



Split-step wavelet propagation modelling using local operators

Thomas Bonnafont^{*(1)}, Rémi Douvenot⁽¹⁾, and Alexandre Chabory⁽¹⁾

(1) ENAC, Université de Toulouse, France, e-mail: thomas.bonnafont@enac.fr, douvenot@recherche.enac.fr, chabory@recherche.enac.fr

Abstract

The atmospheric long-range propagation above the ground is of major importance for many ground systems as radars. The split-step wavelet method allows to compute efficiently this propagation, using a matrix-vector product for the free-space propagation. In this paper we propose a local method for the free-space propagation based on library of propagators. From numerical experiments, we show that this new method is cheaper in terms of memory cost whereas the matrix-vector product solution yields faster computation time.

1 Introduction

Modelling the electromagnetic wave propagation is a major topic for numerous applications. The parabolic equation method (PE)[1, 2] is one of the most popular method for those long-range simulations in inhomogeneous atmospheres. To obtain the PE, a paraxial approximation is made, and only the forward propagation is considered

The split-step Fourier method [2], or its discrete counterpart DSSF [3], is efficient and widely used to compute the propagation. DSSF is an iterative method that computes the free-space propagation in the spectral domain and takes the refractivity into account in the space domain with a phase screen. For the ground condition the variable change of the discrete mixed Fourier transform (DMFT) is used [2, 5].

Recently an alternative method based on the wavelet transform has been proposed [3, 4]. The main difference is that instead of going in the spectral domain for the free-space propagation, the field is decomposed in the wavelet domain with the fast wavelet transform (FWT)[6], and propagated with a wavelet-to-wavelet propagation matrix. The FWT has a lower complexity than the fast Fourier transform (FFT) and the wavelet decomposition allows huge compression. Therefore, SSW outperforms DSSF in terms of computation time.

In this paper SSW method is reminded and a new strategy for the free-space propagation is introduced. It is based on local propagators and has the advantage of reducing the memory storage. Both methods are compared with DSSF in terms of accuracy, speed, and memory usage.

The article is organised as follows. In Section 2 an overview of SSW is presented and the methods for the free-space propagation in the wavelet domain are exposed. Section 3 is devoted to numerical tests. The conclusion and discussions are in Section 4.

2 Split-step wavelet method

2.1 Configuration and discretization

The hypothesis of a $\exp(j\omega t)$ time dependence for the field is made, with ω the angular frequency. The aim is to model the forward 2D (x, z) propagation assuming a y invariance over the ground ($z \geq 0$). The field $u(x, z)$ is known at $x = 0$ with a source at $x < 0$. A decomposition in transverse electric (TE) and magnetic (TM) components for the field can be done. Only the TE case is studied here since the same study goes for the TM case.

The domain of size $x \in [0, x_{\max}[$ and $z \in [0, z_{\max}[$ is discretized with a uniform grid

$$x = p_x \Delta x \text{ for } p_x = 0, \dots, N_x - 1, \quad (1)$$

$$z = p_z \Delta z \text{ for } p_z = 0, \dots, N_z - 1, \quad (2)$$

with Δx and Δz the step of discretization for x and z , respectively.

2.2 Wavelet configuration

The FWT [6] is a multilevel decomposition on $L+1$ levels. It is used in signal theory and data compression, see [6]. Here, the wavelet transform is applied on u along the z -axis. The level $l = 0$ contains coefficients associated with the scaling function. It corresponds to the lower part of the spectrum, including the continuous components. The levels $l \in [1, L[$ correspond to wavelets with various dilations. Greater values of l correspond to faster variations of the signal. In this paper the wavelet family used is the "symlet6" that can be used for the FWT and has a compact support. The maximum level of decomposition is $L = 3$ for the FWT. These choices are discussed in [4, 3].

2.3 Overview of the split-step wavelet method

In this section the SSW method [3, 4] is summarized. SSW is an iterative method to compute the propagation going back and forth from a spatial to a wavelet representation of the wave. The propagation of a field u from x to $x + \Delta x$, with the previous grid is performed as follows:

1. The FWT (denoted W) and a compression (denoted C) are applied to the field $u(x, p_z \Delta z)$ such that

$$\tilde{U}(x) = CWu(x, \cdot). \quad (3)$$

with \tilde{U} a sparse vector.

2. The free-space propagation is computed using a propagator (denoted P)

$$\tilde{U}_{\text{fs}}(x + \Delta x) = P\tilde{U}(x), \quad (4)$$

where P represents the free-space propagator in the wavelet domain. That models the wavelet-to-wavelet propagations. This propagator is pre-computed and stored in either a matrix as described in Section 2.4 or in a library of local propagators as introduced in this paper, see Section 2.5.

3. The IFWT (denoted W^{-1}) is applied to the propagated wavelet coefficients to obtain the field $u_{\text{fs}}(x + \Delta x, p_z \Delta z)$

$$u_{\text{fs}}(x + \Delta x, \cdot) = W^{-1}\tilde{U}_{\text{fs}}(x + \Delta x). \quad (5)$$

4. Apodization, atmosphere and relief can be accounted in the spatial domain, represented altogether by the operator D [2]. Therefore, we have

$$u(x + \Delta x, \cdot) = Du_{\text{fs}}(x + \Delta x, \cdot). \quad (6)$$

A Hanning apodization window [1] is chosen for the apodization. The ground condition is taken into account using the local image theorem presented in [3]. This technique efficiently computes the reflection over the ground along the propagation using a small number of extra-points. A phase screen is applied to the field in the space domain to consider a slowly varying atmosphere [1]. Irregular relief is considered with a staircase model [1].

In the following sections 2.4 and 2.5, the strategies for the free-space propagation (4) are detailed.

2.4 Propagation with a pre-computed matrix

In this section the free-space propagation step using a matrix is described. The propagation is performed by a matrix multiplication. This matrix \mathbf{P} contains the wavelet-to-wavelet propagations. The properties of translation and dilation of wavelets are taken into account for filling \mathbf{P} efficiently. The detailed method is given by Zhou *et al.* [3, 4] and summarised below.

At each level l' , one wavelet $\chi_{l',0}(0)$ is propagated on Δx with DSSF (other propagation methods can be used) to obtain $\chi_{l',0}(\Delta x)$.

The wavelet-to-wavelet coefficients are obtained by means of a FWT and compression of $\chi_{l',0}(\Delta x)$. This operation is repeated at each level. Then translations are applied to obtain the others coefficients of the matrix.

For a scenario the matrix is only computed once and can be used again if the domain size and steps are the same. This method has the advantage to be very efficient in terms of computation time. The main disadvantage is that the propagation matrix has a large memory size for a large domain. The local propagator technique is described in the next section to avoid this latter problem.

2.5 Propagation with pre-computed local propagators

This section introduces a new method to compute the free-space propagation step in SSW. The aim is to reduce the memory size required for the propagation using only a library of local propagators.

Indeed since wavelets are obtained by translations and dilations, only a few local elementary propagators are necessary.

Algorithm 1 explains how to construct the library. In this algorithm the level 0 is considered as the level L . First, at each level l , $q_{\text{max}}(l) = 2^{L+1-l}$ elementary propagators are computed. For each propagator a wavelet of level l at the position q , with $q \in [0, q_{\text{max}}(l)]$ is propagated using DSSF. Then the propagator is obtained with FWT to come back in the wavelet domain. A threshold V_m is applied to obtain sparse elementary propagators.

Algorithm 1 Construction of the propagator library

- 1: **Inputs:** maximum level L , wavelet basis
 - 2: **Output:** elementary propagator list $P_{L, q_{\text{max}}(l)}$
 - 3: \forall at each wavelet level
 - 4: **for** $l \in [1, L]$ **do**
 - 5: $q_{\text{max}} \leftarrow q_{\text{max}} = 2^{L-l}$ number of necessary operators.
 - 6: $\chi_{l,0}(0) \leftarrow$ wavelet at level l and position 0.
 - 7: **for** $q \in [1, q_{\text{max}}]$ **do**
 - 8: $\chi_{l,q}(\Delta x) \leftarrow$ translate $\chi_{l,0}(0)$ along q points.
 - 9: $P_{l,q} \leftarrow$ apply FWT and compression to $\chi_{l,q}(\Delta x)$.
 - 10: **end for**
 - 11: **end for**
-

This library is used to compute the propagation of each non-zero coefficient of the wavelet decomposed field. This propagation is detailed in algorithm 2. First the field is decomposed in the wavelet domain (W) and a threshold V_s with a compression (C) is applied. Then each non-zero coefficient of \tilde{U} is propagated using the appropriate elementary propagator. The propagator number depends on the

level l and the position of the coefficient, which is given by the formula

$$q = p \pmod{q_{\max}(l)}, \quad (7)$$

with p the position of the coefficient and $q_{\max}(l)$ the maximum number of operators needed at level l . Then a multiplication of the propagator by the coefficient gives the propagated coefficients.

Algorithm 2 Free-space propagation with local operators

Inputs: field u_x , elementary propagator list $P_{L, q_{\max}(l)}$
Output: propagated field u_{x+dx} in free-space
 $U_p \leftarrow$ empty list of wavelet coefficient
 $u_x \leftarrow$ field at x .
 $U_x = CWu_x \leftarrow$ wavelet decomposition of the field
 $\tilde{U} \leftarrow$ sparse wavelet field
for each non-zero coefficient of \tilde{U} **do**
 $l, p \leftarrow$ level and position of the coefficient
 $\alpha \leftarrow$ coefficient $\tilde{U}(l, p)$
 $q \leftarrow$ choose the operator number using (7)
 $P_{l, q} \leftarrow$ corresponding elementary propagator
 $U_{x+dx} \leftarrow U_{x+dx} + \alpha P_{l, q}$
end for
 $u_{x+dx} \leftarrow$ IFWT(U_{x+dx})

Furthermore, $N_{\text{coeff}} \ll N_{\text{mat}}$, with N_{coeff} the number of coefficients required for the library and N_{mat} the total number of non-zero elements in \mathbf{P} . Therefore, computing the library is more efficient in time and memory size than computing the matrix.

The local propagation on one step can then be summarized as follows:

1. compute the list of elementary propagators with threshold V_m ;
2. decompose the field in the wavelet domain with threshold V_s ;
3. propagate all non-zero coefficients using the suitable elementary propagators and summed;
4. go back in the spatial domain with IFWT and take into account the relief and the atmosphere.

3 Numerical tests

The aim of this section is to compare SSW with a library of propagators and with a matrix in terms of computation time, memory usage and accuracy. To do that, we compare the results of these 2 methods with DSSF.

The propagation of a complex source point (CSP) in a domain with a trilinear atmosphere and two triangular reliefs is computed. The CSP parameters are: a frequency $f =$

300 MHz, with coordinates $x_s = x_{w0} + jk_0 W_0^2 / 2$, $z_s = 30$ m, with $x_{w0} = -50$ m and $W_0 = 5$ m.

We consider a refractive duct modelled by a trilinear atmosphere. The parameters for the refractivity index are $M_0 = 330$ M-units, $z_b = 100$ m, $z_t = 200$ m, with gradients $c_0 = 0.118$ M-units/m and $c_2 = -0.1$ M-units/m, see Fig. 1.

The relief is chosen as 2 small triangles of heights 100 m and 200 m. The impedance ground is of parameters $\epsilon_r = 20.0$ and $\sigma = 0.02$ S/m.

The domain is of size $x_{\max} = 100$ km in horizontal and $z_{\max} = 2048$ m in vertical. An apodization window of the vertical size is added. The grid size is 200 m in horizontal and 0.5 m in vertical.

The signal and propagator thresholds are $V_s = 10^{-3}$ and $V_m = 6 \times 10^{-5}$ respectively, so as to obtain an error of -30 dB at the last iteration using the formula given by Zhou *et al.* [4, 3]. The image layer for the ground condition is of size $0.1z_{\max}$.

The electric field using the local version of SSW is represented in Fig 2. The effects of both the atmosphere and the relief can be seen on that figure. The results is compared to DSSF for validation. The electric fields at the last range obtained with SSW and DSSF are shown in Fig. 3. The error due to the thresholds can be seen when the field magnitude is below -100 dBV/m. The maximum of the RMS error is -34 dB, which is less than the -30 dB expected value. The computation times and the memory usage of the methods are compared in Table I.

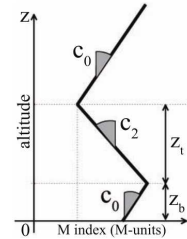


Figure 1. Trilinear model for refractivity.

Table 1. Time and memory storage needed for this scenario with each method.

	SSF	matrix SSW	local SSW
Initialization time (s)	0	2.9	1.0
Propagation time (s)	127	5.51	45
Total time (s)	127	8.41	46
memory for the propagator (Mb)	-	63	2.1

The local method works well for computing the propagation over a relief with a trilinear atmosphere. The computation efficiency is better than SSF but is outperformed by

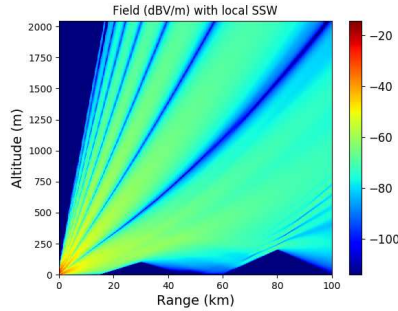


Figure 2. Electric field (dBV/m) in the vertical plane obtained by SSW.

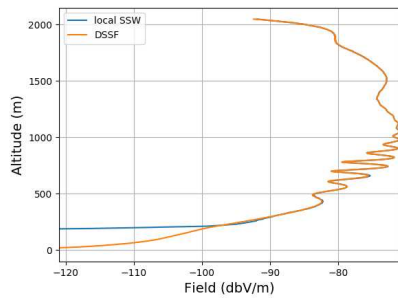


Figure 3. Electric field (dBV/m) at the maximum range obtained by SSF and SSW.

the matrix version of SSW. This is normal since the local propagation is performed only in Python, whereas the matrix product uses a library coded in C language. Finally, the memory usage is better with the library of operators than with the pre-computed matrix.

4 Conclusion

A local strategy for the free-space propagation with SSW has been introduced. This method is based on the wavelet transformation property.

From a recently proposed method (SSW), that is using a matrix-vector product for the free-space propagation, we have proposed a method based on local propagators. The former strategy using a matrix has the advantage of time efficiency. However, it is not efficient regarding the memory storage.

The novel strategy, based on a propagator library, is less efficient in terms of computation times as seen on the numerical tests, see Section 3, but is better in terms of memory resources used. Thus it is the best candidate for modelling 3D domains where the memory usage will be one of the main challenges.

5 Acknowledgements

The authors would like to thank the French Defense Agency (Direction Général de l'Armement, DGA) and the French Civil Aviation University (Ecole Nationale de l'Aviation Civile, ENAC) for the funding.

References

- [1] M. Levy, *Parabolic Equation Methods for Electromagnetic Wave Propagation*, **45**, IET, 2000.
- [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*, **44**, 1996, pp. 1592–1599, doi: 10.1109/8.546245.
- [3] H. Zhou, "Modeling the atmospheric propagation of electromagnetic waves in 2D and 3D using Fourier and wavelet transforms," PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2018.
- [4] H. Zhou, A. Chabory, and R. Douvenot, "Modeling wave propagation by a split-step wavelet method," *Journal of Computational Physics*, 2018, Under review.
- [5] J. R. Kuttler and R. Janaswamy, "Improved Fourier transform methods for solving the parabolic wave equation," *Radio Science*, **37**, no. 2, 2002. doi : 10.1029/2001RS002488
- [6] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic press, 1999.