A local split-step wavelet method for the long range propagation simulation in 2D

Thomas Bonnafont, Rémi Douvenot, Alexandre Chabory

A local split-step wavelet method for the long range propagation simulation in 2D

Thomas Bonnafont, Rémi Douvenot, and Alexandre Chabory,
ENAC, Université de Toulouse, France
thomas.bonnafont@univ-pau.fr, remi.douvenot@enac.fr

January 27, 2021

keypoints

- The efficiency of a long-range propagation model, Split-Step-Wavelet, is improved using wavelet translation properties and finite supports.

- The propagator generation is shown to not depend on geometry. Efficiency gains are assessed on tropospheric and radio-occultation scenarios.

- This improvement adds versatility since adaptative steps can be introduced. This allows a possible generalization to 3D configurations.

Abstract

Modeling long-range propagation of electromagnetic waves is necessary to study the performance of systems, for applications such as radar or navigation. Such models generally rely on split-step Fourier (SSF) because large mesh sizes can be used. The split-step wavelet method (SSW) is a recently developed method allowing to perform the same simulations as with SSF but in a shorter computation time. This method requires the pre-computation of a free-space propagator. Up to now, one limitation of SSW is that the steps must remain constant during the propagation. In this paper, we propose an improvement of SSW in terms of memory size and versatility. This improvement relies on the use of a set of propagators, *i.e.,* the propagation of elementary wavelets. The limited support of wavelets renders the computation of the set of propagators fast, approximately as fast as one step of propagation with SSW. First, a numerical test shows the advantage in terms of computation time. Second, a numerical experience shows the advantage in terms of memory. Finally, the SSW method is applied as the direct method for a radio occultation configuration.

1 Introduction

Long-range propagation of electromagnetic waves is a topic of great interest for a wide range of systems in telecommunications, navigation, and surveillance. For these applications, propagation is typically studied in the presence of relief and/or refraction. The most common method to handle such situations is the parabolic wave equation (PWE) which models the forward propagation from a source based on the paraxial approximation of the Helmholtz equation [1].

Our main goal is to obtain a fast and accurate method to model the forward long-range propagation in 3D. For which the computation time and memory size must be limited. In this article, we focus on the 2D case, in order to obtain an improved method in both requirements. This is an essential step toward an efficient 3D method.

In 2D, to compute the propagation with the PWE, two methods can be used: the split-step Fourier method (SSF) [2] or the finite difference method (FD) [3]. The FD has the advantage of a straightforward implementation of complex relief and atmosphere conditions. However, it is not time-efficient for large domains since the needed horizontal grid step is of order λ [1]. SSF has the advantage of a larger grid step, up to 100λ, which explains its popularity for long-range propagation.

SSF takes into account a slowly varying refractivity index, relief, and ground composition. The propagation on one step is divided into two terms. The first one corresponds to the propagation in free space. It is accounted for in the spectral domain (Fourier domain). The second one is a phase screen taking into account the refraction in the spatial domain. The effect of relief is also considered in the spatial domain with various methods [4, 5, 6]. An impedant ground can be taken into account using the change of variable given by the discrete mixed Fourier transform (DMFT) [2]. A discrete SSF (DSSF) [7] has recently been developed to add self-consistency to this theory. Deriving a complete method from discrete equations allows avoiding spurious solutions.

Wavelets [8, 9] are widely used in signal theory and image processing for denoising and compression, *e.*g., JPEG2000. Wavelets are short oscillating functions localized in both space and frequency. In electromagnetics, they are essentially used because of their compression property in integral equations [10, 11] and time-domain methods [10, 12]. In the domain of propagation modeling, [13, 14] have used wavelets as test and approximation functions for a finite-element implementation of the PWE.

Another advantage of the wavelets is the fast wavelet transform (FWT) [9] of complexity $O(N)$, with N the number of elements of the vector. This complexity is better than the fast Fourier transform (FFT) which is in $O(N \log(N))$. Based on these considerations, the split-step wavelet method (SSW) has recently been introduced [15, 16]. [15] have shown that the com-

plexity of SSW is lower than SSF. The field is iteratively computed as with SSF, the wavelet domain replacing the spectral domain. For the propagation step, a pre-computed propagator is needed. [15] use a matrix containing all the wavelet-to-wavelet propagations on the domain. The filling of this matrix requires computing few propagations of wavelets, and FWT that are duplicated. Besides, this method has the advantage of a fast sparse matrix-vector multiplication for modeling the propagation in the wavelet domain.

However, it requires the storage of all the duplicated wavelet propagations. Moreover, the filling of the matrix requires a significant time. For this reason, the propagator must be calculated once for all, preventing from using a non-uniform grid. This strategy is specifically problematic for huge domains, as can be seen in radio occultation, and does not permit a 3D generalization in its current state. This method [15] that requires a matrix, is denoted by matrix SSW (mSSW).

In this article we propose an improved version of SSW, denoted by local SSW (lSSW), using properties of the wavelets to add versatility to the method and avoid large storage. Duplications of the matrix are avoided to store only the essential information with few propagators. Besides, the limited support of the elementary wavelets is accounted to significantly speed up the propagator computation.

The article is organized as follows. In Section 2, we briefly introduce the SSW method. In Section 3 we present the computation of the propagators. In Section 4, numerical experiments illustrate the interest of this work. First, the improvement in terms of time efficiency using the wavelet support is shown. Then, we show that the memory required with lSSW is better (lower) than with mSSW. Finally, lSSW is used as the direct method to obtain the propagation modeling in a radio occultation scenario.

2   The split-step wavelet method

For the article, an $\exp(j\omega t)$ time dependence is assumed. We aim at solving iteratively the wide-angle PWE, equation (3.19) in [1], using a split-step method based on the wavelet transform in place of the Fourier transform usually used [2]. The PWE with x the direction of propagation is defined as

$$\frac{\partial u}{\partial x} = -j\left(\sqrt{k_0 + \frac{\partial^2}{\partial z^2}} - k_0\right)u - jk_0(n-1)u, \qquad (1)$$

where u corresponds to the reduced field as defined by [17], k_0 to the wave number and n to the slowly-varying refractive index.

2.1   Configuration and discretization

We only compute the 2D – assuming a y-axis invariance – forward, *i.e.*, towards $x > 0$, propagation over an impedance ground. The field is assumed

3

to be known at $x = 0$ and the source is placed at $x \leq 0$. The field can be decomposed in transverse electric (TE) and transverse magnetic (TM) components with respect to z. We only study the TE case, but the adaptation to the TM case is straightforward.

The domain is of size $[0, x_{\max}]$ on x and $[0, z_{\max}]$ on z. On the z-axis, the grid is given by

$$z_{p_z} = p_z \Delta z \text{ with } p_z \in 0, \cdots, N_z, \tag{2}$$

with $N_z = \dfrac{z_{\max}}{\Delta z}$ the number of points. The x-axis is also discretized with N_x points and a step Δx. The discrete version of the field $u(x, z)$ along z is denoted by $u_x[p_z]$. From now on computations are done in this discrete domain.

2.2   The multilevel discrete wavelet transform

In this section, we briefly present the multilevel discrete wavelet transform that is used [9]. The transform is applied in this article to u_{p_x}. Wavelets are functions of zero mean. A wavelet family is constructed by dilating on L levels and translating along the z-axis a mother wavelet ψ. Wavelets ψ_{l,p_z} are then obtained. L corresponds to the maximum level of decomposition. Dilated wavelets cover different parts of the spectrum. With l increasing lower frequencies of the spectrum are covered by the wavelets. Finally, a scaling function ϕ_{L,p_z} of non-zero mean is added and translated (index p_z) to obtain an orthonormal basis. This last function covers the lowest part of the spectrum.

The discrete wavelet transform associates u_x with a sequence of coefficients $a_{L,p}$ and $d_{l,p}$, such that

$$u_x[p_z] = \sum_{p \in \mathbb{Z}} a_{L,p} \phi_{L,p}[p_z] + \sum_{l \in [1,L]} \sum_{p \in \mathbb{Z}} d_{l,p} \psi_{l,p}[p_z]. \tag{3}$$

This decomposition is denoted by W.

In practice, the computation is performed in a finite domain of size N_z. Therefore, the translation coefficients p are limited to $[0, N_z/2^l - 1[$ for $l \in [1, L]$, respectively. For the scaling function $p \in [0, N_z/2^L - 1[$.

For conciseness, the wavelet coefficients are from now on denoted by

$$U_l[p] = \begin{cases} a_{L,p}, & \text{for } l = 0, p \in [0, N_z/2^L - 1] \\ d_{l,p}, & \text{for } l \in [1, L], p \in [0, N_z/2^l - 1]. \end{cases} \tag{4}$$

The decomposition (3) becomes

$$u_x[p_z] = \sum_{p=0}^{N_z/2^L - 1} U_l[p] \chi_0[p_z - p] + \sum_{l=1}^{L} \sum_{p=0,}^{N_z/2^l - 1} U_l[p] \chi_l[p_z - p], \text{ for } p_z \in [0, N_z - 1] \tag{5}$$

with

$$\chi_l[p_z - p] = \begin{cases} \phi_{L,p}[p_z], & \text{for } l = 0, p \in [0, N_z/2^L - 1], \ p_z \in [0, N_z - 1] \\ \psi_{l,p}[p_z], & \text{for } l \in [1, L], p \in [0, N_z/2^l - 1], \ p_z \in [0, N_z - 1]. \end{cases} \tag{6}$$

By choosing well the wavelet family and the maximum level of decomposition lots of coefficients are close to 0 due to the fast decrease of the coefficients. Thus, the coefficients below a certain threshold V_s are set to 0

$$U_l^c[p] = \begin{cases} 0 & \text{if } |U_l[p]| \leq V_s, \\ U_l[p] & \text{otherwise}. \end{cases} \tag{7}$$

This compression, denoted by operator C, is called "hard threshold" compression [9]. This allows to speed-up the propagation by keeping only a small numbers of coefficients.

2.3   Overview of the propagation method

SSW is an iterative method to compute the field at increasing distances from the source, by going back and forth from the wavelet domain to the spatial domain. The ground is supposed to be a PEC. The scheme in Figure 1 sums up the SSW algorithm. A step of SSW on Δx with u_x known is computed as follows:

1. The sparse vector of the wavelet coefficients of u_x is obtained using a FWT (operator W) and a compression of threshold V_s (operator C)

$$U_x = CW u_x. \tag{8}$$

2. The propagated non-zero coefficients $U_{x+\Delta x}$ are obtained using the propagator P

$$U_{x+\Delta x} = PU_x. \tag{9}$$

The propagator P corresponds to a scattering matrix. In [15] this propagator is a pre-computed matrix denoted by P_m. This latter stores all the wavelet-to-wavelet propagations such that the propagation is a matrix-vector product. In this article we propose the use of a minimal size set of local propagators, denoted by P, and described in Section 3.

3. The free space propagated field $u_{x+\Delta x}^{\text{fs}}$ is obtained by applying an inverse FWT

$$u_{x+\Delta x}^{\text{fs}} = W^{-1}U_{x+\Delta x}. \tag{10}$$

$u_{x+\Delta x}^{\text{fs}}$ is expressed in the space domain.

4. To take into account a slowly varying refractive index, the phase-screen method [17] is applied

$$u_{x+\Delta x} = R u^{\text{fs}}_{x+\Delta x}, \tag{11}$$

where R is a diagonal operator, with diagonal elements given by

$$R[p_z, p_z] = \exp\left(-j k_0 \left(n_x[p_z] - 1\right) \Delta x\right). \tag{12}$$

To take into account the ground, as in [15], the local image method is used. It is based on a local replica of the field on few additional points that simulates a reflection coefficient of value -1. By clearing and updating this additional field at each step, the reflection is accurately accounted and spurious fields are avoided [15]. To account for an impedance boundary on the ground, the propagation in the wavelet domain is applied to w, obtained after applying the DMFT on u [2].

The relief is modeled with a staircase model [1], *i.e.*, as a sequence of horizontal steps. The field is translated for increasing relief. Then the field is propagated as over a planar ground if outside the relief and set to 0 inside the relief. This operator is denoted by L. Other methods can be used as for DSSF [18, 4, 5].
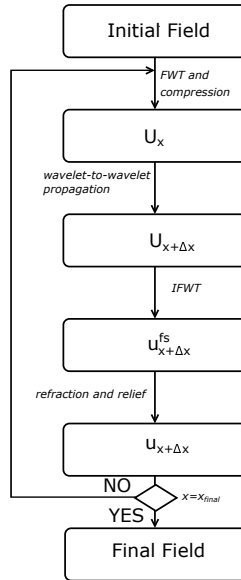

Figure 1: Overview of SSW.

Finally, the domain is limited to $z_{\max}$. At the top the energy must propagate towards infinity and the boundary condition must be transparent. In this work an absorbing apodization layer, denoted by H is applied in the

spatial domain. The Hanning window is chosen, for which the diagonal elements are defined by

$$H[p_z, p_z] = \begin{cases} 1 & \text{if } p_z \in [0, N_z], \\ \frac{1}{2}\left(1 + \cos\left(\pi\left(\frac{p_z \Delta z}{z_{\max}} - 1\right)\right)\right) & \text{if } p_z \in [N_z, N_z + N_{\text{apo}}]. \end{cases} \tag{13}$$

3 Propagation with the local propagators

In this section, the local propagation is introduced. First, a set of propagators, denoted by P, is computed in order to store the wavelet-to-wavelet propagations. Second, a method to propagate all the non-zero wavelet coefficients of the field using this set is described.

3.1 Computing the propagators

In this section, we introduce the efficient computation of the set of propagators. Two important properties of wavelets are used. The invariance by translation property to compute and store only the essential information. The small size of the support of one wavelet, denoted by N_l, to reduce the computation time. We note N_l^{p} the support after propagation, *i.e.*, the subset of the domain where the propagated wavelet is localized.

The propagations of as few as possible elementary wavelets are stored and used to propagate all the wavelet coefficients of the field. These propagations are stored in a set denoted by P. P_{l,p_t^l} corresponds to a N_l^{p}-size vector, $N_l^{\text{p}} \ll N_z$, with the wavelet coefficients of a propagated wavelet of level l at position p_t^l. Then, for each coefficient of the wavelet decomposition of the field, the associated propagator P_{l,p_t^l} returns the propagated coefficients. The vector of wavelet coefficients of the propagated field is obtained by summing all these elementary propagations. These steps amount to the propagation, denoted by operator P.

3.1.1 Computing P_{l,p_t^l}

The space grids of the wavelets are dilated by 2 at each level. Consequently, the decomposition of a wavelet at position p of level l is 2^{L-l}-periodic at each level l. Considering this, the propagation operator in the wavelet domain is 2^L-periodic. Therefore, only 1 period for each level is stored such that the memory requirements are minimum, see Figure 2. We store no redundant information, whereas the scattering matrix used in mSSW contains all the propagations for all levels and positions. Considering two levels l and l' we have the translation property defined as follows:
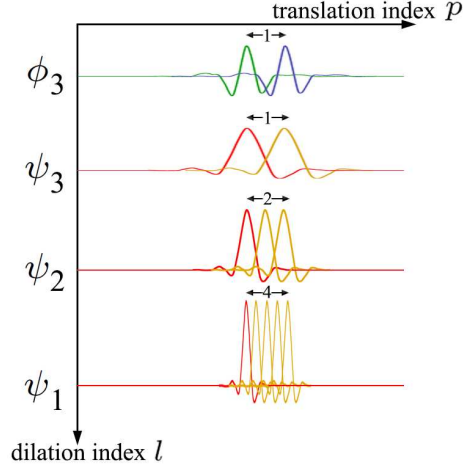
Figure 2: 1 period of the wavelet space grid with $L = 3$.

- if $l = l'$, then a translation of 1 on l corresponds to the same translation on l'.

- if $l' < l$, then there are $2^{l-l'}$ times as many wavelets on l' than on l and a translation 1 on l corresponds to a translation of $2^{l-l'}$ on level l'.

- if $l' > l$, then it is the inverse case, and on l' there are $2^{l'-l}$ times less wavelets than on l. Thus, a translation of $2^{l'-l}$ on l corresponds to a translation of 1 on l'.

Algorithm 1 shows how to compute the set of propagators using this property. The computation of this set can be decomposed into two steps. First, for each level l, the wavelet $\chi_{l,0}[p_z] \forall p_z \in [0, N_z - 1]$ at $x = 0$ is propagated on a distance Δx using DSSF. The wavelet $\chi_{l,\Delta x}[p_z]$ is obtained. Other methods can be used to compute the wavelet propagations, *e.g.*, [13]. It could increase precision at the cost of increasing the computation time.

Secondly, the wavelet field $\chi_{l,\Delta x}[p_z]$ is translated of $p_t^l \in [0, 2^{L-l}[$ to obtain $\chi_{l,\Delta x}[p_z + p_t^l] \forall p_z + p_t^l \in [0, N_z - 1]$ for all the needed translations. Then, it is decomposed and compressed (with threshold V_p) to obtain P_{l,p_t}. These propagations are then stored for all levels l and translations p_t^l as a set of propagators $\left(P_{l,p_t^l}\right)_{l \in [0,L], p_t^l \in [0, 2^{L-l}[}$. Using the translations we only need to compute $L + 1$ propagations and to perform 2^L FWT. Thus, only 2^L propagators are stored. [15] use the translation property to efficiently fill the matrix by duplicating the propagations, thus redundant information was stored [7]. Here, the set of propagators corresponds to the minimal subset of this matrix required for the propagation.

Algorithm 1 Computing the propagators

1: **Inputs:** maximum level L, wavelet basis
2: **Output:** set of propagators $\left(P_{l,p_t^l}\right)_{l\in[0,L],p_t^l\in[0,2^{L-l}[}$
3: \\ at each wavelet level
4: **for** $l \in [1,L]$ **do**
5:     $\chi_{l,0}[p_z] \leftarrow$ wavelet at level l and position 0 .
6:     $\chi_{l,\Delta x}[p_z] \leftarrow$ propagated wavelet with DSSF.
7:     \\ 2^{L-l} periodicity of the decomposition
8:     **for** $p_t^l \in [0,2^{L-l}[$ **do**
9:         $\chi_{l,\Delta x}[p_z + p_t^l] \leftarrow$ translate $\chi_{l,\Delta x}[p_z]$ along p_t^l points.
10:        $P_{l,p_t^l} \leftarrow$ apply FWT and compression to $\chi_{l,\Delta x}[p_z + p_t^l]$.
11:    **end for**
12: **end for**

3.1.2   Reducing the computation time with the wavelet support

Now the small size of the support N_l of the wavelet is used to speed up the computation of this set. Each wavelet at level l has a support of finite size theoretically known [9]. In practice, the support is much smaller than the size of the domain ($N_l \ll N_z$). Therefore, propagation can be performed on a reduced domain, of size N_l^{p}. This latter depends only on the level l and the step size Δx. To assess the value of this parameter, we use the support before propagation N_l, and the angle of propagation corresponding to the validity domain of the wide-angle approximation ($\pm\pi/4$) [1] as shown in Figure 3. Using the propagation step Δx, and the discretization step Δz the number of points after propagation is given by

$$N_l^{\mathrm{p}} = \left\lceil N_l + \sqrt{2}\frac{\Delta x}{\Delta z} \right\rceil, \tag{14}$$

where $\lceil x \rceil$ denotes the ceiling function, which gives the minimum integer greater than or equal to x.

For example, with $\Delta x = 100$ m and $\Delta z = 1$ m using the symlet with $n_v = 6$ the support of the mother wavelet is $2n_v - 1 = 5$ m, with $L = 3$. Thus, for $l \in [1,L]$ we have $N_l = l(2n_v-1) = 5l$ m and $N_l^{\mathrm{p}} = 5l + 100\times\sqrt{2}$. Hence, the support after propagation is at most 156 points. Since the propagation domains are in practice of order thousands of points this latter is much smaller than the total domain. Besides, N_l^{p} remains the same even with N_z increasing if Δx and Δz remain constant.

Thus the DSSF propagation on N_l^{p} is much faster than on the overall domain.

Besides N_l^{p} does not depend on N_z. Then, the propagation time for computing P is independent of N_z. This method is cheap in terms of memory size occupation. Moreover, for large domains ($N_l^{\mathrm{p}} \ll N_z$) the computation
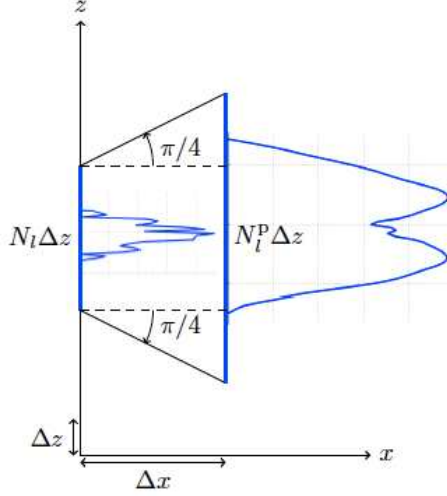
9

Figure 3: Wavelet supports before and after propagation on one step.

of the set, of complexity $O(\sum_l N_l^{\mathrm{p}} \log(N_l^{\mathrm{p}}))$ is faster than the computation of one step of propagation, of complexity $O(N_z)$. Therefore, the propagator can be computed on the fly during the propagation, adding versatility regarding the domain and step sizes.

3.2   Local propagation

In this section, the local propagation method using the previously defined set is introduced. The presentation firstly focuses on one non-zero wavelet coefficient. The generalization consists in adding the contributions of all non-zero coefficients to finally obtain the propagated wavelet coefficients $U_{x+\Delta x}$.

The computation details can be found in Algorithm 2. First, the non-zero coefficients $\alpha_l[p_z]$ of the wavelet decomposition of the field U_x are obtained with a FWT and compression (with threshold V_s). For each coefficient, we need to compute the corresponding propagator P_{l,p_t^l} in the set, from the level l and position p_z of $\alpha_l[p_z]$ following

$$p_{t_{p_z}}^l = p_z \mod 2^{L-l}. \tag{15}$$

This latter must be computed because we only store the elementary propagations in the set. Then, we compute the elementary propagation for this coefficient by multiplying the local propagator by $\alpha_l[p_z]$. Finally, all these elementary propagations are translated to original position p_z of the coefficient and summed as

$$U_{x+\Delta x} = \sum_{l,p_z} \alpha_l[p_z] P_{l,p_{t_{p_z}}^l}[\cdot + p_z] \tag{16}$$

10

to obtain the propagated wavelet coefficients. An inverse FWT finally gives the propagated field in free space.

With compression on the signal V_s and on the propagators V_p an error accumulates throughout iterating on the domain. Using the formula obtained in [15], in Section 5.2, both thresholds are set to obtain an expected error for a given scenario.

Algorithm 2 Free-space propagation with local propagators

Inputs: field u_x, elementary propagator list $\left(P_{l,p_t^l}\right)_{l\in[0,L],p_t^l\in[0,2^{L-l}[}$

Output: propagated field $u_{x+\Delta x}^{fs}$ in free-space

$U_{x+\Delta x} \leftarrow$ empty list of wavelet coefficients (initialization)

$u_x \leftarrow$ field at p_x.

$U_x = CWu_x \leftarrow$ sparse wavelet representation of the field

for each non-zero coefficient $\alpha_l[p_z]$ of U_x **do**

    $l, p_z \leftarrow$ level and position of the coefficient

    $p_{t_{p_z}}^l \leftarrow$ choose the operator number using (15)

    $P_{l,p_{t_{p_z}}^l} \leftarrow$ corresponding elementary propagator

    $U_{x+\Delta x} \leftarrow U_{x+\Delta x} + \alpha_l[p_z]P_{l,p_{t_{p_z}}}$

end for

$u_{x+\Delta x}^{fs} \leftarrow \text{IFWT}(U_{x+\Delta x})$

---

4 Numerical tests

In this section, numerical tests are performed. First, the interest in limiting the computation of the propagator to the wavelet support is analyzed in terms of time efficiency. Then, a realistic test case for tropospheric propagation is performed. We compute a long-range propagation with a tropospheric duct and relief using DSSF, lSSW, and mSSW [15]. Finally, propagation in a radio occultation configuration is proposed where the advantages of lSSW over mSSW in terms of computation time and memory size are shown. All numerical tests are done using a standard computer with 8 cores and a 2.7 GHz processor.

For all experiments, the same wavelet parameters are used. The symlet6 family is used, because this latter is of compact support and almost symmetric [9, 8]. The maximum level L is chosen as $L = 3$. These choices are discussed in [15].

4.1 Computing the propagators

The aim of this test is to show the improvement in terms of computation time using the wavelet supports for their propagations. We compute the

set of propagators with DSSF for several values of N_z. The propagators are computed on N_l^{p} points and compared to propagators obtained by using the total vertical domain N_z. We test this method for wavelets of each level $l \leq 3$. The steps Δx and Δz are 50 m and 0.5 m, respectively. Figure 4 shows the results.
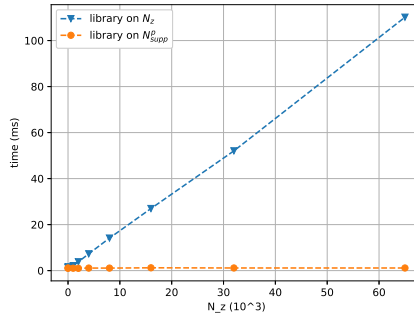

Figure 4: Computation time (ms) to obtain the set of propagators with DSSF on the total domain and on N_l^{p}.

Figure 4 shows that the time to generate the propagators on N_l^{p} is independent of N_z, as expected. Limiting the computation domain saves a lot of computation time for large N_z. Numerical tests show that the normalized difference between both methods is at maximum -250 dB. This shows the interest of considering the support after propagation N_l^{p}. The next simulation illustrates that the local method with N_l^{p} does not add any error in a realistic test case.

4.2 Realistic test case

The aim of this section is to compare lSSW and mSSW in terms of computation time, memory size, and accuracy in a realistic scenario.

The propagation of a complex source point (CSP) between Pau $(43°17'51''\mathrm{N}, 0°22'07''\mathrm{E})$ and Toulouse $(43°36'15''\mathrm{N}, 1°26'36''\mathrm{E})$ is computed using both versions of SSW (lSSW and mSSW). An atmospheric duct is added. The relief between the two cities is taken into account.

The CSP parameters are: a frequency $f = 3$ GHz, with coordinates $x_s = x_{\mathrm{w}0} + jk_0 W_0^2/2$, with $x_{\mathrm{w}0} = -50$ m and $W_0 = 5$ m and $z_s = 50$ m from the ground.

First, we consider a domain of size $x_{\max} = 150$ km (distance between the two cities) and $z_{\max} = 1024$ m. The grid size is $\Delta x = 200$ m in horizontal and $\Delta z = \lambda = 0.1$ m in vertical. The impedance ground is of parameters $\epsilon_r = 20.0$ and $\sigma = 0.02$ S/m.

We consider a refractive duct modelled by a trilinear refractive index, *e.g.*, [19]. The parameters are $M_0 = 330$ M-units, $z_b = 241$ m, $z_t = 391$ m,

with gradients $c_0 = 0.118$ M-units/m and $c_2 = -0.1$ M-units/m.

The signal and propagators thresholds are $V_s = 2.1 \times 10^{-5}$ and $V_p = 4.3 \times 10^{-6}$, respectively, so as to obtain an error of -30 dB at the last iteration using the formula given by [15]. The image layer for applying the local image method is of size $0.1z_{max}$.

The electric field using lSSW is represented in Figure 5. The effects of both the atmosphere – bending of the wave – and the relief – interference and shadow zone – can be seen on that figure. For lSSW and mSSW, the RMS difference with DSSF at each iteration p_x is shown in Figure 6. We can see that the RMS error evolution is exactly the same with both methods.
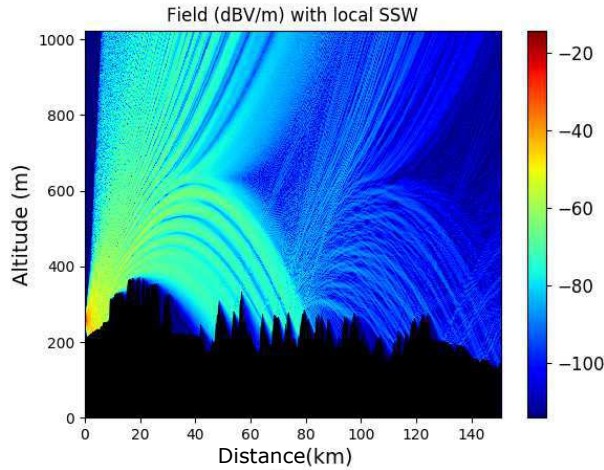

Figure 5: Electric field (dBV/m) in the vertical plane obtained by lSSW.

Table 1 presents the times to compute the propagator, one step of propagation with each method, the total time for the propagation, and the memory needed for mSSW and lSSW. The initialization time corresponds to the time to compute the set of propagators or the matrix.

Table 1: Times and memory size needed with lSSW and mSSW.

method	mSSW	lSSW
Initialization time (s)	135	0.1
One step propagation (s)	0.01 to 0.3	0.01 to 0.4
Total time (s)	160	27.1
Propagator memory size	252 MB	117 kB

Results show that lSSW is better in terms of computation time and memory size than mSSW. First, in terms of propagation lSSW is as fast as mSSW but the initialization time is much larger for mSSW: 1000 times larger. Thus lSSW allows being faster overall. Also, the time to compute
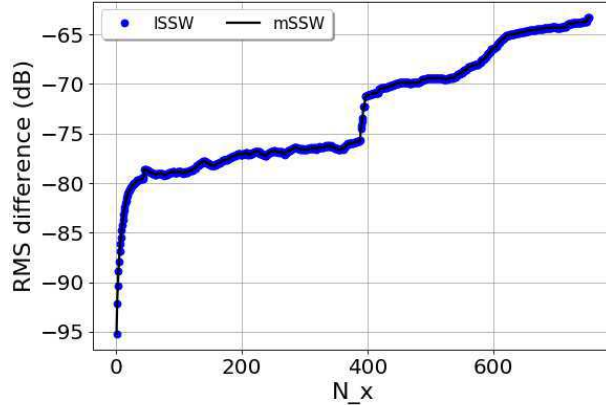
13

Figure 6: RMS difference of lSSW and mSSW with DSSF at each iteration.

the set of propagators (0.1 s) is of the order of one step of propagation and allows to update the propagators with acceptable effort if needed. In terms of memory size lSSW needs 2000 times less memory in this scenario.

Then, we double the number of vertical points ($z_{\max} = 2048$ m) to show further the interest of lSSW against mSSW. All other parameters remain the same. We obtain the results of Table 2 for the times and memory size.

Table 2: Times and memory size needed with $z_{\max} = 2048$ m with lSSW and mSSW.

method	mSSW	lSSW
Initialization time (s)	285	0.1
One step propagation (s)	0.02 to 0.4	0.02 to 0.5
Total time (s)	339	56.1
Propagator memory size	505 MB	117 kB

As expected, the time to initialize and the memory size of the propagator remain the same with N_z increasing. Regarding the time and memory size needed with mSSW, lSSW becomes very attractive in this case, since both parameters have significantly increased with mSSW. Finally, the test shows that the difference with DSSF remains the same at the last iteration. Thus, the new version of SSW allows to compute fields for large scenarios with small memory requirements. In the next section, lSSW and mSSW are compared with a scenario where the memory size and the computation time of the propagator are the main concerns.

14

4.3 Radio occultation scenario

Radio occultation (RO) consists in inferring atmospheric data from the radio frequency (RF) link between two satellites in the limit of line of sight. This is generally used with the RF link between GPS and LEO satellites [20, 21, 22]. In this configuration, the propagation is altered by the refractive condition (troposphere and ionosphere) of the Earth. This altered data carry information about the atmospheric conditions. Since 1995 inverse RO techniques have been used to estimate the physical properties of the Earth atmosphere using inversion techniques [23, 24]. The problem is here simplified to a 2D model of RO where both satellites are in the same plane.

RO inversion methods need to be validated. For that direct models that compute the field for a given atmosphere condition are necessary. These models should be fast and accurate. Indeed in a large domain, like RO, the computation time required for SSF becomes prohibitive. Other methods have been proposed to eliminate this constraint. Using the Gaussian beam as an asymptotic solution of the PE, [25] have proposed a fast method for field propagation in large domains. This method has then been applied to RO [26, 27]. mSSW has also been used as a direct method for RO scenarios [28] but both the time and memory size needed by the matrix become prohibitive to reduce the vertical step. In this section, we use lSSW as the direct method to obtain the field on the overall domain of a RO scenario. With its low requirements in time and memory size for the propagator, lSSW helps to decrease the computation time and thus to compute the propagation with improved grid size.

The following RO scenario is considered here. The Cartesian system with (x, z) coordinates is used. We consider a GPS satellite at frequency $f_0 = 1.575$ GHz with a gain of 16 dBi and a power of 25 W. The transmitter is located at $x_{T_x} = -24647$ km, modeled here by a plane wave illumination at $x = 0$. The receiver is a LEO satellite located around $z = 0$ at $x = 4553$ km. The total distance between both satellites is $x_{\max} = 29200$ km, corresponding to the x-axis coordinate of the LEO satellite. The goal is to obtain the field near the LEO satellite.

No ionosphere is included in the simulation. This could be added using ITU-R P531. The refractive index in the troposphere is computed using ITU-R P834. More realistic models could use the Weather Research and Forecasting model (WRF) [29, 30] or the AROME model from Météo-France [31].

The configuration of the computation domain is shown in Figure 7. Axis $z = 0$ is placed at the center of this window. Moreover, the first vertical is set at $x_0 = 24986$ km from the GPS satellite. Thus, the horizontal size of propagation is $x_{\max} - x_0 = 4214$ km. The vertical size is chosen to 65.536 km. The steps are set to $\Delta x = 1$ km and $\Delta z = 1$ m $\approx 5\lambda$. The propagation is thus computed on $N_x = 4214$ points and $N_z = 65536$ points. Earth is

considered as a sphere and treated as relief with a staircase model. Before the computation domain reaches the Earth, apodization layers are added at the top and the bottom of the domain. When propagating above the Earth, an apodization window is considered at the top. The ground is considered as a PEC and a local image layer of size $0.1N_z$ is added. Finally, when Earth is no more in the computation domain, apodization layers are added at the top and the bottom of the domain. For the source, a windowed plane wave at $x = 0$ and of the size the vertical domain is used.
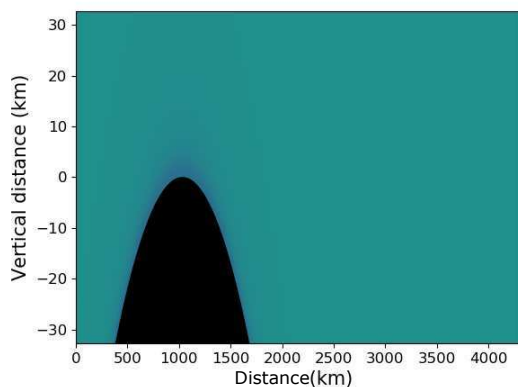

Figure 7: Configuration of the computation domain for the RO scenario.

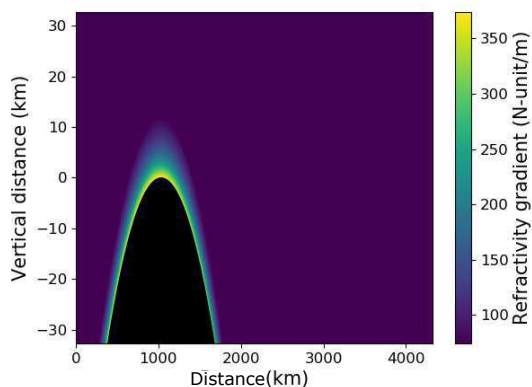The refractivity N is represented in Figure 8. Earth is represented in black.

Figure 8: N index along the propagation scene.

The compression thresholds are set to obtain a negligible final compression error of -30 dB compared to DSSF.

The electrical field computed with lSSW is represented in Figure 9. The

16

two vertical dotted lines show the limits of the propagation over the Earth. We can see the effect of the Earth with the interference pattern of the field and the shadow zone. Also, the maximum error between lSSW and mSSW is still at −140 dB and thus negligible.
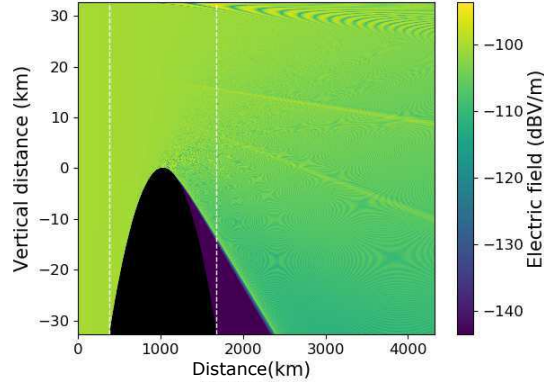

Figure 9: Electrical field (dBV/m) obtained with lSSW.

The results in terms of time and memory size with lSSW and mSSW are stored in Table 3. First, with lSSW the computation time and the memory

Table 3: Times and memory size needed with lSSW and mSSW.

method	mSSW	lSSW
Initialization time (s)	585	0.017
Propagation time (s)	374	466
Total time (s)	959	466
Propagator memory size	759 MB	42 kB

are better than with mSSW. In particular, lSSW initialization time is 50000 less than the one with mSSW, but the propagation time is a bit slower. Overall lSSW is faster than mSSW. For the memory requirement concerns, the impact of the local method becomes clear since lSSW is better of a factor of 25000 against mSSW. Thus with lSSW a smaller grid size is possible.

Also, since the time to generate the propagators is negligible to the propagation, horizontal steps can be changed between vacuum and ground propagations if needed, adding versatility.

To conclude, lSSW is a good method to model the field with a RO scenario, since the computation time and memory requirements are low in comparison to mSSW.

17

5   Conclusion

In this paper, the local method for SSW, namely lSSW, has been introduced. It is based on an efficient strategy to compute the propagators using the support of the wavelet for the propagation, and storing only the essential information.

First, an overview of the SSW method has been given. We have reminded that the wavelet coefficients are propagated using a pre-computed propagator. The propagation can thus either be a matrix-vector product or a sum of local propagations. The other steps (relief and atmosphere taken into account in the space domain) are independent of the choice of the propagation.

Then, the method to compute the local propagators has been comprehensively detailed. Based on the translation property of the wavelets, few propagators are computed and stored. Using the wavelet supports as vertical domains for the propagation of these wavelets allows to be faster and to use less memory. Thus, the propagators are generated independently of the vertical size of the domain for given horizontal and vertical steps. The propagation is then performed by summing all the local propagations associated to the wavelet coefficients of the field.

Numerical tests have shown that lSSW does not add any additional error compared to its matrix counterpart, mSSW. In terms of computation time with the domain growing, the time to propagate the wavelet stays constant. For complex environment simulations and large domain scenarios, the same electric fields as with mSSW are obtained. Nevertheless, the memory size is way better with lSSW and allows to compute propagations in larger domains. The computation time needed for initialization is also significantly reduced. Finally, the limited support of wavelets renders the computation of the set of propagators fast, approximately as fast as one step of propagation with SSW, adding versatility to the method.

This new version of SSW allows to tackle the problem of memory size and computation time for the propagator in mSSW. Since the computation time for the propagators is low compared to the propagation time, adaptative steps can be introduced with this method. Besides the improvements obtained in the memory and time requirements are an essential step towards an efficient 3D version of SSW, where local propagators gain an additional dimension.

acknowledgments

not used, nor created for this research.

References

[1] M. Levy, *Parabolic Equation Methods for Electromagnetic Wave Propagation*. IET, 2000, no. 45.

[2] D. Dockery and J. R. Kuttler, "An improved impedance-boundary algorithm for Fourier split-step solutions of the parabolic wave equation," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 12, pp. 1592–1599, 1996.

[3] D. Lee, G. Botseas, and J. S. Papadakis, "Finite-difference solution to the parabolic wave equation," *The Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 795–800, 1981.

[4] A. E. Barrios, "A terrain parabolic equation model for propagation in the troposphere," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 1, pp. 90–98, 1994.

[5] R. Janaswamy, "A curvilinear coordinate-based split-step parabolic equation method for propagation predictions over terrain," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 7, pp. 1089–1097, 1998.

[6] D. J. Donohue and J. R. Kuttler, "Propagation modeling over terrain using the parabolic wave equation," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 2, pp. 260–277, 2000.

[7] H. Zhou, "Modeling the Atmospheric Propagation of Electromagnetic waves in 2-D and 3-D using Fourier and Wavelet Transforms," PhdThesis, Université Paul Sabatier - Toulouse III, 2018.

[8] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[9] ——, *A Wavelet Tour of Signal Processing*. Academic press, 1999.

[10] T. K. Sarkar, M. Salazar-Palma, and M. C. Wicks, *Wavelet Applications in Engineering Electromagnetics*. Artech House, 2002.

[11] B. Z. Steinberg and Y. Leviatan, "On the use of wavelet expansions in the method of moments (EM scattering)," *IEEE Transactions on Antennas and Propagation*, vol. 41, no. 5, pp. 610–619, 1993.

[12] T.-K. Hong and B. L. N. Kennett, "On a wavelet-based method for the numerical simulation of wave propagation," *Journal of Computational Physics*, vol. 183, no. 2, pp. 577–622, 2002.

[13] A. Iqbal and V. Jeoti, "A novel wavelet-Galerkin method for modeling radio wave propagation in tropospheric ducts," *Progress in Electromagnetics Research*, vol. 36, pp. 35–52, 2012.

[14] ——, "Numerical modeling of radio wave propagation in horizontally inhomogeneous environment using split-step wavelet method," in *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS),*, vol. 1.   IEEE, 2012, pp. 200–205.

[15] 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.

[16] T. Bonnafont, R. Douvenot, and A. Chabory, "Split-step wavelet propagation modelling using local operators," in *2019 URSI International Symposium on Electromagnetic Theory (EMTS)*. IEEE, 2019, pp. 1–4.

[17] J. R. Kuttler and G. D. Dockery, "Theoretical description of the parabolic approximation/Fourier split-step method of representing electromagnetic propagation in the troposphere," *Radio Science*, vol. 26, no. 2, pp. 381–393, 1991.

[18] R. J. Mac Arthur and D. H. O. Bebbington, "Diffraction over simple terrain obstacles by the method of parabolic equation," in *ICAP91, IEE Conf. Pub*, 1991, pp. 2824–2827.

[19] E. Gossard and R. Strauch, "Radar observation of clear air and clouds," *Developments in Atmospheric Science*, vol. 14, 1983.

[20] E. R. Kursinski, G. A. Hajj, J. T. Schofield, R. P. Linfield, and K. R. Hardy, "Observing Earth's atmosphere with radio occultation measurements using the Global Positioning System," *Journal of Geophysical Research: Atmospheres*, vol. 102, no. D19, pp. 23 429–23 465, 1997.

[21] C. Rocken, R. Anthes, M. Exner, D. Hunt, S. Sokolovskiy, R. Ware, M. Gorbunov, W. Schreiner, D. Feng, B. Herman *et al.*, "Analysis and validation of GPS/MET data in the neutral atmosphere," *Journal of Geophysical Research: Atmospheres*, vol. 102, no. D25, pp. 29 849–29 866, 1997.

[22] E. R. Kursinski, G. A. Hajj, S. S. Leroy, and B. Herman, "The GPS radio occulation technique," *Terr. Atmos. Ocean. Sci.*, vol. 11, no. 1, p. 53, 2001.

[23] S. B. Healy and J. R. Eyre, "Retrieving temperature, water vapour and surface pressure information from refractive-index profiles derived by radio occultation: A simulation study," *Quarterly Journal of the Royal Meteorological Society*, vol. 126, no. 566, pp. 1661–1683, 2000.

[24] S. B. Healy and J.-N. Thépaut, "Assimilation experiments with CHAMP GPS radio occultation measurements," *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography*, vol. 132, no. 615, pp. 605–623, 2006.

[25] C.-A. L'Hour, V. Fabbro, A. Chabory, and J. Sokoloff, "2-D propagation modeling in inhomogeneous refractive atmosphere based on Gaussian beams Part I: Propagation modeling," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 8, pp. 5477–5486, 2019.

[26] ——, "2D modeling of the atmospheric refraction based on Gaussian beams," in *2016 10th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2016, pp. 1–3.

[27] ——, "2-D propagation modeling in inhomogeneous refractive atmosphere based on Gaussian beams Part II: Application to radio occultation," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 8, pp. 5487–5496, 2019.

[28] R. Douvenot, A. Chabory, and S. Rougerie, "Modelling the radiowave propagation with a split-step wavelet method for radio occultation," in *2018 22nd International Microwave and Radar Conference (MIKON)*. IEEE, 2018, pp. 562–564.

[29] X.-Y. Huang, Q. Xiao, D. M. Barker, X. Zhang, J. Michalakes, W. Huang, T. Henderson, J. Bray, Y. Chen, Z. Ma *et al.*, "Four-dimensional variational data assimilation for WRF: Formulation and preliminary results," *Monthly Weather Review*, vol. 137, no. 1, pp. 299–314, 2009.

[30] J. G. Powers, J. B. Klemp, W. C. Skamarock, C. A. Davis, J. Dudhia, D. O. Gill, J. L. Coen, D. J. Gochis, R. Ahmadov, S. E. Peckham *et al.*, "The weather research and forecasting model: Overview, system efforts, and future directions," *Bulletin of the American Meteorological Society*, vol. 98, no. 8, pp. 1717–1737, 2017.

[31] Y. Seity, P. Brousseau, S. I. Malardel, G. Hello, P. Bénard, F. Bouttier, C. Lac, and V. Masson, "The AROME-France convective-scale operational model," *Monthly Weather Review*, vol. 139, no. 3, pp. 976–991, 2011.