & n_1 \in \mathbf{n}_{\text{IBC}} \\ 0, & \text{otherwise}\end{cases} \\ + &\begin{cases}\boldsymbol{\Phi}_{\text{Gabor}}[n_1, \mathbf{n}_2, \mathbf{m}_1, \mathbf{m}_2, x_0 + \Delta x], & n_1 \in \mathbf{n}_{\text{Gabor}} \\ 0, & \text{otherwise}\end{cases}\end{aligned}. \tag{5.8}$$

Equation (5.8) is used to march a Gabor-domain field through space subject to IBCs.

5.4 Numerical Results

A. Validation of Reflection Coefficient

The first numerical study is a comparison of the simulated versus theoretical reflection coefficient. Since IBCs are expressed in terms of reflection coefficients, a solver that accurately reproduces reflection coefficients can be judged to correctly implement IBCs.

A 600 MHz Gaussian beam with a 15 m waist is launched through free space towards a flat surface with permittivity $\epsilon_{\text{terrain}} = 1.1 - 0.1i$. Thirteen angles of incidence are evaluated from 3 degrees to 15 degrees from horizontal, and for each angle, two separate runs are executed in TE and TM incidence. The Gabor parameters are $N_{p_z} = N_{p_y} = 32$, and $\mathbf{n}_{\text{terrain}}$ is defined such that the support of $\sum_{n_1 \in \mathbf{n}_{terrain}} \mathbf{s}_{n_1}[z]$ spans $\max_y \mathbf{h}[y, x_0] - 2N_{p_z} \le z \le \max_y \mathbf{h}[y, x_0] + 2N_{p_z}$. The domain is discretized with $\Delta x = 35$ m, $\Delta z = \Delta y = 0.6\lambda$, and $N_z = N_y = 1024$.

A reference scenario is constructed in which there are no IBCs; the Gaussian beam is marched through free space. This reference scenario is used to evaluate the reflection coefficient; the reflection coefficient is defined as the ratio of the magnitude of the reflected beam subject to IBCs to the magnitude of the free-space beam. The reflection coefficient is compared with the theoretical Fresnel reflection coefficient in Fig. 5.2.

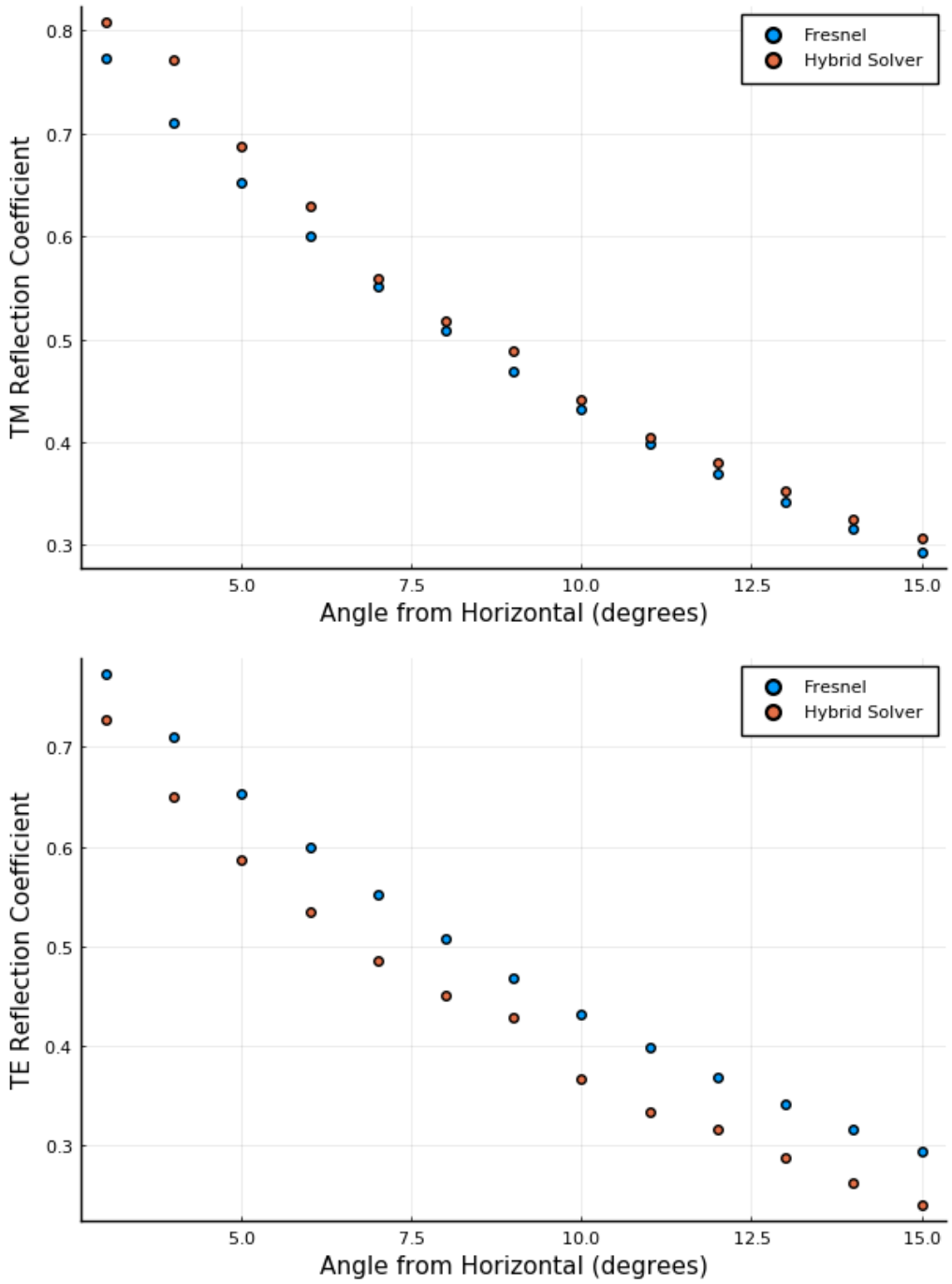


Fig. 5.2. Simulated reflection coefficient versus theoretical Fresnel reflection coefficient at different incident angles for TM incidence (top) and TE incidence (bottom)

The hybrid Gabor-IBC solver shows close agreement with the Fresnel reflection coefficient over all studied incident angles for both TE and TM incidence. It follows that the solver accurately marches fields subject to an Impedance Boundary Condition.

B. Reflection from Rough Surface

The next study to be performed is scattering from a rough surface. A 300 MHz Gaussian beam with a 10 m waist and TM polarization is launched at a periodic sinusoidal terrain with 20 m amplitude, 0.25 m period in x, and 250 m period in y. The terrain has permittivity $\epsilon_{\text{terrain}} = 3 - 4i$. The Gabor parameters are $N_{p_z} = N_{p_y} = 64$, and $\mathbf{n}_{\text{terrain}}$ is defined such that $\sum_{n_1 \in \mathbf{n}_{terrain}} \mathbf{s}_{n_1}[z] > 0$ for $\max_y \mathbf{h}[y, x_0] - 2N_{p_z} \le z \le \max_y \mathbf{h}[y, x_0] + 2N_{p_z}$. The domain is discretized with $\Delta x = 25$ m, $\Delta z = \Delta y = 0.501\lambda$, and $N_z = N_y = 1024$. Propagating fields are sparsified and are compared to a case where SSF and IBCs are used to march over the entire domain. The scattering profile is shown in Fig. 5.3, the sparsity is shown in Fig. 5.4, and the error with respect to full SSF and IBCs is shown in Fig. 5.5.

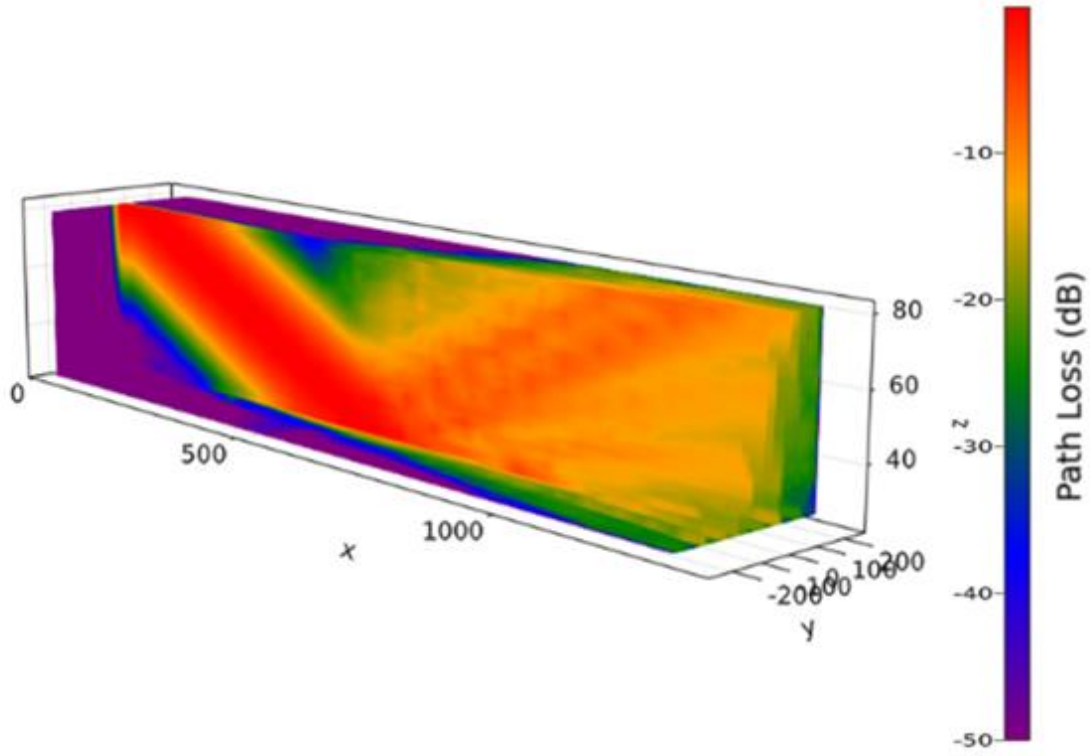


Fig. 5.3. Heatmap of field strength (dB) of a Gaussian beam scattering off of a rough surface.

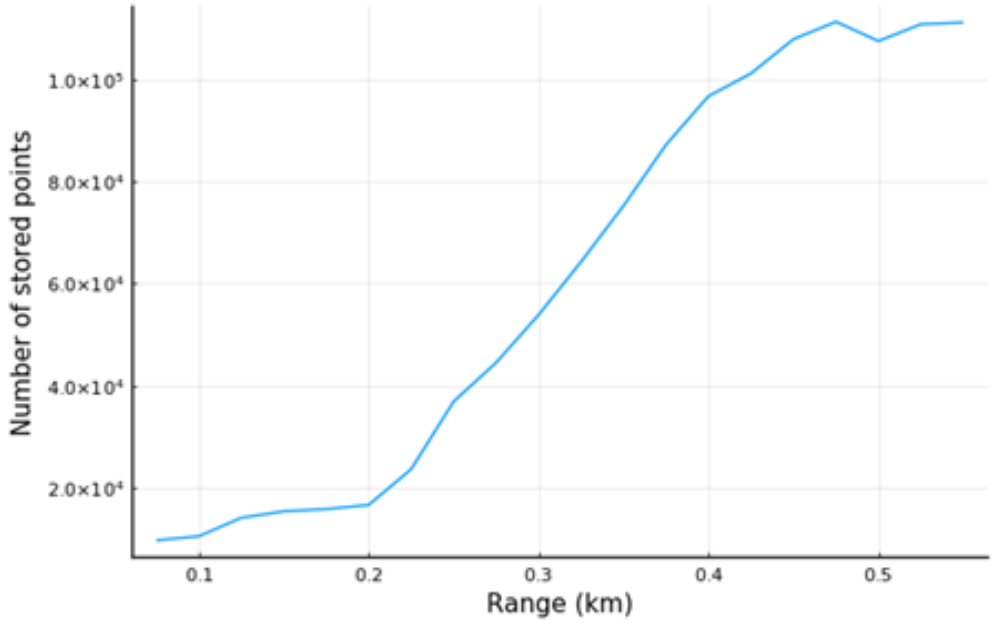


Fig. 5.4. Number of stored Gabor coefficients at each location in x in the rough surface scattering problem.

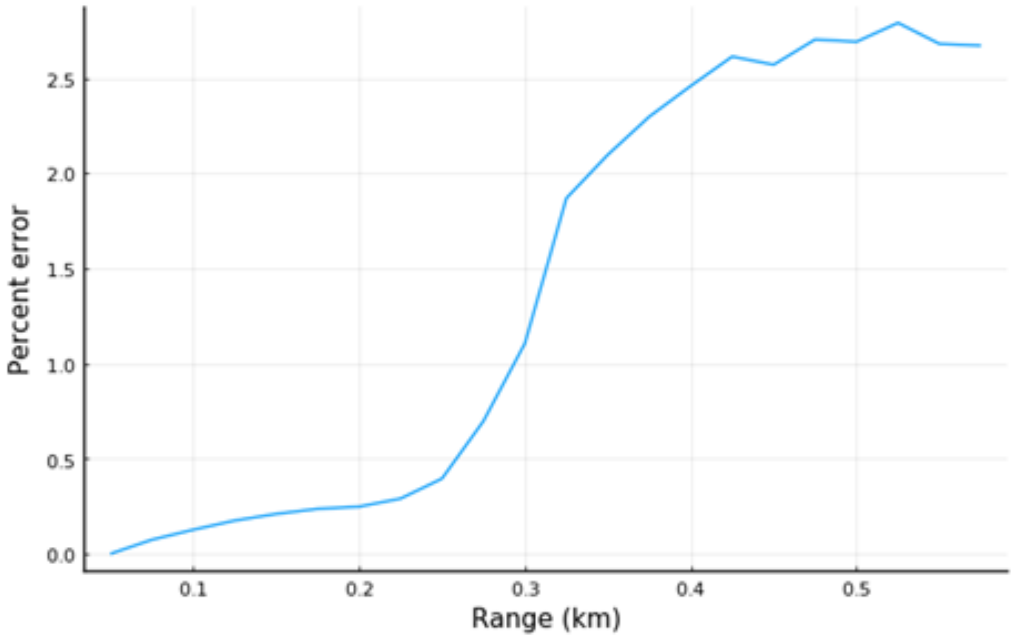


Fig. 5.5. Error of fields at each location in x, measured as the square of the difference in magnitudes between the hybrid Gabor-IBC fields, and the fields solved with a full SSF solver with IBCs.

In Fig. 5.3, fields reach the surface and then scatter off in a diffuse pattern; a peak is observed in the spectral direction and field strength tapers off for scattering angles far from the spectral direction. This behavior is consistent with scattering from rough surfaces; this indicates that the staircasing model is reasonably able to capture basic terrain features. Fig. 5.4 and Fig. 5.5 show that the Gabor-domain fields are sparsified to 80% while incurring 2.5% maximum error; when the storage of the IBC region is factored in, the problem is compressed by 30%.

The above experiment was repeated for many different compression thresholds in order to gauge the relationship between sparsity and error. The results are shown in Fig. 5.6.

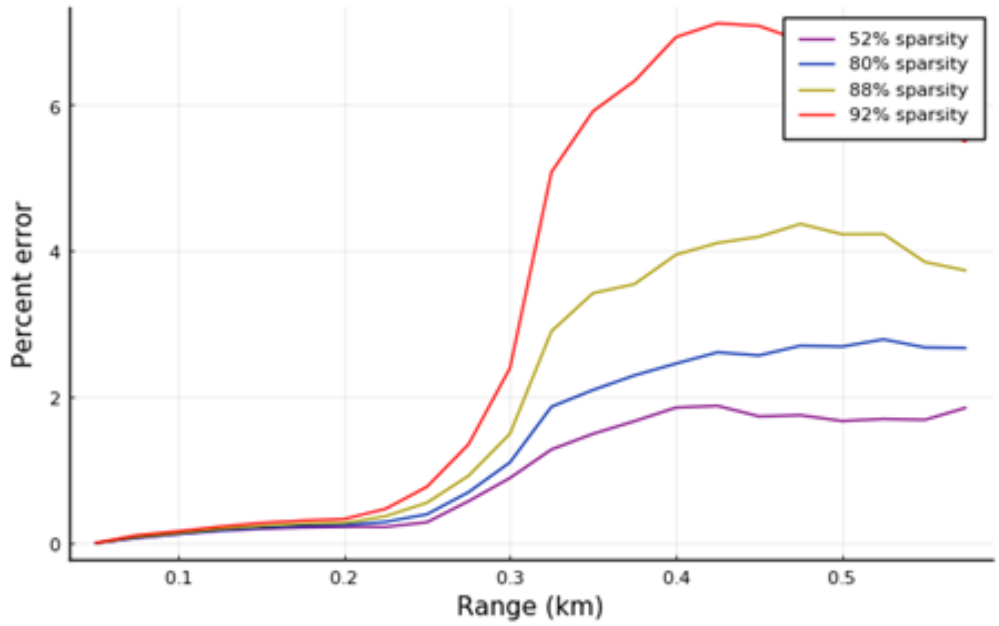


Fig. 5.6. Error of fields at each location in x for the rough surface experiment shown in fig. 5.4; each line is for a different compression level.

Fig. 5.6 shows that the error is controllable with respect to sparsity; as in previous chapters, error grows with sparsity and there are no signs of instability.

C. Propagation over Terrain

The solver is tested for a realistic scenario of long-range propagation over terrain. A 1 GHz TM-polarized Gaussian beam with a 4 m waist is launched over mountainous 3D terrain extracted from satellite data. The terrain has permittivity $\epsilon_{\text{terrain}} = 3 - 4i$, and the atmosphere is a trilinear duct as used in prior chapters. The Gabor parameters are $N_{p_z} = N_{p_y} = 512$, and $\mathbf{n}_{\text{terrain}}$ is defined such that

$\sum_{n_1 \in \mathbf{n}_{terrain}} \mathbf{s}_{n_1}[z] > 0$ for $\max_y \mathbf{h}[y, x_0] - 2N_{p_z} \le z \le \max_y \mathbf{h}[y, x_0] + 2N_{p_z}$. The domain is discretized with $\Delta x = 25$ m, $\Delta z = \Delta y = 3\lambda / 4$, and $N_z = N_y = 16384$. The results are shown in Fig. 5.7, Fig. 5.8, and Fig. 5.9.

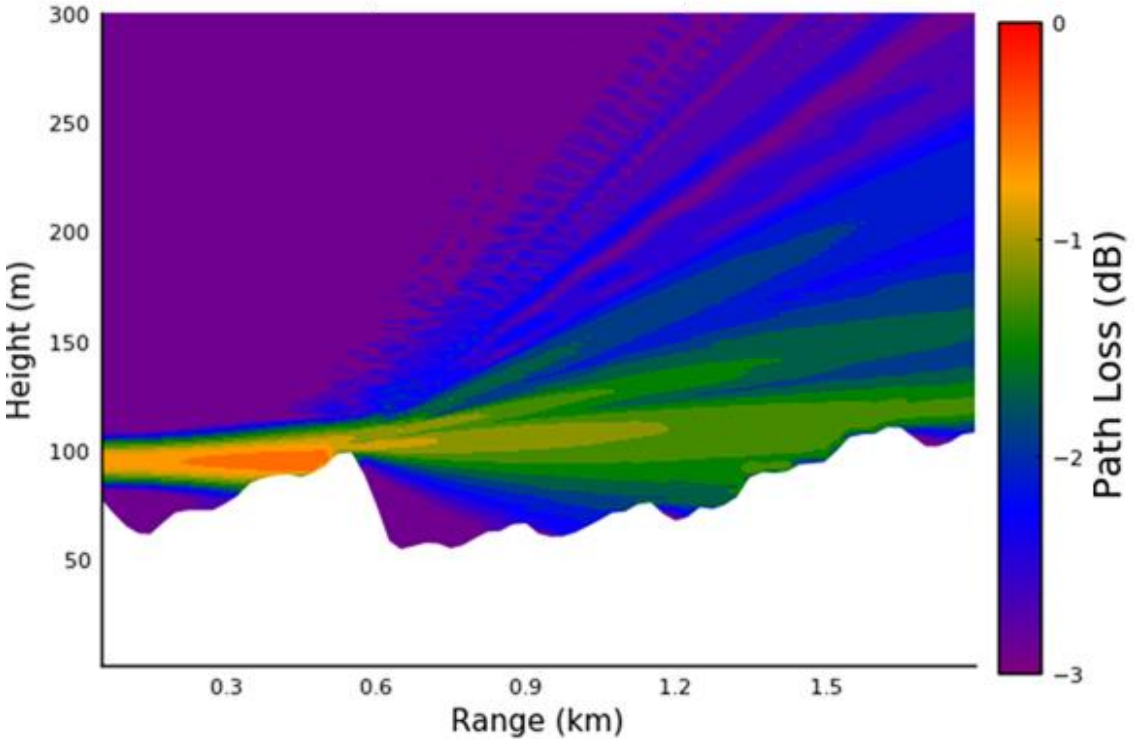


Fig. 5.7. Heatmap of field strength (dB) of propagation over 3D terrain.

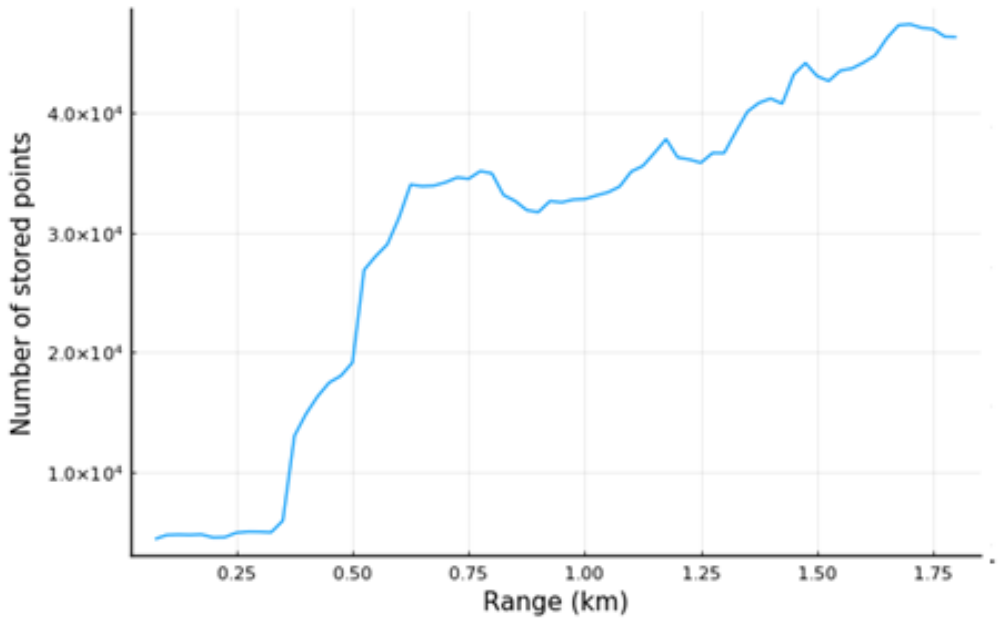


Fig. 5.8. Number of stored Gabor coefficients at each location in x in the 3D terrain problem.

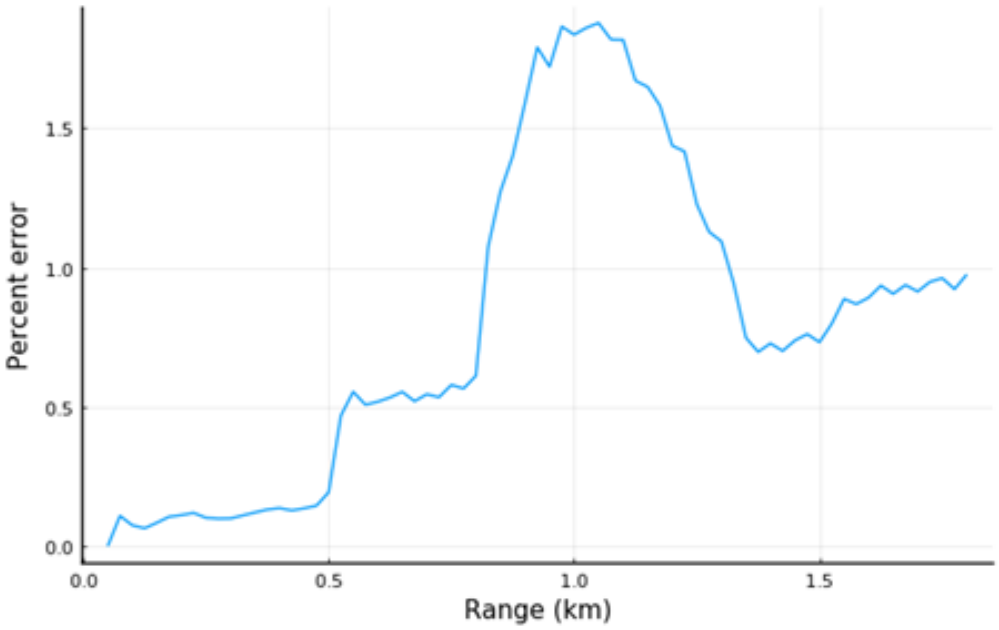


Fig. 5.9. Error of fields at each location in x, measured as the square of the difference in magnitudes between the hybrid Gabor-IBC fields, and the fields solved with a full SSF solver with IBCs.

Fig. 5.7 shows a beam marching towards a hilltop at 0.6 km and then scattering from it; scattered waves can be observed in the upper region of the domain. Past the mountain at 0.6 km, a diffraction pattern is observed, and steep shadows are seen behind each of the peaks in the terrain profile. This indicates that the IBC-staircasing model can reasonably describe terrain. Fig. 5.8 and Fig. 5.9 show that the Gabor-domain fields can be compressed by 99.98% while introducing only 2% error. When the IBC region is factored in, the total effective compression is 93%, showing a substantial reduction in stored information over the full SSF model.

Chapter 6. Conclusion

6.1 Research Contributions

This thesis presented fast methods for propagating fields over long ranges in rural environments. A Gabor-Frame-based solver was introduced that replaced an expensive Fourier-domain Split-Step Fourier propagator with a sparse Gabor-domain propagator. Representations of propagating fields were sparsified, leading to a significant improvement in performance over conventional methods; in 2D, improvements in CPU time ranged from 240% to 800%, and in 3D they ranged from 120% to 20%. In 2D, stored field information was reduced by a factor of 107 to 239, and in 3D by a factor of 10211 to 17655. The Gabor method was improved beyond existing Split-Step Wavelet literature with a Radiation Boundary Condition method and a Gabor-domain phase screen method. The Gabor Radiation Boundary Conditions eliminate excess memory consumption from previous boundary condition schemes, and absorption is extremely strong for steep angles of incidence. Phase screens were moved partially or fully into the Gabor domain by approximating the atmospheric refractive index profile as piecewise linear and computing the propagation characteristics of Gabor beamlets through each component of the piecewise linear profile. Implementation of accurate phase screens in the Gabor domain eliminates the need for full transforms to and from the spatial domain without sacrificing accuracy. Multi-window Gabor Transforms were implemented as a straightforward modification to the Gabor propagator, enabling adaptive, localized Gabor Transforms to maximally sparsify propagating fields.

The necessity for expensive rigorous solvers for accurate propagation was reduced by hybridizing propagator methods within a domain. A rural scenario is partitioned into an upper and lower region, with the upper region storing only atmosphere and the lower region encompassing the terrain and a small amount of atmosphere above the terrain. In 2D, a Split-Step Fourier solver is used to march fields in the atmosphere and a Finite Difference solver to march fields near the terrain; in 3D, a sparse Gabor propagator is used to march fields in the atmosphere and a Split-Step Fourier solver subject to Impedance Boundary Conditions is used to march fields near the terrain. By using a fast, scalable solver to efficiently march fields in the upper atmospheric region

and an expensive solver to only march fields in the small lower region, the accuracy benefits of rigorous solvers are realized without using significant computational time to solve fields in regions where Split-Step Fourier already performs optimally.

6.2 Future Work

There are many promising avenues of further exploration for Gabor frame-based solvers. The non-uniqueness of Gabor Transforms indicates that a field profile can have multiple Gabor-domain representations. Of these available representations, a scheme could be devised to arrive at the representation with optimum sparsity, either via a numerical search or through a modified transform procedure. Dictionary learning methods can also be used to optimally sparsify a Gabor representation.

The Gabor-accelerated scheme for solving propagation problems also facilitates further large-scale studies. Uncertainty Quantification methods can be used to perform a Monte-Carlo-like characterization of the sensitivity of a wireless link to different environmental parameters. Uncertainty Quantification methods, despite requiring fewer runs of a simulation than Monte-Carlo, still require a significant number of iterations in order to build an accurate surrogate model; fast Gabor frame-based propagation can reduce the time needed to perform said iterations.

Hybrid Split-Step Fourier – Finite Difference solvers can also be further improved. Forests were approximated by dielectric blocks; future models could use more accurate tree parameters to produce more realistic scattering profiles from forests and individual trees. The hybrid method is compatible with any other rigorous solution methods, including other methods for accelerating the convergence of the Finite Difference Helmholtz equation solution, as well as Finite Element or Integral Equation techniques.

Bibliography

[1] M. Levy, *Parabolic Equation Methods for Electromagnetic Wave Propagation*, 1st ed. London, UK: Institute of Engineering Technology, 2000.

[2] D. J. Thomson and N. R. Chapman, "A wide- angle split- step algorithm for the parabolic equation," *J. Acoust. Soc. Am.*, 1983, doi: 10.1121/1.390272.

[3] K. H. Craig and M. F. Levy, "Parabolic equation modelling of the effects of multipath and ducting on radar systems," *IEE Proc. F Radar Signal Process.*, 2010, doi: 10.1049/ip-f-2.1991.0021.

[4] A. E. Barrios, "A Terrain Parabolic Equation Model for Propagation in the Troposphere," *IEEE Trans. Antennas Propag.*, 1994, doi: 10.1109/8.272306.

[5] R. Janaswamy, "Path loss predictions in the presence of buildings on flat terrain: A 3-D vector parabolic equation approach," *IEEE Trans. Antennas Propag.*, 2003, doi: 10.1109/TAP.2003.815415.

[6] B. Engquist and A. Majda, "Absorbing boundary conditions for the numerical simulation of waves," *Math. Comput.*, 1977, doi: 10.2307/2005997.

[7] M. D. Collins, "A split- step Padé solution for the parabolic equation method," *J. Acoust. Soc. Am.*, 2005, doi: 10.1121/1.406739.

[8] M. D. Collins, R. J. Cederberg, D. B. King, and S. A. Chin- Bing, "Comparison of algorithms for solving parabolic wave equations," *J. Acoust. Soc. Am.*, 2005, doi: 10.1121/1.415921.

[9] E. M. Salomons, "Improved Green's function parabolic equation method for atmospheric sound propagation," *J. Acoust. Soc. Am.*, 2002, doi: 10.1121/1.423260.

[10] J. P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, 1994, doi: 10.1006/jcph.1994.1159.

[11] F. Collino, "Perfectly matched absorbing layers for the paraxial equations," *J. Comput. Phys.*, vol. 131, pp. 164–180, 1997.

[12] M. F. Levy, "Transparent boundary conditions for parabolic equation solutions of radiowave propagation problems," *IEEE Trans. Antennas Propag.*, 1997, doi: 10.1109/8.554242.

[13] M. F. Levy, "Perfectly matched layer truncation for parabolic wave equation models," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, 2001, doi: 10.1098/rspa.2001.0848.

[14] A. H. Heidari and M. N. Guddati, "Highly accurate absorbing boundary conditions for

wide-angle wave equations," *GEOPHYSICS*, 2006, doi: 10.1190/1.2192914.

[15] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans. Inf. Theory*, 1990, doi: 10.1109/18.57199.

[16] L. Chen, R. Wu, and Y. Chen, "Target-oriented beamlet migration based on Gabor-Daubechies frame decomposition," *GEOPHYSICS*, 2006, doi: 10.1190/1.2187781.

[17] T. Bonnafont, R. Douvenot, and A. Chabory, "Split-step wavelet propagation modelling using local operators," 2019, doi: 10.23919/URSI-EMTS.2019.8931470.

[18] H. Zhou, R. Douvenot, and A. Chabory, "Modeling the long-range wave propagation by a split-step wavelet method," *J. Comput. Phys.*, 2020, doi: 10.1016/j.jcp.2019.109042.

[19] H. Zhou, A. Chabory, and R. Douvenot, "A Fast Wavelet-To-Wavelet Propagation Method for the Simulation of Long-Range Propagation in Low Troposphere," *IEEE Trans. Antennas Propag.*, vol. 70, no. 3, pp. 2137–2148, Mar. 2022, doi: 10.1109/TAP.2021.3118821.

[20] P. Cruz, A. Mendes, and F. D. Magalhães, "Wavelet-based adaptive grid method for the resolution of nonlinear PDEs," *AIChE J.*, vol. 48, no. 4, 2002, doi: 10.1002/aic.690480412.

[21] J. C. Santos, P. Cruz, M. A. Alves, P. J. Oliveira, F. D. Magalhaes, and A. Mendes, "Adaptive multiresolution approach for two-dimensional PDEs," *Comput. Methods Appl. Mech. Eng.*, vol. 193, no. 3–5, 2004, doi: 10.1016/j.cma.2003.10.005.

[22] H. Yousefi, A. Noorzad, and J. Farjoodi, "Wavelet based adaptive solution of wave equation: Multiresolution simulation of wave propagation," 2011, doi: 10.1109/ICMT.2011.6002350.

[23] A. Harten, "Multiresolution algorithms for the numerical solution of hyperbolic conservation laws," *Commun. Pure Appl. Math.*, vol. 48, no. 12, 1995, doi: 10.1002/cpa.3160481201.

[24] A. Cohen, S. M. Kaber, S. Müller, and M. Postel, "Fully adaptive multiresolution finite volume schemes for conservation laws," *Math. Comput.*, vol. 72, no. 241, 2001, doi: 10.1090/s0025-5718-01-01391-6.

[25] R. Behera and M. Mehra, "A Dynamic Adaptive Wavelet Method for Solution of the Schrodinger Equation," *http://dx.doi.org/10.1142/S1756973714500012*, vol. 06, no. 01, p. 1450001, Jul. 2015, doi: 10.1142/S1756973714500012.

[26] M. Zibulski and Y. Y. Zeevi, "Analysis of Multiwindow Gabor-Type Schemes by Frame Methods," *Appl. Comput. Harmon. Anal.*, 1997, doi: 10.1006/acha.1997.0209.

[27] T. Bonnafont, R. Douvenot, and A. Chabory, "A Local Split-Step Wavelet Method for the Long Range Propagation Simulation in 2D," *Radio Sci.*, 2021, doi: 10.1029/2020RS007114.

[28] P. L. Longley, A.G. Rice, "PREDICTION 3F TROPOSPHERIC RADiv TRANSMISSION LOSS OVER IRREGULAR TERRAIN- A COMPUTER METHOD -

i1968," *ESSA Tech. Rep.*, 1968.

[29] F. Ikegami, S. Yoshida, T. Takeuchi, and M. Umehira, "Propagation Factors Controlling Mean Field Strength on Urban Streets," *IEEE Trans. Antennas Propag.*, 1984, doi: 10.1109/TAP.1984.1143419.

[30] J. P. Rossi and Y. Gabillet, "A mixed ray launching/tracing method for full 3-D UHF propagation modeling and comparison with wide-band measurements," *IEEE Trans. Antennas Propag.*, 2002, doi: 10.1109/TAP.2002.1003388.

[31] M. Lebherz, W. Wiesbeck, and W. Krank, "A Versatile Wave Propagation Model for the VHF/UHF Range Considering Three-Dimensional Terrain," *IEEE Trans. Antennas Propag.*, 1992, doi: 10.1109/8.182444.

[32] R. J. Luebbers, "Propagation Prediction for Hilly Terrain Using GTD Wedge Diffraction," *IEEE Trans. Antennas Propag.*, 1984, doi: 10.1109/TAP.1984.1143449.

[33] D. Huang, "Finite Element Solution To The Parabolic Wave Equation," *J. Acoust. Soc. Am.*, 1988, doi: 10.1121/1.396587.

[34] B. W. Dowd and R. E. Diaz, "FDTD Simulation of Very Large Domains Applied to Radar Propagation Over the Ocean," *IEEE Trans. Antennas Propag.*, 2018, doi: 10.1109/TAP.2018.2852141.

[35] X. Xi, L. Zhou, J. Zhang, J. Liu, and L. Wang, "Combined IE-FDTD algorithm for long-range Loran-C ground-wave propagation," *IEEE Trans. Antennas Propag.*, 2012, doi: 10.1109/TAP.2012.2201102.

[36] R. J. Mcarthur and D. H. O. Bebbington, "Diffraction over simple terrian obstacles by the method of parabolic equations," 1991.

[37] G. Daniel Dockery and J. R. Kuttler, "An improved impedance-boundary algorithm for fourier split-step solutions of the parabolic wave equation," *IEEE Trans. Antennas Propag.*, 1996, doi: 10.1109/8.546245.

[38] R. Janaswamy, "A curvilinear coordinate-based split-step parabolic equation method for propagation predictions over terrain," *IEEE Trans. Antennas Propag.*, 1998.

[39] R. J. McArthur, "Propagation modelling over irregular terrain using the split-step parabolic equation method," 1992.

[40] G. P. dos Santos Cavalcante, D. A. Rogers, and A. J. Giarola, "Radio loss in forests using a model with four layered media," *Radio Sci.*, 1983, doi: 10.1029/RS018i005p00691.

[41] L. W. Li, T. S. Yeo, P. S. Kooi, and M. S. Leong, "Radio wave propagation along mixed paths through a four-layered model of rain forest: an analytic approach," *IEEE Trans. Antennas Propag.*, 1998, doi: 10.1109/8.704814.

[42] B. Jean-Pierre, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, 1994.

[43] D. Lee and S. T. McDaniel, "OCEAN ACOUSTIC PROPAGATION BY FINITE DIFFERENCE METHODS.," *Comput. Math. with Appl.*, 1987.

[44] M. F. Levy, “Parabolic Equation Modelling of Propagation Over Irregular Terrain,” *Electron. Lett.*, 1990, doi: 10.1049/el:19900746.

[45] J. Feng, L. Zhou, X. Xu, and C. Liao, “A Hybrid TDPE/FDTD Method for Site-Specific Modeling of O2I Radio Wave Propagation,” *IEEE Antennas Wirel. Propag. Lett.*, 2018, doi: 10.1109/LAWP.2018.2861471.

[46] Z. El Ahdab and F. Akleman, “An Efficient 3-D FDTD-PE Hybrid Model for Radio Wave Propagation with Near-Source Obstacles,” *IEEE Trans. Antennas Propag.*, vol. 67, no. 1, 2019, doi: 10.1109/TAP.2018.2876719.

[47] G. K. Theofilogiannakos, T. D. Xenos, and T. V. Yioultsis, “A hybrid parabolic equation-integral equation technique for wave propagation modeling of indoor communications,” *IEEE Trans. Magn.*, 2009, doi: 10.1109/TMAG.2009.2012642.

[48] W. Hou, J. Wang, and Y. Li, “A hybrid method of FDTD and vector parabolic equation for radio wave propagation prediction in tunnels,” 2017, doi: 10.1109/APUSNCURSINRSM.2017.8072862.

[49] S. Reynaud, C. Guiffaut, A. Reineix, and R. Vauzelle, “Modeling indoor propagation using an indirect hybrid method combining the UTD and the FDTD methods,” 2004.

[50] R. N. Baer, “Propagation through a three- dimensional eddy including effects on an array,” *J. Acoust. Soc. Am.*, vol. 69, no. 1, p. 70, Aug. 1998, doi: 10.1121/1.385253.

[51] W. L. Siegmann, G. A. Kriegsmann, and D. Lee, “A wide- angle three- dimensional parabolic wave equation,” *J. Acoust. Soc. Am.*, vol. 78, no. 2, p. 659, Aug. 1998, doi: 10.1121/1.392434.

[52] A. Tolstoy, “3-D PROPAGATION ISSUES AND MODELS,” *https://doi.org/10.1142/S0218396X96000076*, vol. 4, no. 3, pp. 243–271, Nov. 2011, doi: 10.1142/S0218396X96000076.

[53] C. A. Zelley and C. C. Constantinou, “A three-dimensional parabolic equation applied to vhf/uhf propagation over irregular terrain,” *IEEE Trans. Antennas Propag.*, vol. 47, no. 10, pp. 1586–1596, 1999, doi: 10.1109/8.805904.

[54] H. F. Rasool, X. Pan, and X. Sheng, “A Fourier Split-Step Based Wide-Angle Three-Dimensional Vector Parabolic Wave Equation Algorithm Predicting the Field Strength Over Flat and Irregular Forest Environments,” 2019.

[55] A. A. Zaporozhets and M. F. Levy, “Bistatic RCS calculations with the vector parabolic equation method,” *IEEE Trans. Antennas Propag.*, vol. 47, no. 11, pp. 1688–1696, Nov. 1999, doi: 10.1109/8.814948.

[56] J. S. Perkins and R. N. Baer, “An approximation to the three- dimensional parabolic-equation method for acoustic propagation,” *J. Acoust. Soc. Am.*, vol. 72, no. 2, p. 515, Aug. 1998, doi: 10.1121/1.388032.

[57] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, “Computational Ocean Acoustics,” *Comput. Ocean Acoust.*, 2011, doi: 10.1007/978-1-4419-8678-8.

[58] H. Zhou, A. Chabory, and R. Douvenot, “A 3-D Split-Step Fourier Algorithm Based on a Discrete Spectral Representation of the Propagation Equation,” *IEEE Trans. Antennas Propag.*, vol. 65, no. 4, pp. 1988–1995, Apr. 2017, doi: 10.1109/TAP.2017.2671018.

[59] Y. Shen and C. Cao, “Parallel method of parabolic equation for electromagnetic environment simulation,” 2016, doi: 10.1109/ITNEC.2016.7560414.

[60] O. Ozgun, G. Apaydin, M. Kuzuoglu, and L. Sevgi, “PETOOL: MATLAB-based one-way and two-way split-step parabolic equation tool for radiowave propagation over variable terrain,” *Comput. Phys. Commun.*, vol. 182, no. 12, 2011, doi: 10.1016/j.cpc.2011.07.017.

[61] H. F. Rasool, X. M. Pan, and X. Q. Sheng, “Radiowave Propagation prediction in the Presence of Multiple Knife Edges using 3D Parabolic Equation Method,” 2019, doi: 10.23919/ACESS.2018.8669168.

[62] Z. El Ahdab and F. Akleman, “Radiowave Propagation Analysis with a Bidirectional 3-D Vector Parabolic Equation Method,” *IEEE Trans. Antennas Propag.*, vol. 65, no. 4, 2017, doi: 10.1109/TAP.2017.2670321.

[63] F. Jaillet and B. Torrésani, “Time-frequency jigsaw puzzle: Adaptive multiwindow and multilayered Gabor expansions,” *Int. J. Wavelets, Multiresolution Inf. Process.*, 2007, doi: 10.1142/S0219691307001768.

[64] T. Bonnafont, R. Douvenot, and A. Chabory, “Split-step wavelet with local operators for the 3D long-range propagation,” *15th Eur. Conf. Antennas Propagation, EuCAP 2021*, Mar. 2021, doi: 10.23919/EUCAP51087.2021.9411173.

[65] T. S. Lee, “Image representation using 2d gabor wavelets,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, 1996, doi: 10.1109/34.541406.

[66] J. M. Jin, *Theory and Computation of Electromagnetic Fields*. 2010.

[67] S. C. H. A. Taflove, “Computational electrodynamics: the finite-difference time-method.,” *Sencond Ed. Artech house Publ. Norwood.*, 2000.

[68] B. Engquist and L. Ying, “Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers,” *Multiscale Model. Simul.*, 2011, doi: 10.1137/100804644.

[69] O. Ozgun, “Recursive two-way parabolic equation approach for modeling terrain effects in tropospheric propagation,” *IEEE Trans. Antennas Propag.*, 2009, doi: 10.1109/TAP.2009.2027166.

[70] J. A. Azevedo and F. E. Santos, “A model to estimate the path loss in areas with foliage of trees,” *AEU - Int. J. Electron. Commun.*, 2017, doi: 10.1016/j.aeue.2016.10.018.