Split-step framelet: leveraging wavelet frames for tropospheric long-range propagation

Thomas Bonnafont
Lab-STICC, UMR CNRS 6285, ENSTA, Institut Polytechnique de Paris
Brest, France
thomas.bonnafont@ensta-bretagne.fr

***Abstract*—Modeling electromagnetic wave propagation over long distances is crucial for radar systems and navigation applications. This work introduces a split-step method based on wavelet frames to model tropospheric long-range propagation. This approach capitalizes on a fully wavelet-to-wavelet framework, designed to enhance computational efficiency, accuracy, and ease of implementation on GPUs. Numerical experiments in the VHF frequency band demonstrate the effectiveness and advantages of the proposed method.**

***Index Terms*—long-range propagation, parabolic wave equation, split-step, wavelet, frames**

I. Introduction

Efficient computational methods for modeling tropospheric long-range propagation are essential for various applications, including optimized antenna placement and radar coverage prediction. In this context, the parabolic wave equation (PWE) asymptotic model [1] has become a widely adopted approach. Indeed, by considering only the forward propagation, the latter enables using a coarser grid along the propagation direction, drastically reducing the computation time. Moreover, with this model, we can consider the relief, the refraction, and the ground composition.

In particular, the split-step Fourier [1], [2] (SSF) method has been widely used to solve the PWE efficiently. The SSF method iteratively computes the field by alternating between the spectral and spatial domains. Its efficiency arises from the diagonal structure of the scattering operator in the spectral domain. However, for large-scale 2D or 3D domains [3], the SSF method becomes impractical due to high memory requirements and computational complexity.

To address these limitations, wavelet-based split-step methods have been recently proposed [4]–[6]. These methods reduce computation time and memory usage while maintaining a good level of accuracy, leveraging the high compression rates achievable in the wavelet domain [7]. Most of these approaches rely on wavelet bases. Additionally, a full wavelet-to-wavelet split-step method has been introduced in [8] to improve computational efficiency further. However, wavelet bases have the drawback of losing the translation-invariance property, which complicates the scattering operator. Furthermore, certain operations must still be performed in the spatial domain to ensure adequate accuracy.

To overcome some of the aforementioned limitations. We propose here a fully wavelet-to-wavelet split-step method, called split-step framelet (SSFW), using wavelet frames [7] instead of wavelet bases to preserve the translation-invariance property, thereby simplifying the scattering operator and enhancing computational efficiency. In addition, the wavelet frame-based operations can more easily be implemented on GPU.

II. The propagation model

A. Notations, assumptions, and discretization

In this work, we adopt a Cartesian coordinate system (x, z), where x represents the propagation direction and z denotes the altitude. The refractive index is denoted by $n(x, z)$, and we assume it varies slowly along the x-direction. The field is modeled with a time dependence of $\exp(j\omega t)$, where j is the imaginary unit and ω is the angular frequency.

We consider a source located at $x_s \leq 0$, with the field specified at $x = 0$. We aim to compute the field over the ground to a maximum range $x_{\max}$. The computational domain is thus defined as $\Omega = [0, x_{\max}] \times [0, z_{\max}]$.

For numerical purposes, the domain is discretized along both axes, with N_x and N_z denoting the number of points in the x- and z-directions, respectively. The resulting mesh sizes are $\Delta x = x_{\max}/N_x$ and $\Delta z = z_{\max}/N_z$. A discrete representation of any function $u(x, z)$ is expressed as $u[n_x, n_z]$, where $n_x \in \{0, \ldots, N_x - 1\}$ and $n_z \in \{0, \ldots, N_z - 1\}$. Additionally, $u_x[n_z]$ refers to the semi-discretized version of u along z.

Finally, here we work in a transverse electric configuration, but it can easily be generalized to the transverse magnetic case.

B. The parabolic wave equation (PWE) model

To compute forward long-range propagation up to the maximum range $x_{\max}$, we employ the widely used parabolic wave equation (PWE) [2], [3], [5], [8]. Denoting the reduced field by $u(x, z)$, the PWE is expressed as [1]:

$$\frac{\partial u}{\partial x} = -j \left(\sqrt{\frac{\partial^2}{\partial z^2} + k^2} - k \right) u - jk\,(n-1)\,u, \qquad (1)$$

where k is the wavenumber. The PWE is an asymptotic form of the Helmholtz equation that accounts only for forward propagation within a paraxial cone of 45° along the x-axis [1]. Since eq. (1) is an ordinary differential equation along x, it supports larger discretization steps in this direction, making it highly efficient for our application. Additionally, the PWE allows us to account for the effects of the refraction, the terrain relief, and the ground composition [1].

III. Split-step framelet (SSFW)

In this section, we present an efficient wavelet frame-based method to solve eq. (1), leveraging wavelet frames. The latter is called split-step framelet and is denoted from here on as SSFW.

A. A brief introduction on wavelet frames

Frames can be seen as a generalization of bases, where the family can be redundant. More formally, denote $(\chi_n)_n$ as a countable family, and u as the reduced field. The family $(\chi_n)_n$ is a frame if

$$A\|u\|^2 \leq \sum_n |\langle u, \chi_n\rangle|^2 \leq B\|u\|^2,$$

where $0 < A < B$, and $\langle\cdot,\cdot\rangle$ is the corresponding scalar product. If $A = B$, then the frame is said to be tight. Furthermore, we have an energy conservation property. Therefore, we only use wavelet frames with $A = B = 1$ in the following.

One question that can arise is *why use wavelet frames instead of a wavelet basis?* The main problem with the wavelet basis is that translation invariance is not preserved, since subsampling is needed in the fast wavelet transform [7] (FWT). Using wavelet frames allows us to keep the translation invariance property and manipulate vectors of the same size. The latter two properties are important for accurate propagation computation. Also, note that the first level $n = 0$ corresponds here to the scaling function [7].

Furthermore, wavelet frames are better for GPU implementation even if the FWT is computationally efficient. Indeed, computing $\langle u, \chi_n\rangle$ amounts to convolution with wavelets that have compact support and can thus be efficiently computed in the spatial domain. However, we can also opt for a Fourier transform-based convolution in this case since the fast Fourier transformed library is highly efficient on GPU.

In particular, in the following, since in 2D, it has been shown that the wavelet family has almost no impact on the accuracy [4], we use the Haar family. This allows us to have a symmetric scaling function and an antisymmetric wavelet function, which is useful to efficiently handle the ground, as will be seen later on. Nonetheless, most of the parts can be generalized to other families.

To conclude this part, we denote the wavelet frame transform of any function, such as u, by an upper case, e.g. U. In this case $U[l,p]$ corresponds to the coefficient at level $l \in [0, L]$ and position $p \in [0, N_z - 1]$.

B. Description of SSFW

The SSFW algorithm builds upon the wavelet-to-wavelet split-step wavelet method (wSSW) introduced in [8]. It is an iterative algorithm for computing the field u entirely within the wavelet domain, differing from the original split-step wavelet (SSW) method [4], [5], as no spatial operators are required.

First, a redundant wavelet transform, $\mathbf{S}_w$, over L levels is applied to the initial field u_0 of size N_z, resulting in U_0 of size $L \times N_z$. To reduce computational effort, a hard threshold compression operator, $\mathbf{C}_{V_s}$, is applied:

$$U_0 = \mathbf{C}_{V_s}\mathbf{S}_w u_0, \tag{2}$$

where V_s is the threshold chosen to ensure high accuracy while significantly reducing the number of elements in U_0. Unlike wSSW [8], where a wavelet basis is used, SSFW relies on a frame, making the vector L times larger. Nevertheless, the number of nonzero coefficients, $N_s \ll N_z$, remains low.

Then, propagation from x to $x + \Delta x$ is carried out in three main steps:

1) **Wavelet-domain propagation:** Using a precomputed, compressed local scattering operator $\mathbf{P}_{V_p}$, propagation is performed as:

$$U^{\mathrm{fs}}_{x+\Delta x} = \mathbf{P}_{V_p} U_x. \tag{3}$$

This operator is constructed as in [5] by computing the redundant wavelet transform of a propagated wavelet of each level. This propagation is calculated using the SSF scheme. Since a frame is used, the translation invariance property ensures that only one propagator per level is needed, reducing the scattering operator's size. The propagation step is described in more detail in the next section.

2) **Refraction and apodization:** The phase screen $\mathbf{R}$ and the apodization $\mathbf{A}$ (Hamming window) operators are directly applied in the wavelet domain. Unlike [8], no additional transformation is required, as $\mathbf{R}$ operates directly on all L levels:

$$U^{\mathrm{R}}_{x+\Delta x} = \mathbf{A}\mathbf{R}U^{\mathrm{fs}}_{x+\Delta x}. \tag{4}$$

3) **Compression:** A hard threshold compression is applied again to retain only significant coefficients:

$$U_{x+\Delta x} = \mathbf{C}_{V_s} U^{\mathrm{R}}_{x+\Delta x}. \tag{5}$$

These steps are repeated iteratively until $x_{\max}$, at which point an inverse redundant wavelet transform is applied to recover the field. While the approach of SSW [4], [5] involves alternating between wavelet and spatial domains, the SSFW method avoids this, by applying the wavelet transform only at the start and end. This significantly reduces computational complexity, and memory usage, and enhances efficiency, particularly for GPU implementation.

C. Propagating in the wavelet domain

As detailed in the previous section, the scattering operator $\mathbf{P}_{V_p}$ contains the wavelet transform of L propagated wavelets. This operator is structured as a tensor of size $L \times L \times N_z$, where L represents the number of wavelets to propagate, and $L \times N_z$ corresponds to the coefficients generated by the redundant wavelet transform.

The propagation step in the wavelet domain is computed as

$$U_{x+\Delta x}[l'] = \sum_l U_x[l] \otimes P[l][l'], \tag{6}$$

where $\otimes$ denotes the convolution operation. Unlike SSW, the use of frames with translation invariance enables direct computation of the coefficients at each level via convolution. This significantly simplifies the GPU implementation of the method by avoiding additional transformations. In terms of computational complexity, the use of frames increases the overall cost of SSW by a factor of L, which remains relatively low in most cases. This makes the method both efficient and well-suited for practical applications.

D. Accounting for the ground

Assuming the ground is a perfect electric conductor, we provide a local image method for frames that allows accounting for the ground directly in the wavelet domain at almost no cost. In a few words, the local image method [5] accounts for the ground by adding a thin image layer in the spatial domain. In this layer, reflection is accounted for by adding an antisymmetric replica. Nonetheless, since a wavelet basis was used in [4], [5], [8], one could not directly incorporate it in the wavelet domain even in [8], where the propagator is modified near the ground if needed. Here, with frames, we propose a local image method in the wavelet domain.

Proposition 1 (A wavelet local image method)**.** *Assuming we use the Haar wavelet family and $L = 2$. For all levels 0 to 2, we add $N_{\mathrm{im}} \ll N_z$ points at the beginning of each vector of size N_z, leading to U_x^{im}, and we set*

$$\forall (l,p) \in [0,L] \times [0, N_z - 1],\ U_x^{\mathrm{im}}[l, N_{\mathrm{im}} + p] = U_x[l,p].$$

Due to the antisymmetry of the Haar wavelet, for level L then

$$\forall p \in [0, N_{\mathrm{im}} - 1],\ U_x^{\mathrm{im}}[2,p] = U_x[2, N_{\mathrm{im}} - 1 - p],$$

corresponding to a symmetric replication. For the level $l = 1$, this leads to

$$\begin{aligned} \forall p \in [0, N_{\mathrm{im}} - 3],\ U_x^{\mathrm{im}}[1,p] &= U_x[1, N_{\mathrm{im}} - 3 - p], \\ U_x^{\mathrm{im}}[1, N_{\mathrm{im}} - 2] &= \tfrac{3U_x[2,0] + U_x[2,1]}{2}, \\ U_x^{\mathrm{im}}[1, N_{\mathrm{im}} - 1] &= \tfrac{3U_x[2,0] + U_x[2,1]}{2}. \end{aligned}$$

Finally, for the scaling function, i.e. the level $l = 0$, which is symmetric, this leads to

$$\begin{aligned} \forall p \in [0, N_{\mathrm{im}} - 3],\ U_x^{\mathrm{im}}[0,p] &= -U_x[0, N_{\mathrm{im}} - 3 - p], \\ U_x^{\mathrm{im}}[0, N_{\mathrm{im}} - 2] &= \tfrac{U_x[2,0] + U_x[2,1]}{2}, \\ U_x^{\mathrm{im}}[0, N_{\mathrm{im}} - 1] &= -\tfrac{U_x[2,0] + U_x[2,1]}{2}. \end{aligned}$$

Proof. The proof of the proposition follows directly from calculating the wavelet coefficients U_x and U_x^{im}. For this, we use the filter coefficients of the Haar redundant wavelet transform, given by:

$$h_0 = \left[\tfrac{1}{2}, \tfrac{1}{2}\right],\ g_0 = \left[-\tfrac{1}{2}, \tfrac{1}{2}\right],\ h_1 = \left[\tfrac{1}{2}, 0, \tfrac{1}{2}\right],\ g_1 = \left[-\tfrac{1}{2}, 0, \tfrac{1}{2}\right].$$

Let $u = [0, u_1, u_2, \dots, u_{N_z - 1}]$ represent the vector of the reduced field, and

$$u^{\mathrm{im}} = [-u_{N_{\mathrm{im}}}, \dots, -u_2, -u_1, 0, u_1, u_2, \dots, u_{N_z - 1}]$$

denote its local image replica in the spatial domain. The computation of the wavelet coefficients is then achieved by cascading convolutions of u or u^{im} with these filters. Finally, we obtain the proposed results by comparing the expression of U_x^{im} and U_x. □

The proposition and proof have been presented for the Haar wavelet but can be extended to other wavelet families with symmetry and antisymmetry properties. Moreover, the method is generalizable to any maximum level L. Additionally, a dielectric ground can be incorporated by using the Fresnel coefficient.

This implementation significantly reduces the complexity compared to [4], [5], [8], as all computations are performed entirely in the wavelet domain. Furthermore, it facilitates a more efficient GPU implementation, enhancing computational performance. Since, the GPU implementation is not the core of the paper, here we use the functions provided by the PyTorch package of Python.

IV. Numerical tests

In this section, numerical tests are performed to validate the method by comparing it with SSF and highlighting its advantages over SSW.

A. Free-space propagation

To validate the method, we consider a simple scenario with a domain size of 5 km and 2048 m in the x and z directions, respectively. The source is a complex source point (CSP) with a width of $w_0 = 5$ m, placed at $x_s = -50$ m and $z_s = 1024$ m. Setting $n = 1$ allows us to simulate free-space propagation. In this case, we expect SSFW to produce results comparable to SSW.

For thresholds, we use $V_s = 10^{-3} \|u_0\|_\infty$ and $V_p = 10^{-5} \|P\|_\infty$, leading to an expected error below -30 dB when compared to SSF [9]. The method is tested for different maximum levels of decomposition, specifically $L \in \{1, 2\}$. Figure 1 shows the reduced field at $x_{\max}$ for SSF (solid line) and SSW, SSFW, and SSFW/GPU (dashed lines) when $L = 1$.

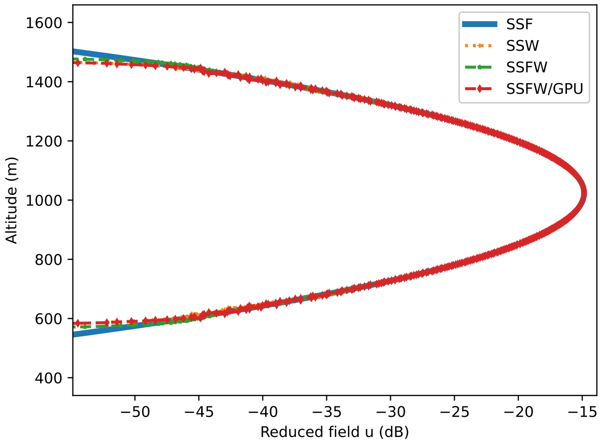

Fig. 1. Reduced field u in dB at $x_{\max}$ computed with different methods.

In addition, Table I provides the computation times needed for the different methods and levels L. We also show the different maximum differences with SSF.

TABLE I
COMPARISON OF SSW, AND SSFW ON CPU OR GPU REGARDING COMPUTATION TIME AND ACCURACY.

	SSW		SSFW		SSFW/GPU	
L	1	2	1	2	1	2
Time (s)	0.063	0.11	0.062	0.13	1.6	1.7
Error (dB)	-50.2	-50.9	-51.5	-54.4	-49.7	-50.4

Figure 1 and Table I demonstrate that the proposed SSFW method achieves accuracy comparable to SSW and closely matches the results from SSF. The only discrepancy between SSFW and SSF arises from the hard threshold compression applied in the former. As anticipated, the computation times for SSW [5] and SSFW are nearly identical, although this particular scenario does not fully showcase the advantages of SSFW, as it does not involve ground effects or refraction. Regarding the GPU implementation, a similar observation holds: the domain size in this case is relatively small, so the CPU implementation performs better. Additionally, the GPU implementation is carried out using PyTorch's built-in functions, without further optimization in this instance

B. Propagation in an atmospheric duct over a planar ground

In this test, we consider a larger domain that includes ground reflections and a tropospheric duct. Specifically, the domain spans 30 km in x and 512 m in z. The source is the same as before, except that it is positioned at $z_s = 50$ m, with a width of $w_0 = 5$ m. Additionally, we incorporate a PEC ground and model the atmospheric duct using a tri-linear refractive index profile. The wavelet parameters remain unchanged, and the field computed at the final iteration is shown in Figure 2. In addition, as for the previous test, we provide a comparison in Table II.

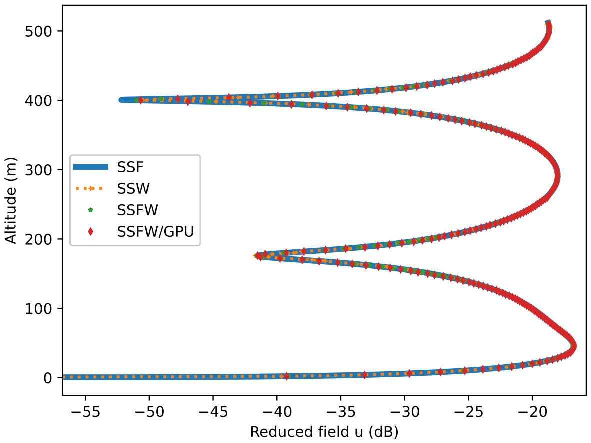

Fig. 2. Reduced field u in dB at $x_{\max}$ computed with different methods for the realistic test case.

TABLE II
COMPARISON OF SSW, AND SSFW ON CPU OR GPU REGARDING COMPUTATION TIME AND ACCURACY FOR THE SECOND TEST.

	SSW		SSFW		SSFW/GPU	
L	1	2	1	2	1	2
Time (s)	0.1	0.21	0.09	0.18	1.8	1.9
Error (dB)	-52.8	-51.7	-72.5	-64.9	-51.5	-43.9

Here, the accuracy of SSFW surpasses SSW's, while maintaining a similar computation time. Unlike the approach in [8], incorporating refraction and the local image method directly in the wavelet domain introduces no additional error. This validates both the developed strategy and the use of frames. The computation time for the GPU implementation remains comparable to that of the previous test, despite slight accuracy differences attributed to the PyTorch implementation. These results confirm the method's effectiveness, emphasize its advantages, and demonstrate the feasibility of GPU implementation, even though further optimization is necessary.

V. CONCLUSION

This article introduced SSFW, a wavelet frame-based method for simulating tropospheric long-range propagation. This approach employs a split-step method to iteratively compute field propagation directly in the wavelet domain. Furthermore, we proposed a local image method in the wavelet domain to avoid the need for going back to the spatial domain. Additionally, the use of frames simplifies the methodology compared to SSW, enabling efficient GPU implementation.

Numerical tests validate the accuracy and effectiveness of SSFW, compared to SSW. The inclusion of ground effects and refraction directly in the wavelet domain does not introduce any additional errors, further confirming the robustness of the proposed approach.

Future work will focus on incorporating terrain reliefs, which should be easily integrated using a staircase model thanks to the translation invariance property of frames. Moreover, efforts will be directed toward optimizing the GPU implementation to enhance its efficiency.

REFERENCES

[1] M. Levy, *Parabolic equation methods for electromagnetic wave propagation.* No. 45, IET, 2000.

[2] J. R. Kuttler and R. Janaswamy, "Improved Fourier transform methods for solving the parabolic wave equation," *Radio Science*, vol. 37, no. 2, pp. 1–11, 2002.

[3] 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 Transactions on Antennas and Propagation*, vol. 65, no. 4, pp. 1988–1995, 2017.

[4] H. Zhou, R. Douvenot, and A. Chabory, "Modeling the long-range wave propagation by a split-step wavelet method," *Journal of Computational Physics*, vol. 402, p. 109042, 2020.

[5] T. Bonnafont, R. Douvenot, and A. Chabory, "A local split-step wavelet method for the long range propagation simulation in 2D," *Radio science*, vol. 56, no. 2, pp. 1–11, 2021.

[6] R. Douvenot, "SSW-2D: some open-source propagation software introducing split-step wavelet and wavelet-to-wavelet propagation techniques," in *2023 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC)*, pp. 020–024, IEEE, 2023.

[7] S. Mallat, "A wavelet tour of signal processing," 1999.

[8] 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 Transactions on Antennas and Propagation*, vol. 70, no. 3, pp. 2137–2148, 2021.

[9] T. Bonnafont, R. Douvenot, and A. Chabory, "Determination of the thresholds in split-step wavelet to assess accuracy for long-range propagation," *Radio Science Letters*, vol. 3, 2021.