The Adaptive Multi-Resolution Frequency-Domain ParFlow (MR-FDPF) Method for Indoor Radio Wave Propagation Simulation. Part I : Theory and Algorithms

Jean-Marie Gorce, Katia Jaffrès-Runser, Guillaume de la Roche

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Adaptive Multi-Resolution Frequency-Domain ParFlow (MR-FDPF) Method for Indoor Radio Wave Propagation Simulation. Part I : Theory and Algorithms.

Jean-Marie Gorce — Katia Jaffrès-Runser — Guillaume de la Roche

N° 5740

november 2005

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5740--FR+ENG

INRIA
RHÔNE-ALPES

The Adaptive Multi-Resolution Frequency-Domain ParFlow (MR-FDPF) Method for Indoor Radio Wave Propagation Simulation. Part I : Theory and Algorithms.

Jean-Marie Gorce*, Katia Jaffrès-Runser, Guillaume de la Roche

Thème COM — Systèmes communicants
Projet ARES

Rapport de recherche n° 5740 — november 2005 — 57 pages

Abstract: This report presents the theoretical background and new developments of the multi-resolution frequency domain ParFlow (MR-FDPF) approach for the calculus or radio propagation in Indoor environments for centimetric waves. This method has been developed to face the need of a best understanding of Indoor propagation and to help the WiFi network planning task. Indeed, the development of a wireless design tool is based firstly on a radio propagation engine to predict accurately the radio coverage of access points, with a limited computational load. Usual approaches in the literature are based on either empiric modeling, deducted from measurements, or geometrical optic formalism leading to ray-tracing. While the former suffers a lake of accuracy, the later needs a trade-off between accuracy and computational load, often difficult to assess. The approach proposed herein is based on a finite element approach. Once the problem developed in the frequency domain, the linear system thus obtained is solved in two steps: a pre-processing step which consists in an adaptive multi-resolution (multi-grid) pre-conditioning and a propagation step. The second step computes the coverage of a point source with an up-and-down propagation through the binary tree associated with the multi-resolution description. This approach solves exactly the linear system but with a strongly reduced computational complexity when compared to the time domain approach. For example, a full AP coverage at a macroscopic resolution and for an environment of 1000x600 pixels (i.e. $6000m^2$ at a $10cm$ resolution) lasts less than $200ms$.

Key-words: indoor radio propagation, numerical simulation, multi-resolution, frequency domain TLM, ParFlow

This project has been supported by the Rhône-Alpes council (technological transfert) and the french ministry of research (ACI Blanche 2002)

* jean-marie.gorce@insa-lyon.fr

Méthode multi-résolution adaptative des flux partiels dans le domaine fréquentiel (MR-FDPF) pour la simulation de la propagation des ondes radio en environnement indoor.
partie I : Théorie et algorithmes.

Résumé : Ce rapport présente les fondements théoriques et le développement de la méthode MR-FDPF pour le calcul de la propagation radio en environnement Indoor des ondes centimétriques. Cette méthode a été développée pour répondre au besoin croissant de mieux appréhender le déploiement des réseaux sans fil en environnement confiné, et plus particulièrement les réseaux de technologie WiFi. En effet, le développement d'un logiciel d'aide à la planification de réseaux sans fil repose en premier lieu sur l'intégration d'un moteur de propagation capable de prédire convenablement la couverture radio des points d'accès à déployer, tout en limitant la puissance de calcul nécessaire. Les approches de la littérature reposent soit sur une modélisation empirique, déduite des mesures, soit sur une modélisation basée sur l'optique géométrique et conduisant aux méthodes de lancer de rayons et assimilées. Alors que les méthodes empiriques souffrent d'un manque de précision, les méthodes géométriques nécessitent un compromis entre précision et temps de calcul parfois difficile. L'approche que nous proposons repose sur une formulation de type éléments finis. Une fois exprimé dans le domaine fréquentiel, le système linéaire ainsi obtenu est résolu en 2 étapes: une étape de pré-traitement qui consiste en un pré-conditionnement multi-résolution (ou multi-grilles) adaptatif et une étape de propagation proprement dite qui effectue le calcul de propagation en propageant une source ponctuelle par un parcours "up and Down" de l'arbre binaire associé à la structure multi-grilles. Cette approche permet de résoudre de façon exacte le système linéaire, en un temps de calcul très réduit par rapport à l'approche initiale développée dans le domaine temporel. A titre d'exemple, le calcul de couverture d'un AP à une résolution dite des blocs homogènes et pour un environnement de 1000 par 600 pixels (soit $6000m^2$ à une résolution de 10cm) nécessite de l'ordre de 200ms.

Mots-clés : propagation indoor des ondes RF, simulation numérique, multi-résolution, TLM, ParFlow

1 Introduction

During the last decade, successive advances in radiocommunication technologies led to a depth increase of the number of Indoor radio systems deployed in personal or professional environments. Bluetooth, WiFi (IEEE802.11b, .11a, .11n, ...) and UMTS are the more recent ones. The strong increase of the number of private local wireless LAN (wLAN) calls for more efficient design and planning tools. Most of installers currently used their experience and only few measurements to define the number of access point (AP's), their location and the associated emitted power: the result is often very different than expected. Wave propagation suffers indeed numerous reflections and diffractions in Indoor environments, making difficult to anticipate accurately the coverage area of each AP. For more than 10 years [1, 31, 43, 23, 53, 8] this problem has been deeply studied and today several tools are existing (see for instance [12, 52, 35, 54, 55]). The heart of these tools is obviously the wave propagation engine. As for micro-cells planning in urban areas, two kinds of propagation engines are widely used for Indoor pico-cells : empirical and deterministic approaches [1, 32].

- Empirical approaches [43, 34, 10, 20, 15] are based on statistical models of propagation. Wide databases are firstly built from exhaustive measurement campaigns and secondly empiric evolving laws are extracted as functions of meaningful parameters such as distance, frequency, base station height, and others. These methods allow to predict the average behavior of waves in typical environments but surely not accurate predictions in each room of the coverage area. These approaches are widely used for network design, because of their low computational load requirement.

- In the opposite, deterministic approaches have been initially developed to improve the propagation prediction in small dense urban areas [31, 23, 42], where global empirical models were not enough accurate. By analogy with the light propagation, these approaches are based on geometrical optic (G.O.) laws and efficient computation engines take their origin in the world of synthetic visualization and image rendering. Rays are launched attenuated and reflected depending on the obstacles they meet during their travel until the energy get under the sensibility threshold. Because these approaches are based on a deterministic model, they provide more accurate results than empiric approaches. They suffer however of a high computational load for wide open areas. Typically, these approaches are used in dense-urban areas and take into account few reflections per rays only.

The indoor context strongly differs from urban environment. Although having a smaller size, the propagation complexity increases because much more reflections and diffractions occur during the propagation between walls. One can believe that is the reason why empirical models failed to provide accurate Indoor propagation predictions. To improve the one-slope model (1SM), the multi-wall model (MWM) [1] was proposed to take wall effects into account but only on the direct path between the emitter and the receiver. Some improvements are possible, as for instance with the modified MWM proposed by Cheung [10] using a depth-varying attenuation slope coefficient. However, in these approaches multi-path cannot be formally considered because all of these models assume the only dependence of the path-loss with the emitter-receiver distance. As discussed in next paragraph, two recent approaches [20, 55] can be described as a way to overcome this limit.

Many works have been devoted to geometrical optic based techniques for Indoor in two (2D) or three dimensions (3D) since the beginning of nineties [31, 23, 42, 53, 44, 33, 36, 13, 14, 57, 54, 24, 56, 55, 47, 48, 49, 25, 26, 9, 3, 2]. Although these methods are very attractive, they find their limitation in the trade-off between computational load and accuracy. Indeed, the aim of accuracy calls for a fine angular sampling and the management of numerous reflections, while both are widely time-consuming. The computational load is indeed estimated in $O(k_r \cdot 2^r)$ where k_r is the number of launched rays and r the mean number of obstacles crossed by a ray [14, 49, 3]. Many works have been devoted either to improve the accuracy [49, 6, 5] or to reduce the computational complexity [24, 56, 3, 25]. Unfortunately both are mainly opposite. For computational trends, the number of reflection or diffraction effects is often limited to about 5 for each ray. This is really a problem in Indoor like environments because some peculiar effects such as the wave guiding in corridors cannot be taken into account. Wolfe et al [54, 24, 55] introduced the concept of the dominant path model in order to reduce drastically the complexity of ray-tracing. As other modern ray-tracers [14, 49, 25, 3], their approach is firstly based on a visibility graph computation aiming to reduce the search of intersection between rays and walls because this search yields the most computational load. Note that in their approach the concept of rooms having edges is used to build a room-oriented graph [24]. The dominant path model is then obtained by removing all paths except the one providing the most part of power, i.e. the dominant path. Although developed from the ray-tracing formulation, the dominant path model could be also seen as an improved empirical approach since only one path is considered. It should be noted however that the dominant path does not correspond exactly to a unique physical path but rather to a combination of multi-paths.

Hassan-Ali et al. have also proposed an approach in [20] introducing the geometrical optic theory inside an empirical approach. In this work, a room-based formalism was also used. Starting from the geometrical optic formulation, they proposed a probabilistic analysis of possible paths in each room. Then an ellipsoid having its centroids at the emitter and receiver locations is traced (like the Frenel ellipsoid). Finally the probability of multi-effects in each room having an intersection with this ellipsoid is used to compute a mean path-loss.

It appears clearly that most of recent works concerning Indoor propagation are firstly devoted to the computational efficiency of algorithms, both in two(2D) and three dimensions (3D). It is obvious that modern ray-tracing tools have the capability to predict accurately the Indoor propagation. In one side this is true only if the number of reflections is unlimited (or at least high) and if diffraction is considered. In the other side, since path loss predictions are intended to be used in a planning process, the computational load is expected to be low. Indeed, planning process requires often hundreds coverage estimations in order to find the best configuration of AP's. Furthermore, such a process should run on standard computers in a couple of minutes to be of interest for radio wLAN designers.

Compared to geometrical optics and empirical approaches, only few works tackled this problem with finite element approaches [51, 29, 41]. The reason is of course the high computational load usually achieved by these approaches. However, in 1998, Luthi et al. [11, 30] have proposed a new discrete approach, ParFlow, based on the Cellular Automata formalism. Although the theoretical background is different, the ParFlow algorithm is similar to the famous Transmission Line Matrix model [22, 28, 38] with a specific dispersion node. This approach is often used for circuit design and many works have been devoted to optimize these algorithms in the time-domain. Because of the lack of computational efficiency of such an approach (when compared to ray-tracing), Luthi [30] has proposed an implementation on parallel architectures, taking advantage of the Cellular Automata structure. These authors have proposed a dedicated implementation of their approach for Urban micro-cellular GSM simulations. The main advantage of this approach is that all propagation effects including reflection and diffraction are naturally taken into account but the required spatial resolution is in turns theoretically very high. Practically, these authors have proposed to simulate the 2D propagation using an intermediate frequency (not the true RF frequency) but rather a frequency adapted to the expected spatial resolution and depending on the size of the obstacles in the environment. The ParFlow method is presented and discussed in section 2.

At our knowledge, this method has never been applied in Indoor environments, albeit it has many advantages in this context. Firstly the computational load does not increase with diffractions and reflections. Secondly, all geometries of obstacles can be easily tackled with. However, the computational load of such an approach remains high. In order to reduce it and because we are dealing only with path loss predictions and absolutely not with time spreading and other wide-band effects, we propose in this report to formulate the ParFlow theory in the frequency domain leading to the Frequency Domain ParFlow (FDPF) approach described firstly in [19, 16]. This formulation is detailed in section 2.2. Then we propose an original way to solve this kind of problem with a multi-resolution approach in section 3 which has been presented in [17]. It is shown that our algorithm allows to reduce drastically the computational time. The main computational load is gathered into a pre-processing phase (before the placement of emitters), providing a very efficient propagation step. In section 4, few results are provided with accuracy and computational load estimations. It is shown that a full coverage on more than $10000m^2$ Indoor area is achieved in few hundreds milliseconds furthermore including all diffraction and reflection effects [39, 18]. Few accuracy results are herein provided but a strong validation on the basis of measurement campaigns has been conducted. The results have been presented in [40] and will be detailed in a next report.

2 ParFlow theory

In this part, the Frequency-Domain ParFlow (FDPF) method is presented starting from the time-domain formulation. First, bases of the ParFlow theory are provided. Besides, the multi-resolution approach in the frequency domain is described. It should be noticed that for the sake of simplicity, the problem is addressed in this report, in two dimensions (2D) for the orthogonal electric field only. A 2D scalar field is thus considered.

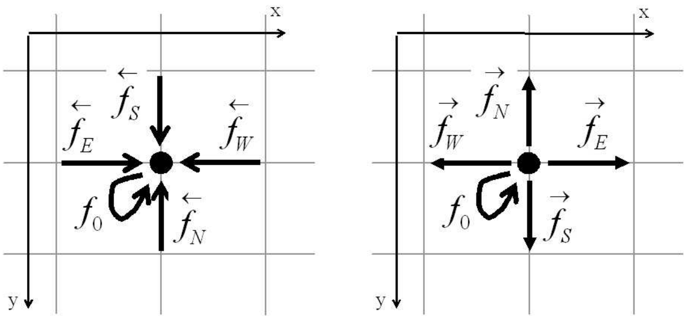


Figure 1: *5 inward flows (a) and 5 outward flows (b) are associated with each pixel.*

2.1 The time domain ParFlow formulation

2.1.1 Theoretical basics

Like other finite difference approaches, ParFlow (Partial Flows) is based on physics equations. The starting point is the Maxwell's wave equation in pure dielectric media given by

$$\delta_t^2 \Psi(\mathbf{r},t) - \left(\frac{c_0}{n}\right)^2 \cdot \nabla^2 \Psi(\mathbf{r},t) = 0 \qquad (1)$$

where $\Psi(\mathbf{r},t)$ is the electric field, c_0 the speed of light and n the refraction index of the medium.

In two dimensions (2D), $\Psi(\mathbf{r},t)$ is a scalar (corresponding to a vertical polarization). The equation used in the ParFlow approach is the first order approximation of the wave equation on a 2D regular grid leading to

$$\frac{\Psi(\mathbf{r},t-dt) - 2\cdot\Psi(\mathbf{r},t) + \Psi(\mathbf{r},t+dt)}{dt^2} = (\frac{c_0}{ndr})^2 \cdot \left[-4\cdot\Psi(\mathbf{r},t) + \sum_{i=1}^{4}\Psi(\mathbf{r}+\mathbf{dr_i},t)\right] \qquad (2)$$

The originality of the ParFlow approach when compared to other FDTD approaches stems from the flows associated with the scalar field. The electric field is indeed divided into 5 components: 4 directive flows bringing each one energy in a cardinal direction and an additional stationary flow [30]. These flows are driven by a local transition matrix derived from the discrete Maxwell's wave equation (2). The flows are shown in Figure 1 and are referred to as $\overleftarrow{f_d}$ and $\overrightarrow{f_d}$ respectively for inward and outward flows. The index $d, (d \in \{E, W, S, N\})$ indicates the flow direction (East, West, South, North). The stationary (or inner) flow, referred to as $\breve{f}_0$, reflects a dielectric media having a relative permittivity $\epsilon_r \neq 1$.

The ParFlow algorithm involves the bounds between flows over time. $\overleftarrow{F}(m,t)$ and $\overrightarrow{F}(m,t)$ referring respectively to inward and outward flow vectors are given by

$$\overleftarrow{F}(m,t) = \begin{pmatrix} \overleftarrow{f_E}(i,j,t) \\ \overleftarrow{f_W}(i,j,t) \\ \overleftarrow{f_S}(i,j,t) \\ \overleftarrow{f_N}(i,j,t) \\ \breve{f}_0(i,j,t) \end{pmatrix}; \quad \overrightarrow{F}(m,t) = \begin{pmatrix} \overrightarrow{f_E}(i,j,t) \\ \overrightarrow{f_W}(i,j,t) \\ \overrightarrow{f_S}(i,j,t) \\ \overrightarrow{f_N}(i,j,t) \\ \breve{f}_0(i,j,t) \end{pmatrix} \tag{3}$$

where m refers to pixel (i,j) with $m = j + i \cdot N_c$, N_c being the columns number. These flows are bound by

$$\overleftarrow{F}(m,t) = \mathcal{N}(\overrightarrow{F}(m,t)) = \begin{pmatrix} \overrightarrow{f_E}(i-1,j) \\ \overrightarrow{f_W}(i+1,j) \\ \overrightarrow{f_S}(i,j-1) \\ \overrightarrow{f_N}(i,j+1) \\ \breve{f}_0(i,j,t) \end{pmatrix} \tag{4}$$

The electrical field $\Psi(\mathbf{m},t)$ is related the local flows according to

$$\Psi(\mathbf{m},t) = \frac{1}{n_m^2} \cdot (\overleftarrow{f_E}(m,t) + \overleftarrow{f_W}(m,t) + \overleftarrow{f_S}(m,t) + \overleftarrow{f_N}(m,t) + Y_m \cdot \breve{f}_0(m,t)) \tag{5}$$

where $Y_m = 4n_m^2 - 4$, is the local admittance.

$\Psi(\mathbf{m},t)$ is the solution of (2) if the flows are evolving according to

$$\overrightarrow{F}(m,t) = \Sigma(m) \cdot \overleftarrow{F}(m,t-dt) \tag{6}$$

where $\Sigma(m)$ is the local scattering matrix defined by

$$\Sigma(m) = \frac{1}{2n_m^2} \cdot \begin{pmatrix} 1 & \alpha_m & 1 & 1 & Y_m \\ \alpha_m & 1 & 1 & 1 & Y_m \\ 1 & 1 & 1 & \alpha_m & Y_m \\ 1 & 1 & \alpha_m & 1 & Y_m \\ 1 & 1 & 1 & 1 & \beta_m \end{pmatrix} \tag{7}$$

with $\alpha_m = 1 - 2n_m^2$; $\beta m = 2n_m^2 - 4$.

According to the Luthi's proposal, the TLM matrix should be adapted to introduce the effect of absorbing materials leading to the following scattering matrix :

$$\Sigma_a(m) = a_m \cdot \Sigma(m) \tag{8}$$

where a_m is a normalized absorption coefficient.

2.1.2 Implementation

In [11], the ParFlow equation(6) is implemented in a cellular automata to update the instantaneous values of the flows. The instantaneous electric field is then computed with (5). Figure 2 illustrates the ability of the approach to take into account reflection and diffraction effects. In this example, the data are the source location and the indices of the materials (air and wall) only. It can be emphasized that these virtual flows are introduced at a microscopic level (pixel) but carry out the simulation of reflection and diffraction effects at a macroscopic level (see Figure 2). Choppard et al. showed that such a formalism takes into account implicitly losses, reflection, and diffraction effects. As pointed out by Luthi himself [30], this approach is strictly equivalent to the usual SCN (Super Condensed Node) TLM approach used for electronic circuits and antennas design [38], but derived in a broader context than radio-wave propagation.

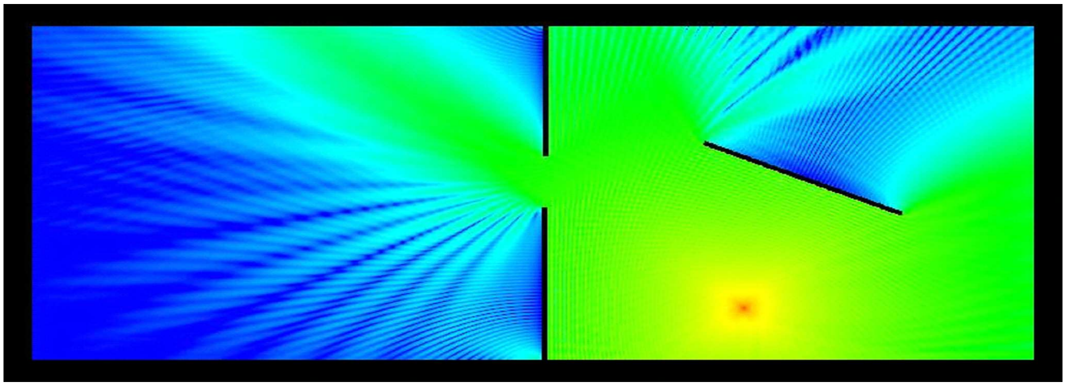


Figure 2: *this figure illustrates the hability of the method to simulate both reflection and diffraction effects. The environment size is 15m x 35m with a step of 5 cm. The wall thickness is 20cm, and the environment is surrounded by a simulated absorbing material.*

In this approach, the environment is described by the local parameters, n_m and a_m, according to the constitutive material (e.g. air, concrete, plaster, ...) associated with each pixel m. The environment is thus described by a full-space description contrarily to the description used in ray-tracing for which an object oriented description is used.

In the formulation of Luthi et al. [30] a radiating source is introduced by fixing the flows corresponding to the source cell. Equation (6) is then replaced by the following equation, for the site m_s of a source:

$$\overrightarrow{F}(m_s,t) = \overrightarrow{S}(m_s,t) \tag{9}$$

where $\overrightarrow{S}(m_s,t)$ is the vector of flows associated with the radiating source and is defined by the emitted pulse $s_0(t)$ as:

$$\overrightarrow{S}(m_s,t) = [s_0(t), s_0(t), s_0(t), s_0(t), 0]^t \tag{10}$$

We propose a different formulation for consistency with the full wave equation including a source. This furthermore helps the development of the frequency domain formulation. The field associated with a source site obeys to the extended wave equation:

$$\delta_t^2 \Psi(\mathbf{r},t) - (\frac{c_0}{n})^2 \cdot \delta^2 \Psi(\mathbf{r},t) = -\frac{1}{\epsilon} \cdot \delta_t i(\mathbf{r},t) \tag{11}$$

where $i(\mathbf{r},t)$ stands for a current source corresponding in 2D, to an infinite radiating cable located in r. A radiating source may thus be introduced in (6) as follows :

$$\overrightarrow{F}(m_s,t) = \Sigma(m_s) \cdot \overleftarrow{F}(m_s,t-dt) + \overrightarrow{S}(m_s,t) \tag{12}$$

It may be shown that the numerical source $s_0(t)$ is bound to the actual source by:

$$s_0(t) = \frac{dt^2}{4\epsilon} \cdot \delta_t i(\mathbf{r},t) \tag{13}$$

The main drawback of the Parflow method is its slowness. Indeed, it requires a high computational time to get the radio coverage over a large space such as a building floor. This high computational load is due to the high resolution required for such a method, having in 2D [30]:

$$\begin{aligned} &dr = c_0\sqrt{2} \cdot dt \\ &dr \ll \lambda \end{aligned} \tag{14}$$

Of course, the higher the spatial resolution is, the more accurate the results are but in turns the higher the computational load is. A limit has been found in [30] which states that the resolution should be at least 6 times lower than the wavelength. Besides, the number of iterations should be equal to several times the discrete length of the simulated area in order to take into account multi-path propagation and to avoid simulation artifacts. Thus the main drawback of this approach is on the one hand the high computational load, even when compensated for by the use of a parallel architecture [11]. But on the other hand, this approach is attractive for its efficiency to model natural propagation effects. Moreover, the algorithm allowing to implement this approach is very simple.

2.1.3 Algorithm

In the time-domain, the usual algorithm is summarized below:

Algorithm 2.1

```
Initialization :
|   $\forall m = \{i,j\} \neq s, \forall d \in \{E,W,S,N\}$,set $s_d(m) = 0$
|   $\forall d \in \{E,W,S,N\}$,set $s_d(m) = s_0$
Do (t=t+dt)
|   Compute outward flows :
    |    $\forall m,\ \overrightarrow{F}(m,t) = \Sigma(m) \cdot \overleftarrow{F}(m,t-dt) + \overrightarrow{S}(m,t)$
    |  Update inward flows:
    |    $\forall m,\ \overleftarrow{F}(m,t) = \mathcal{N}(\overrightarrow{F}(m,t))$
Until convergence
```

2.1.4 Global formulation

Let the flows be gathered into global vectors according to

$$\overleftarrow{\underline{F}}(t) = \begin{pmatrix} \overleftarrow{F}(0,t) \\ \overleftarrow{F}(1,t) \\ \vdots \\ \overleftarrow{F}(m,t) \\ \vdots \\ \overleftarrow{F}(M-1,t) \end{pmatrix} \quad \text{and} \quad \overrightarrow{\underline{F}}(t) = \begin{pmatrix} \overrightarrow{F}(0,t) \\ \overrightarrow{F}(1,t) \\ \vdots \\ \overrightarrow{F}(m,t) \\ \vdots \\ \overrightarrow{F}(M-1,t) \end{pmatrix} \tag{15}$$

with $M = N_y \cdot N_x$.
The source vector is given accordingly by

$$\overrightarrow{\underline{S}}(t) = \begin{pmatrix} \overrightarrow{S}(0,t) \\ \overrightarrow{S}(1,t) \\ \vdots \\ \overrightarrow{S}(m,t) \\ \vdots \\ \overrightarrow{S}(M-1,t) \end{pmatrix} \tag{16}$$

where $\overrightarrow{S}(m,t)$ is null if no source is active in $m = (i,j)$.

ParFlow is then written as a global linear system given by

$$\begin{aligned} \overrightarrow{\underline{F}}(t) &= \underline{\Sigma} \cdot \overleftarrow{\underline{F}}(t-dt) + \overrightarrow{\underline{S}}(t) \\ \overleftarrow{\underline{F}}(t) &= \underline{P} \cdot \overrightarrow{\underline{F}}(t) \end{aligned} \tag{17}$$

where $\underline{\Sigma}$ is the global scattering matrix and $\underline{P}$ the global permutation matrix.
Note that underlined vectors and matrices refer to the global system. $\underline{\Sigma}$ is a block-diagonal matrix, having blocks of size 5×5 pixels on its diagonal. Each block is defined by (8). $\underline{P}$ implements the equality between inward flows of a pixel and outward flows of its neighbors: it contains only row permutations. The propagation matrix $\underline{\Omega}$ is defined as the product between $\underline{\Sigma}$ and $\underline{P}$:

$$\underline{\Omega} = \underline{P} \cdot \underline{\Sigma} \tag{18}$$

$\underline{\Omega}$ is thus a $(N_x \cdot N_y \times N_x \cdot N_y)$ square matrix and is (5×5)-blocks pentadiagonal.

Combining (18) and (17) leads to:

$$\overleftarrow{\underline{F}}(t) = \underline{\Omega} \cdot \overleftarrow{\underline{F}}(t - dt) + \overleftarrow{\underline{S}}(t) \tag{19}$$

where $\overleftarrow{\underline{S}}(t) = \underline{P} \cdot \overrightarrow{\underline{S}}(t)$. Note that the multiplication of the source vector by $\underline{P}$ allows to put source coefficients defined as outward flows at their right place in the inward vector of flows.

This system implements exactly the ParFlow method working with inward flows only. However, it is obvious that the same vectorial space is associated with both $\overleftarrow{\underline{F}}(t)$ and $\overrightarrow{\underline{F}}(t)$ bounded by a permutation. Then, the global formulation could be established as well for outward flows.

2.2 Frequency domain formulation

In this section, we propose a new formulation of the ParFlow theory by setting the problem in the frequency domain. It should be noticed that the frequency domain (FD) TLM formulation has been already proposed, see for instance [7, 27, 21]. Let us however derive the frequency domain formulation starting from the time-domain ParFlow formalism, to ensure the strict equivalence between both formulations.

2.2.1 Passage to frequency domain

Equation (19) is transposed into the frequency domain using a Fourier transform. This leads to the Frequency Domain ParFlow equation (FDPF) :

$$\overleftarrow{\underline{F}}(\nu) = \underline{\Omega} \cdot e^{-j2\pi\nu dt} \overleftarrow{\underline{F}}(\nu) + \overleftarrow{\underline{S}}(\nu) \tag{20}$$

To solve this equation, the study is restricted to an harmonic mode and thus for a specific frequency ν_0, e.g. the carrier frequency. In this way the problem is reduced to a linear problem. To address the task of radio coverage prediction, such an harmonic steady-state study should be considered unappropriate because of time spreading. However this choice is argued by the following:

- Firstly, the time spreading of the radio channel in Indoor environments at 2.4GHz or 5GHz is small enough compared to the duration of WiFi pulses. The radio channel can be therefore assumed to be not dispersive. In this case, the harmonic response is enough to describe the radio channel.
- Secondly, even in the time-domain, Chopard et al. used themselves harmonic sources, in order to reduce the analysis complexity arising with a pulsed excitation.

This is an important aspect which emphasizes a limit of the FDPF model. Nevertheless, the multi-path characterization of the radio channel by the FDPF approach could however be done if the propagation would be estimated independently for several frequencies around the carrier. From these, the time response of the channel would be obtained by an inverse Fourier transform. Nevertheless, this is out of the scope of this report.

Under this assumption, equation (20) is solved only for a specific frequency ν_0 and leads to a traditional inverse problem given by:

$$(I_d - \underline{\Omega}_0) \cdot \overleftarrow{\underline{F}} = \overleftarrow{\underline{S}} \qquad \text{where} \quad \underline{\Omega}_0 = \underline{\Omega} \cdot e^{-j2\pi\nu dt} \tag{21}$$

In this equation and bellow, the variable (ν_0) has been removed for the sake of clarity. This equation may be obviously written as:

$$\overleftarrow{\underline{F}} = M^{-1} \cdot \overleftarrow{\underline{S}} \qquad \text{where} \quad M = I_d - \underline{\Omega}_0 \tag{22}$$

Even with a fast algorithm dedicated to sparse matrices, the direct inversion becomes rapidly unbearable as the environment size increases. For instance an environment of 1000 × 1000 pixels, such as a floor of $100m \times 100m$ at a resolution of $10cm$ would require the inversion of a sparse matrix of $(5\dot{}10^6)^2$ elements. Therefore, the development of a fast dedicated algorithm is very challenging. This is the aim of this work.

2.2.2 Removing the inner flows

The inner flow $\breve{f}_0$ has been introduced in the time domain to simulate different media. But in fact the inner flow is not needed in the frequency domain approach.

Let the inner flow be isolated from the rest of the flows according to

$$\overleftarrow{F}(m) = \begin{pmatrix} \overleftarrow{F_e}(m) \\ \breve{f}_0(m) \end{pmatrix} \quad \text{and} \quad \overrightarrow{F}(m) = \begin{pmatrix} \overrightarrow{F_e}(m) \\ \breve{f}_0(m) \end{pmatrix} \tag{23}$$

The source vector in given by

$$\overrightarrow{S}(m) = \begin{pmatrix} \overrightarrow{S}_e(m) \\ 0 \end{pmatrix} \tag{24}$$

These vectors are involved in the local frequency domain (FD) scattering equation derived from (12) and given by

$$\overrightarrow{F}(m) = \Sigma_f(m) \cdot \overleftarrow{F}(m) + \overrightarrow{S}(m) \tag{25}$$

where $\Sigma_f(m) = \Sigma(m) \cdot e^{-j2\pi\nu dt}$.

The flow $\breve{f}_0$ has been called *inner* flow because it does not participate to energy exchange between adjacent pixels. It only takes energy inside its node and relaxes it progressively. This flow is used to model a dielectric medium with a refraction index different from those of free-space and thus plays a fundamental role in the ParFlow approach. Reflection and diffraction effects follow directly from this flow.

Let the scattering matrix be divided into 4 blocks involving either inner (i) or exchange (e) flows according to

$$\Sigma_f(m) = \begin{pmatrix} \Sigma_{ee}(m) & \Sigma_{ei}(m) \\ \Sigma_{ie}(m) & \Sigma_{ii}(m) \end{pmatrix} \tag{26}$$

where the blocks are given by

$$\begin{array}{cc} \Sigma_{ee}(m) = \sigma_0 \cdot \begin{pmatrix} 1 & \alpha_m & 1 & 1 \\ \alpha_m & 1 & 1 & 1 \\ 1 & 1 & 1 & \alpha_m \\ 1 & 1 & \alpha_m & 1 \end{pmatrix} & \Sigma_{ei}(m) = \sigma_0 \cdot \begin{pmatrix} Y_m \\ Y_m \\ Y_m \\ Y_m \end{pmatrix} \\ \Sigma_{ie}(m) = \sigma_0 \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} & \Sigma_{ii}(m) = \sigma_0 \cdot \beta_m \end{array} \tag{27}$$

with $\sigma_0 = \frac{a_m}{2n_m^2} e^{-j2\pi\nu dt}$.
By introducing (51), the local scattering equation (25) now expends as

$$\begin{pmatrix} \overrightarrow{F_e}(m) \\ \breve{f}_0(m) \end{pmatrix} = \begin{pmatrix} \Sigma_{ee}(m) & \Sigma_{ei}(m) \\ \Sigma_{ie}(m) & \Sigma_{ii}(m) \end{pmatrix} \cdot \begin{pmatrix} \overleftarrow{F_e}(m) \\ \breve{f}_0(m) \end{pmatrix} + \begin{pmatrix} \overrightarrow{S}_{ex}(m) \\ 0 \end{pmatrix} \tag{28}$$

Solving this equation with respect to $\breve{f}_0(m)$ leads to

$$\breve{f}_0(m) = (Id - \Sigma_{ii}(m))^{-1} \cdot \left(\Sigma_{ie}(m) \cdot \overleftarrow{F_e}(m) \right) \tag{29}$$

The outward flows are then obtained according to

$$\overrightarrow{F_e}(m) = \Sigma_e(m) \cdot \overleftarrow{F_e}(m) + \overrightarrow{S}_e(m) \tag{30}$$

where

$$\Sigma_e(m) = \Sigma_{ee}(m) + \Sigma_{ei}(m) \cdot (Id - \Sigma_{ii}(m))^{-1} \cdot \Sigma_{ie}(m) \tag{31}$$

and expends as

$$\Sigma_{ex}(m) = \sigma_0 \cdot \begin{bmatrix} \sigma_1 & \sigma_2 & \sigma_1 & \sigma_1 \\ \sigma_2 & \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma_2 \\ \sigma_1 & \sigma_1 & \sigma_2 & \sigma_1 \end{bmatrix} \tag{32}$$

with

$$\sigma_1 = 1 + Y_m \cdot k_m \quad \text{and} \quad \sigma_2 = \alpha_m + Y_m \cdot k_m \tag{33}$$

and

$$k_m = \frac{\sigma_0}{1 - \sigma_0 \cdot \beta_m} \tag{34}$$

Finally, the electric field can be computed directly, according to:

$$\Psi(\mathbf{m}, \nu_0) = \frac{1 + k_m}{n_m^2} \cdot [\overleftarrow{f}_{E}(m) + \overleftarrow{f}_{W}(m) + \overleftarrow{f}_{S}(m) + \overleftarrow{f}_{N}(m)] \tag{35}$$

This new ParFlow equation can be still gathered into a global formalism involving exchange flows only. Let the global flow vectors be gathered into column vectors according to

$$\overleftarrow{\underline{F}}_e = \begin{pmatrix} \overleftarrow{F}_e(0) \\ \overleftarrow{F}_e(1) \\ \vdots \\ \overleftarrow{F}_e(M-1) \end{pmatrix} \tag{36}$$

The subscript e indicates that the corresponding vector or matrix refers to exchange flows only. The system is governed by the following equation

$$(I_d - \underline{\Omega}_e)\,\overleftarrow{\underline{F}}_e = \overleftarrow{\underline{S}}_e \tag{37}$$

with

$$\underline{\Omega}_e = \begin{pmatrix} [0] & \Omega^e_{W}(1) & & & \Omega^e_{N}(m) & \\ \Omega^e_{E}(0) & [0] & \Omega^e_{W}(2) & & & \ddots \\ & \Omega^e_{E}(1) & \ddots & \ddots & & \\ \Omega^e_{S}(0) & & \ddots & & & \\ & \ddots & & & & \end{pmatrix} \tag{38}$$

where the propagation sub-matrices are now:

$$\begin{aligned} \Omega^e_{E}(m) = \sigma_0 \cdot \begin{pmatrix} \sigma_1 & \sigma_2 & \sigma_1 & \sigma_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; \Omega^e_{W}(m) = \sigma_0 \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ \sigma_2 & \sigma_1 & \sigma_1 & \sigma_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; \\ \Omega^e_{S}(m) = \sigma_0 \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma_2 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; \Omega^e_{N}(m) = \sigma_0 \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \sigma_1 & \sigma_1 & \sigma_2 & \sigma_1 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; \end{aligned} \tag{39}$$

2.2.3 Iterative solving

Because the direct inversion is not applicable, an iterative scheme should be used to solve the linear problem. A conjugate gradient descent or other usual iterative technique could be used. Due to the sparse nature of the propagation matrix, efficient algorithms may be developed for this purpose. This approach has been proposed in [7, 27, 21] for the frequency Domain TLM (FD-TLM) algorithm devoted to circuit design.

Another approach exploits the structure of (21) in which M can be expressed as a matrix geometric series according to:

$$\overleftarrow{\underline{F}}_e = \sum_{k=0}^{\infty} (\underline{\Omega}_e)^k \cdot \overleftarrow{\underline{S}}_e = \overleftarrow{\underline{S}}_e + \underline{\Omega}_e \cdot \overleftarrow{\underline{S}}_e + (\underline{\Omega}_e)^2 \cdot \overleftarrow{\underline{S}}_e + \dots \tag{40}$$

In this approach, each term $(\underline{\Omega}_e)^k \cdot \overleftarrow{\underline{S}}_e$ adds to the final prediction the contribution of the source propagated k times. The algorithm is the following:

Algorithm 2.2

```
Initialization : $k = 0$; $\overleftarrow{\underline{F}}_e = \overleftarrow{\underline{F}}_e^{(k)} = \overleftarrow{\underline{S}}_e$
Do    *(until stability is reached)*
    k=k+1
    Compute : $\overleftarrow{\underline{F}}_e^{(k)} = \underline{\Omega}_e \cdot \overleftarrow{\underline{F}}_e^{(k-1)}$

    Accumulate : $\overleftarrow{\underline{F}}_e = \overleftarrow{\underline{F}}_e + \overleftarrow{\underline{F}}_e^{(k)}$

Until convergence
```

For best computational performances, the algorithm is obviously not global but rather grid-based exploiting the local scattering equation (30). The algorithm is given by

Algorithm 2.3

```
Initialization
    $\forall d \in \{E, W, S, N\}$,set $\overrightarrow{f_s^d} = s_0$
    $\forall m \neq s$, set $\overrightarrow{F_e}(m) = 0$
    $\forall m, \overrightarrow{F_e}(m) = \overrightarrow{F_e}(m)$
Do    *(until stability is reached)*
    Update inward flows:
        $\forall m$, $\overleftarrow{F_n}(m) = N(\overrightarrow{F_n}(m))$
    Compute outward flows
        $\forall m$, $\overrightarrow{F_n}(m) = \Sigma_e(m) \cdot \overleftarrow{F_n}(m)$
    Accumulate
        $\forall m$, $\overrightarrow{F_e}(m) = \overrightarrow{F_e}(m) + \overrightarrow{F_n}(m)$
Until convergence ($\forall m$, $\overrightarrow{F_n}(m) \ll \overrightarrow{F_e}(m)$)
```

Note that only very few differences hold when compared to the initial TDPF algorithm 2.1. For both, optimizations are possible exploiting advantageously the sparse nature of the propagation matrices. Thus, the outgoing flows could be computed only on pixels having *enough* incoming flow energy. It is furthermore possible with FDPF to keep and accumulate in memory non-propagated flows until getting a threshold over.

3 The multi-resolution (MR) approach

The multi-resolution (MR) principle is based on the FD previous model (now referred to as (0)-level model) exploiting recursively the concept of inner flows. In section 3.1 the node-level FDPF algorithm is derived. This algorithm is built considering multi-resolution nodes. A MR-node is a rectangular set of pixels and a flow vector is associated with each side. In this section, the scattering matrix associated with these flows is derived. Section 3.2.2 details how a MR-node of a given size is divided into two child nodes. A pyramidal structure is built recursively when the gathering is extended up to the larger MR-node, named the head-node. This head-node encompasses the whole environment. The regular multi-resolution FDPF algorithm (MR-FDPF) exploits efficiently this structure. In section 3.4, the adaptive MR-FDPF algorithm is described. This algorithm is named adaptive because the pyramidal structure is built to fit the environment. Starting from the head node, MR-nodes are recursively divided into children nodes. The dividing line of each node is chosen along discontinuities. Then, the minimization of computational load and memory consumption are discussed. Other interesting properties of the adaptive approach are also presented.

3.1 MR-node based approach

3.1.1 Formulation and notations

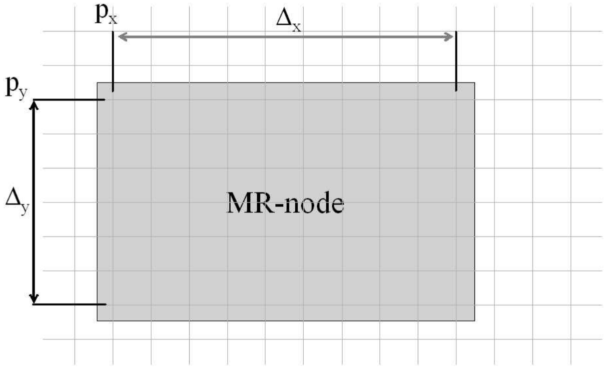


Figure 3: A MR-node is defined as a rectangular set of nodes.

The MR-FDPF algorithm is developed on the block concept derived in 2001 [19]. In this paper, we have defined a block as a set of aggregated pixels. Shlepnev in [45, 46] proposed a similar concept from the TLM formalism and called the brick concept. In his work, Shlepnev studied the wave propagation inside rectangular bricks using continuous formulation and allowing to derive scattering equations. Although probably less accurate, our discrete approach is mandatory to derive the multi-resolution framework. The term MR-node (multi-resolution node) refers to a rectangular set of pixels, defined by its size ($\overrightarrow{\Delta} = (\Delta_x, \Delta_y)$), and the position of its top-left pixel ($\overrightarrow{p} = (p_x, p_y)$) : $B(\overrightarrow{\Delta}, \overrightarrow{P})$ (see Fig. 3), shortened as b_k in the following.

For the sake of simplicity, the environment of N_{pix} pixels is firstly assumed square, having a size of $N_y = N_x = 2^Q; \quad Q \in \mathcal{N}$. The generalization to any rectangular shape will be provided later. As a starting point of the multi-resolution approach, the environment is regularly divided into MR-nodes of a given size, corresponding to a level ℓ resolution such as

$$\begin{aligned} \Delta_x(\ell) = \Delta_y(\ell) = 2^q \quad \text{if} \quad \ell = 2q \\ \Delta_x(\ell) = 2 \cdot \Delta_y(\ell) = 2^{q+1} \quad \text{if} \quad \ell = 2q+1 \end{aligned} \tag{41}$$

A (ℓ)-level node is called a (ℓ)-node and the environment is divided into K (ℓ)-nodes

$$\mathcal{B}_\ell = \left\{ b_k^\ell ; \forall k \in 1; \ldots ; K(\ell) \right\} \tag{42}$$

Exchange flows associated with each (ℓ)-node b_k^ℓ are now defined according to Fig.4a and b and leading to the eight directive flow vectors given by:

$$\begin{aligned}
&\overleftarrow{f}_{E}(b_k^\ell) = [\overleftarrow{f}_{E}(0,0), \overleftarrow{f}_{E}(1,0), \ldots, \overleftarrow{f}_{E}(N_y-1,0), \\
&\overleftarrow{f}_{W}(b_k^\ell) = [\overleftarrow{f}_{W}(0,N_x-1), \overleftarrow{f}_{W}(1,N_x-1), \ldots, \overleftarrow{f}_{W}(N_y-1,N_x-1), \\
&\overleftarrow{f}_{S}(b_k^\ell) = [\overleftarrow{f}_{S}(0,0), \overleftarrow{f}_{S}(0,1), \ldots, \overleftarrow{f}_{S}(0,N_x-1), \\
&\overleftarrow{f}_{N}(b_k^\ell) = [\overleftarrow{f}_{N}(N_y-1,0), \overleftarrow{f}_{N}(N_y-1,1), \ldots, \overleftarrow{f}_{N}(N_y-1,N_x-1), \\
\\
&\overrightarrow{f}_{E}(b_k^\ell) = [\overrightarrow{f}_{E}(0,N_x-1), \overrightarrow{f}_{E}(1,N_x-1), \ldots, \overrightarrow{f}_{E}(N_y-1,N_x-1), \\
&\overrightarrow{f}_{W}(b_k^\ell) = [\overrightarrow{f}_{W}(0,0), \overrightarrow{f}_{W}(1,0), \ldots, \overrightarrow{f}_{W}(N_y-1,0), \\
&\overrightarrow{f}_{S}(b_k^\ell) = [\overrightarrow{f}_{S}(N_y-1,0), \overrightarrow{f}_{S}(N_y-1,1), \ldots, \overrightarrow{f}_{S}(N_y-1,N_x-1), \\
&\overrightarrow{f}_{N}(b_k^\ell) = [\overrightarrow{f}_{N}(0,0), \overrightarrow{f}_{N}(0,1), \ldots, \overrightarrow{f}_{N}(0,N_x-1),
\end{aligned} \tag{43}$$

where the indices $(\cdot,\cdot)$ refer to the flow coordinates inside the MR-node.

The inward and outward *exchange* flow vectors are given by

$$\overleftarrow{F}_e(b_k^\ell) = \begin{pmatrix} \overleftarrow{f}_{E}(b_k^\ell) \\ \overleftarrow{f}_{W}(b_k^\ell) \\ \overleftarrow{f}_{S}(b_k^\ell) \\ \overleftarrow{f}_{N}(b_k^\ell) \end{pmatrix} ; \overrightarrow{F}_e(b_k^\ell) = \begin{pmatrix} \overrightarrow{f}_{E}(b_k^\ell) \\ \overrightarrow{f}_{W}(b_k^\ell) \\ \overrightarrow{f}_{S}(b_k^\ell) \\ \overrightarrow{f}_{N}(b_k^\ell) \end{pmatrix} \tag{44}$$

These vectors are bound by the local scattering equation

$$\overrightarrow{F}_e(b_k^\ell) = \Sigma_e(b_k^\ell) \cdot \overleftarrow{F}_e(b_k^\ell) + \overrightarrow{S}_e(b_k^\ell) \tag{45}$$

where $\Sigma_e(b_k^\ell)$ is the scattering matrix involving *exchange* flows only. This matrix can be divided into 16 blocks, each one involving flows from one side to one other. For instance, $\sigma^e_{WS}(k)$ relates west outward and south inward flows. Compared to the usual ParFlow scattering matrix, the scattering coefficients are replaced by scattering blocks.

3.1.2 Computing the scattering matrix

The computation of the scattering matrix associated with each MR-node starts from the usual formulation. To relate both formulations, let be defined inner flows belonging to MR-nodes. On the one hand, inward and outward flows are the flows located on the bounds of the MR-node as illustrated in Fig.4a and b. On the other hand, the inner flows are connecting two pixels of the same MR-node as illustrated in Fig.4c. The inner flow vector is given by

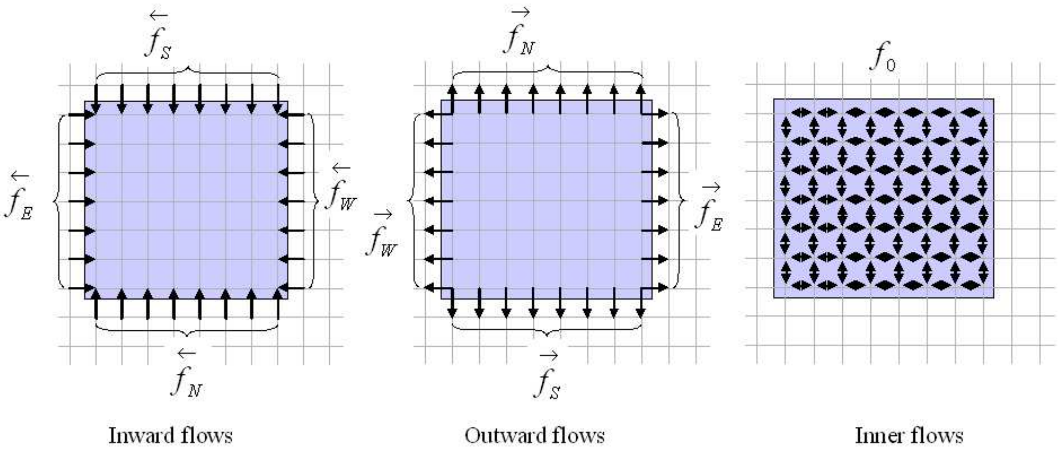


Figure 4: inward flows (a), outward flows (b) and inner flows (c) are associated with each MR-node.

$$\begin{aligned}\breve{f}_0(b_k^\ell) = [&\overrightarrow{f_E}(0,0), \overrightarrow{f_E}(0,1), \ldots, \overrightarrow{f_E}(N_y-1, N_x-2),\\ &\overrightarrow{f_W}(0,1), \overrightarrow{f_W}(0,2), \ldots, \overrightarrow{f_W}(N_y-1, N_x-1),\\ &\overrightarrow{f_S}(0,0), \overrightarrow{f_S}(0,1), \ldots, \overrightarrow{f_S}(N_y-2, N_x-1),\\ &\overrightarrow{f_N}(1,0), \overrightarrow{f_N}(1,1), \ldots, \overrightarrow{f_N}(N_y-1, N_x-1)]^t\end{aligned} \tag{46}$$

Gathering these flows with exchange flows provides the inward and outward flow vectors as

$$\overleftarrow{F}(b_k^\ell) = \begin{pmatrix} \overleftarrow{f_E}(b_k^\ell) \\ \overleftarrow{f_W}(b_k^\ell) \\ \overleftarrow{f_S}(b_k^\ell) \\ \overleftarrow{f_N}(b_k^\ell) \\ \breve{f}_0(b_k^\ell) \end{pmatrix} \quad ; \quad \overrightarrow{F}(b_k^\ell) = \begin{pmatrix} \overrightarrow{f_E}(b_k^\ell) \\ \overrightarrow{f_W}(b_k^\ell) \\ \overrightarrow{f_S}(b_k^\ell) \\ \overrightarrow{f_N}(b_k^\ell) \\ \breve{f}_0(b_k^\ell) \end{pmatrix} \tag{47}$$

These flows are bound by the local scattering equation:

$$\overrightarrow{F}(b_k^\ell) = \Sigma(b_k^\ell) \cdot \overleftarrow{F}(b_k^\ell) + \overrightarrow{S}(b_k^\ell) \tag{48}$$

where

$$\Sigma(b_k^\ell) = \begin{bmatrix} \sigma_{EE}(k) & \sigma_{EW}(k) & \sigma_{ES}(k) & \sigma_{EN}(k) & \sigma_{E0}(k) \\ \sigma_{WE}(k) & \sigma_{WW}(k) & \sigma_{WS}(k) & \sigma_{wN}(k) & \sigma_{W0}(k) \\ \sigma_{SE}(k) & \sigma_{SW}(k) & \sigma_{SS}(k) & \sigma_{SN}(k) & \sigma_{S0}(k) \\ \sigma_{NE}(k) & \sigma_{NW}(k) & \sigma_{NS}(k) & \sigma_{NN}(k) & \sigma_{N0}(k) \\ \sigma_{0E}(k) & \sigma_{0W}(k) & \sigma_{0S}(k) & \sigma_{0N}(k) & \sigma_{00}(k) \end{bmatrix} \tag{49}$$

where k herein and below stands for b_k^ℓ for the sake of clarity.

The only difference between (45) and (48) holds in the fact that the former involves only exchange flows, while the later involves also inner flows. Furthermore, in (49), the scattering submatrices derive directly from the usual formulation. Indeed, $\Sigma(b_k^\ell)$ is equals to the global ParFlow scattering matrix associated with the MR-node b_k^ℓ subspace after appropriate row and column permutations.

The calculus of the exchange flow scattering matrix involved in (45) is then obtained by solving (48) with respect to the inner flows. This is strictly equivalent to the derivation proposed in section 2.2.2. Let thus $\Sigma(b_k^\ell)$ be divided into 4 blocks as in (26) leading to

$$\Sigma(b_k^\ell) = \begin{pmatrix} \Sigma_{ee}(k) & \Sigma_{ei}(k) \\ \Sigma_{ie}(k) & \Sigma_{ii}(k) \end{pmatrix} \tag{50}$$

The blocks are given by

$$\Sigma_{ee}(k) = \begin{pmatrix} \sigma_{EE}(k) & \sigma_{EW}(k) & \sigma_{ES}(k) & \sigma_{EN}(k) \\ \sigma_{WE}(k) & \sigma_{WW}(k) & \sigma_{WS}(k) & \sigma_{wN}(k) \\ \sigma_{SE}(k) & \sigma_{SW}(k) & \sigma_{SS}(k) & \sigma_{SN}(k) \\ \sigma_{NE}(k) & \sigma_{NW}(k) & \sigma_{NS}(k) & \sigma_{NN}(k) \end{pmatrix} \quad \Sigma_{ei}(k) = \begin{pmatrix} \sigma_{E0}(k) \\ \sigma_{W0}(k) \\ \sigma_{S0}(k) \\ \sigma_{N0}(k) \end{pmatrix}$$
$$\Sigma_{ie}(k) = \begin{pmatrix} \sigma_{0E}(k) & \sigma_{0W}(k) & \sigma_{0S}(k) & \sigma_{0N}(k) \end{pmatrix} \qquad \Sigma_{ii}(k) = \sigma_{00}(k) \tag{51}$$

The local scattering equation (48) now expends as

$$\begin{pmatrix} \overrightarrow{F_e}(k) \\ \breve{f}_0(k) \end{pmatrix} = \begin{pmatrix} \Sigma_{ee}(k) & \Sigma_{ei}(k) \\ \Sigma_{ie}(k) & \Sigma_{ii}(k) \end{pmatrix} \cdot \begin{pmatrix} \overleftarrow{F_e}(k) \\ \breve{f}_0(k) \end{pmatrix} + \begin{pmatrix} \overrightarrow{S}_{0,e}(k) \\ \overrightarrow{S}_{0,i}(k) \end{pmatrix} \tag{52}$$

The exchange flow scattering matrix is obtained by setting the source vector to 0 and by solving (52) with respect to $\breve{f}_0(k)$ leading to

$$\Sigma_e(k) = \Sigma_{ee}(k) + \Sigma_{ei}(k) \cdot (Id - \Sigma_{ii}(k))^{-1} \cdot \Sigma_{ie}(k) \tag{53}$$

It can be emphasized that this calculus involves the computation of $(Id - \Sigma_{ii}(k))^{-1}$ which is nothing else that solving the usual ParFlow problem on an isolated sub-space corresponding to the MR-node k.

To solve the exact initial ParFlow equation any source located in any place inside a MR-node should be taken into account. For this purpose an equivalent source flow is computed from (52) by setting inward flows to 0, and leading to

$$\overrightarrow{S}_e(k) = \overrightarrow{S}_{0,e}(k) + \Sigma_{ei}(k) \cdot (Id - \Sigma_{ii}(k))^{-1} \cdot \overrightarrow{S}_{0,i}(k) \tag{54}$$

Then all required elements are obtained to solve the exchange flows based equation given by (45). This equation can be solved, by an usual iterative approach until the steady-state is reached. The computation the steady-state of inner flows involves the source contribution inside its MR-node on the one hand, and inward flows contribution in each MR-node independently on the other hand.

Both contributions are obtained by solving (52). The former by setting inward flows to 0 and the later by setting source flows to 0. This leads to:

$$\breve{f}_0(k) = (Id - \Sigma_{ii}(k))^{-1} \cdot \left(\overrightarrow{S}_{0,in}(k) + \Sigma_{ie}(k) \cdot \overleftarrow{F_e}(k)\right) \tag{55}$$

The propagation over the whole environment is thus computed in four steps as illustrated in Fig.5:

- **The Inner step**. It corresponds to the propagation of the elementary source inside its MR-node as if it was isolated from the rest of the world. It is mathematically represented in (55) by the first term. It can then be written:

  $$\breve{f}_0(k) = I(k) \cdot \overrightarrow{S}_0(k) \tag{56}$$

  where the local inner matrix $I(k)$ is given by

  $$I(k) = (Id - \Sigma_{ii}(k))^{-1} \tag{57}$$

  It applies node per node, on inner flows only. $\breve{f}_0(k)$ is definitely the solution of the ParFlow equation solved on the isolated source MR-node.

- **The Upward step**. Outward flows of the source MR-node are computed, providing the equivalent source node. The equivalent source is given by (54)

  $$\overrightarrow{S}_e(k) = \overrightarrow{S}_{0,e}(k) + U(k) \cdot \breve{f}_0(k) \tag{58}$$

  where the local upward matrix $U(k)$ is defined by

  $$U(k) = \Sigma_{ei}(k) \tag{59}$$

  This equation shows that the steady-state MR-node source is obtained as the sum of the source flows naturally located on the sides of the node with the outward flows resulting from the propagation of the other source flows in the source MR-node itself. Note that this step applies into source MR-nodes only.

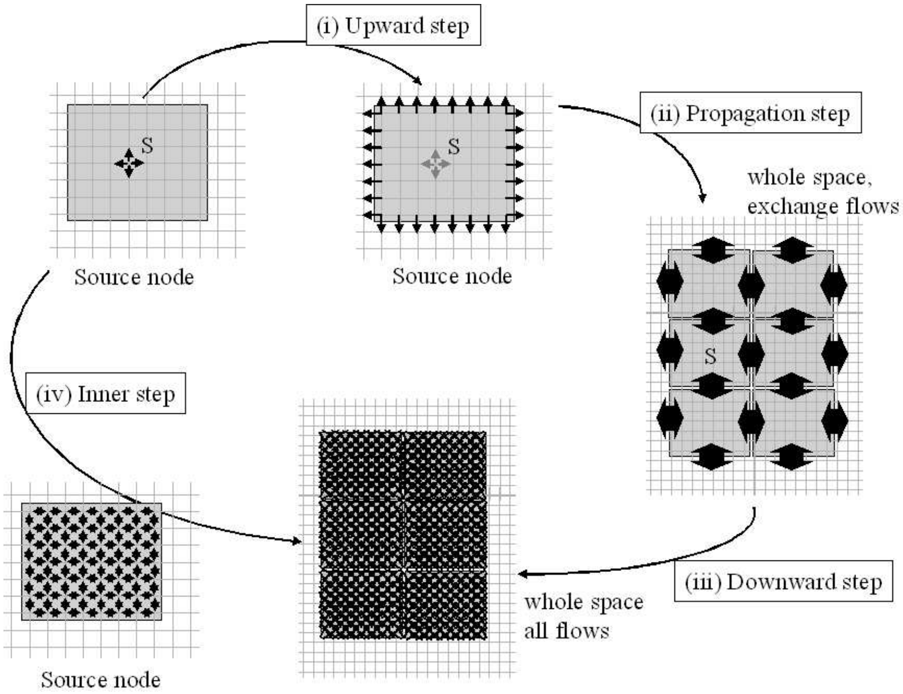


Figure 5: Illustration of the four computation steps for the block level algorithm. The computation starts on the top-left from the source pixel. the source node is computed during the upward step (i). Exchange flows are updated during the iterative propagation step (ii). Inner flows are estimated independently in each node during the downward step (iii). Simultaneously the source pixel contribution inside its node is computed during the inner step (iv).

- **The Iterative scattering step**. The MR-node based iterative algorithm is given by the algorithm 2.3 but applying on exchange flows only according to (45), with the scattering matrix computed in (53). This local scattering matrix involves exchange flows only. Since inner flows are naturally withdrawn during the iterative step, this approach can be computationally efficient. The number of variables is drastically reduced if MR-nodes are large enough.

- **The Downward step**. It corresponds to the computation of inner flows in each MR-node, when the steady-state is reached with exchange flows. It is mathematically given by the second term in (55) leading to

$$\breve{f}_0(k) = \begin{cases} I(k) \cdot D(k) \cdot \overleftarrow{F_{ex}}(k) & \text{if} \quad k \neq s \\ I(k) \cdot D(k) \cdot \overleftarrow{F_{ex}}(k) + \breve{f}_0(s) & \text{if} \quad k = s \end{cases} \tag{60}$$

  where $D(k)$ is the downward matrix given by

$$D(k) = \Sigma_{ie}(k) \tag{61}$$

  The incoming flows are propagated toward inner flows in each MR-node, independently.

A more formal proof of the equivalence between the MR-node based and the usual ParFlow algorithms is given in appendix 6.1, by the use of a global formalism.

3.1.3 MR-node level algorithm

The MR-node level algorithm is based on the algorithm 2.3 defined above, yielding:

Algorithm 3.1

```
Initialization :
    $\forall k,\ \overrightarrow{F_n}(k) = 0$ ; $\overrightarrow{F_e}(k) = 0$; $\breve{f}_0(k) = 0$
Inner step : Compute the steady-state inside the source MR-node
    Set $\breve{f}_0(s) = I(s) \cdot S_{in}(s)$
Upward step : Compute the source node outward flows
    Set $\overrightarrow{F_n}(s) = \overrightarrow{S}_{ex}(s) + U(s) \cdot \breve{f}_0(s)$
Iterative Scattering step : Compute exchange flows
Do
    Update inward flows: $\forall k,\ \overleftarrow{F_n}(k) = N(\overrightarrow{F_n}(k))$

    Compute temporary outward flows : $\forall k,\ \overrightarrow{F_n}(k) = \Sigma_{ex}(k) \cdot \overleftarrow{F_n}(k)$
    Accumulate : $\forall k,\ \overrightarrow{F_e}(k) = \overrightarrow{F_e}(k) + \overrightarrow{F_n}(k)$
Until convergence $(\forall k; ||\overrightarrow{F_n}(k)|| \ll ||\overrightarrow{F_e}(k))||$

**Downward step** : compute inner flows
    Compute : $\forall k,\ \breve{f}_0(k) = \breve{f}_0(k) + D(k) \cdot \overleftarrow{F_e}(k)$.
```

3.1.4 Complexity study of the MR-node level algorithm

The complexity is estimated for a direct implementation and with no further optimizations. The environment is assumed square for the sake of simplicity. Let us first estimate the complexity of the standard ParFlow algorithm. Since the time-domain algorithm is iterative, the exact computational load is difficult to assess and depends on the desired accuracy. A first estimation can be found by approximating the number of iterations to *few* times the environment size (i.e. $N_{it} = k \cdot N_x(\ell)$). k should be large enough to simulate multiple reflected waves. Of course, the value of k depends on the expected accuracy in the one hand, and on the loss factor of obstacles in the other hand. A second approximation valid only for large environments would consider that waves vanish before traveling the whole space. The number of iterations would therefore be estimated from the range of emitters. The first approximation applies better in the context of short-size Indoor environments. The reference computational load is then estimated as

$$\mathcal{C}(ref) \propto O\left(16 \cdot k \cdot N_x^3\right) \tag{62}$$

The complexity of the node based algorithm is divided into 4 parts:

- *Pre-processing*. The pre-processing phase consists in estimating the sub-matrices associated with each MR-node. This estimation requires for each node the inversion of a matrix of size $4 \cdot \Delta_x(\ell) \cdot \Delta_y(\ell) - 2 \cdot (\Delta_x(\ell) + \Delta_y(\ell))$. The computational load for one MR-node is thus in $O\left((4 \cdot \Delta_x(\ell) \cdot \Delta_y(\ell))^3\right)$. Then, for $\ell = 2q$, and if each node in the environment is different from each other, the computational load for the whole pre-processing is in $O\left(K_\ell \cdot (2 \cdot \Delta_x(\ell))^6\right)$ leading to:

  $$\mathcal{C}(prep) \propto O\left(64 \cdot N_x^2 \cdot \Delta_x(\ell)^4\right) \tag{63}$$

  The computational load increases as the fourth power of the MR-node size. The maximal size corresponds to a unique MR-node and then to the direct inversion (in $O\left(64 \cdot N_x^6\right)$).

- *Inner step.* The source is propagated inside its node. The computational load is:

  $$\mathcal{C}(down) \propto O\left(16 \cdot \Delta_x(\ell)^2\right) \tag{64}$$

  The computational load is not significant and proportional to to the MR-node surface.

- *Upward step.* The source MR-node is computed. This calculus is done for the source node only, with a computational load:
$$\mathcal{C}(up) \propto O\left(32 \cdot \Delta_x(\ell)\right) \tag{65}$$
The computational load is not significant and is proportional to the MR-node area.

- *Scattering step.* The computational load is the product between three terms : the number of iterations $N_{it} = k \cdot 2^{Q-\ell/2}$, the number of nodes $K_\ell = 2^{2Q-l}$ and the computational load of the scattering equation of a node in $O\left(4(\Delta_x(\ell) + \Delta_y(\ell))^2\right)$, leading to:
$$\mathcal{C}(scat) \propto O\left(k \cdot 16 \cdot N_x^3 \cdot \Delta_x(\ell)^{-1}\right) \tag{66}$$
In fact, the complexity evolves according to the number of iterations. The complexity is then inversely proportional to the MR-node size because the higher the size of the node is, the lower the number of iterations is. Note that the computational load for a node and the number of nodes counter-balance each other so that the computational load of one iteration is constant, whatever the MR-node size. But the profit of this approach holds in the fact that at each iteration, wave propagates over about $\Delta_x(\ell)$ pixels. Our experiments have shown that this MR-node level algorithm is really efficient, and reduces really the computational time by a factor proportional to the MR-node size. Furthermore, the memory management is strongly simplified. However, this improvement is counterbalanced by the time required for the pre-processing phase. Therefore, this approach is really efficient, if several sources have to be tested for the same environment. In this case, the pre-processing is done once for all possible source positions.

- *Downward step.* The inward flows of each node are down-propagated inside each one. The computational load is:
$$\mathcal{C}(down) \propto O\left(16 \cdot N_x^2 \cdot \Delta_x(\ell)\right) \tag{67}$$

For small MR-nodes, the main computational load is due to the iterative scattering step. In the opposite, for large MR-nodes, the preprocess and downward steps can make the problem untractable.

Pointing out the propagation phase and then putting the preprocessing step away, the trade-off can be estimated by minimizing the sum of $\mathcal{C}(scatt) + \mathcal{C}(down)$. This leads to an optimal block size about:
$$\Delta_x(\ell) = \sqrt{k \cdot N_x} \tag{68}$$
The computational load thus obtained when compared to the reference is given by :
$$\begin{aligned} \mathcal{C}(prep) &\propto 4k \cdot N_x \cdot \mathcal{C}(ref) \\ \mathcal{C}(scat) + \mathcal{C}(down) &\propto \frac{2}{\sqrt{k \cdot N_x}} \cdot \mathcal{C}ref \end{aligned} \tag{69}$$
It appears that the propagation phase is substantially reduced but the cost, i.e. the pre-processing phase, remains high.

3.2 The multi-resolution concept

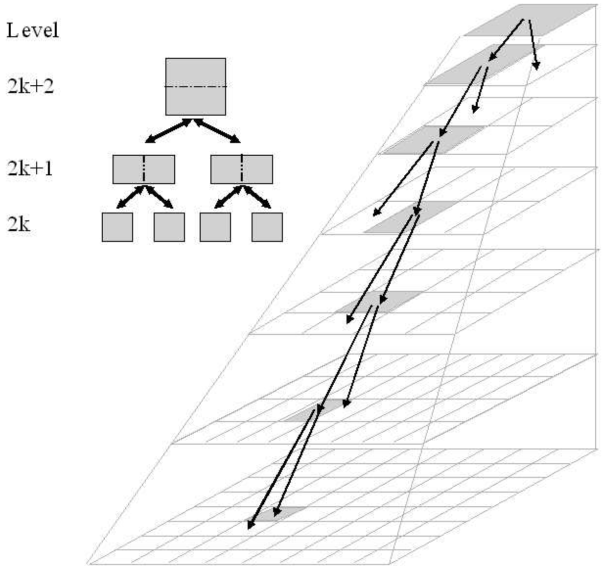


Figure 6: The regular multi-resolution approach is based on a regular binary tree for assembling nodes.

We have shown in the previous section that the ParFlow system can be expressed as K independent ParFlow systems of smaller size. The K systems are solved independently during the pre-processing, and then interact during the iterative scattering step. The same approach could be used to solve the problem in each sub-space, i.e. a sub-block decomposition could be used in each MR-node, and so on recursively, avoiding explicit matrix inversions.

It is in fact more efficient to build a multi-resolution structure by the use of a gathering process of MR-nodes. In this case, a regular binary-tree, or pyramid, is built as illustrated in Fig.6. In this structure, a (ℓ)-node (node of level $\ell+1$) contains two (ℓ)-nodes. The relationships between the levels $\ell+1$ and ℓ are now investigated.

3.2.1 Formulation at level (ℓ)

The notations of section 3.1 are still used. The local description of the ParFlow problem at level ℓ is summarized as follows:

- The exchange flows at level ℓ are given by (44)
- These flows are bound by the scattering matrix $\Sigma_e(b_k^\ell)$ according to (45).

3.2.2 Inter-level bounds

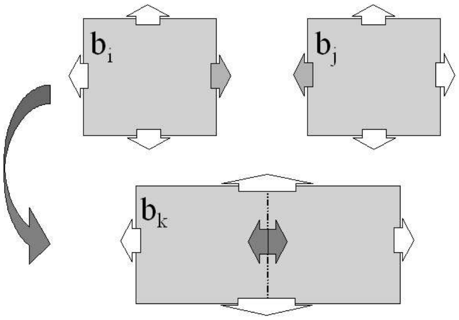


Figure 7: A node is obtained from two child nodes. The exchange flows between child nodes become inner flows (grey arrows).

Let b_i^ℓ and b_j^ℓ be two adjacent nodes. The scattering inside each one evolves according to (45) where $\Sigma_e(b_i^\ell)$ and $\Sigma_e(b_j^\ell)$ are assumed known.

To solve this (ℓ)-level equation at $(\ell+1)$-level, (ℓ)-nodes b_i^ℓ and b_j^ℓ are assumed gathered at level $(\ell+1)$ in the node $b_k^{\ell+1}$. The relationships between (ℓ) and $(\ell+1)$ flows depend on the direction of the gathering.

Horizontal gathering illustrated in Fig.7 corresponds to the case $\ell = 2q$ and $j = i+1$. In following equations, i, j and k respectively stand for b_i^ℓ, b_j^ℓ and $b_k^{\ell+1}$ for the sake of clarity.

The inward flow relationships are thus

$$\begin{aligned} &\overleftarrow{f_E}(k) = \overleftarrow{f_E}(i); \quad \overleftarrow{f_W}(k) = \overleftarrow{f_W}(j) \\ &\overleftarrow{f_S}(k) = \begin{pmatrix} \overleftarrow{f_S}(i) \\ \overleftarrow{f_S}(j) \end{pmatrix}; \quad \overleftarrow{f_N}(k) = \begin{pmatrix} \overleftarrow{f_N}(i) \\ \overleftarrow{f_N}(j) \end{pmatrix} \end{aligned} \tag{70}$$

while the outward flow relationships are

$$\begin{aligned} &\overrightarrow{f_E}(k) = \overrightarrow{f_E}(j); \quad \overrightarrow{f_W}(k) = \overrightarrow{f_W}(i) \\ &\overrightarrow{f_S}(k) = \begin{pmatrix} \overrightarrow{f_S}(i) \\ \overrightarrow{f_S}(j) \end{pmatrix}; \quad \overrightarrow{f_N}(k) = \begin{pmatrix} \overrightarrow{f_N}(i) \\ \overrightarrow{f_N}(j) \end{pmatrix} \end{aligned} \tag{71}$$

Let now the above defined inner flow $\breve{f}_0(k)$ be modified such as involving flows between child nodes only, according to:

$$\breve{f}_0(k) = \begin{pmatrix} \overleftarrow{f_W}(i) \\ \overleftarrow{f_E}(j) \end{pmatrix} = \begin{pmatrix} \overrightarrow{f_W}(j) \\ \overrightarrow{f_E}(i) \end{pmatrix} \tag{72}$$

The vertical gathering corresponds to the case $\ell = 2q+1$ and $j = i + N_x(\ell)$. The corresponding relationships are obtained by replacing subscripts E and W by S and N in (70)-(72).

These vectors are bound by the local scattering equation (48) at level $(\ell+1)$ involving exchange and inner flows. The $(\ell+1)$-level scattering matrix $\Sigma(k)$ is obtained from the *exchange* flows scattering matrices $\Sigma_e(i)$ and $\Sigma_e(j)$ by appropriate concatenation and interleaving considering (70)-(72) for horizontal gathering and equivalent equations for vertical gathering. For the horizontal gathering the relationship is:

$$\Sigma(k) = \begin{bmatrix} [0] & \sigma^e_{EW}(j) & [0] & \sigma^e_{ES}(j) & [0] & \sigma^e_{EN}(j) & [0] & \sigma^e_{EE}(j) \\ \sigma^e_{WE}(i) & [0] & \sigma^e_{WS}(i) & [0] & \sigma^e_{WN}(i) & [0] & \sigma^e_{WW}(i) & [0] \\ \sigma^e_{SE}(i) & [0] & \sigma^e_{SS}(i) & [0] & \sigma^e_{SN}(i) & [0] & \sigma^e_{SW}(i) & [0] \\ [0] & \sigma^e_{SW}(j) & [0] & \sigma^e_{SS}(j) & [0] & \sigma^e_{SN}(j) & [0] & \sigma^e_{SE}(j) \\ \sigma^e_{NE}(i) & [0] & \sigma^e_{NS}(i) & [0] & \sigma^e_{NN}(i) & [0] & \sigma^e_{NW}(i) & [0] \\ [0] & \sigma^e_{NW}(j) & [0] & \sigma^e_{NS}(j) & [0] & \sigma^e_{NN}(j) & [0] & \sigma^e_{NE}(j) \\ [0] & \sigma^e_{WW}(j) & [0] & \sigma^e_{WS}(j) & [0] & \sigma^e_{WN}(j) & [0] & \sigma^e_{WE}(j) \\ \sigma^e_{EE}(i) & [0] & \sigma^e_{ES}(i) & [0] & \sigma^e_{EN}(i) & [0] & \sigma^e_{EW}(i) & [0] \end{bmatrix} \tag{73}$$

and for the vertical gathering:

$$\Sigma(k) = \begin{bmatrix} \sigma^e_{EE}(i) & [0] & \sigma^e_{EW}(i) & [0] & \sigma^e_{ES}(i) & [0] & \sigma^e_{EN}(i) & [0] \\ [0] & \sigma^e_{EE}(j) & [0] & \sigma^e_{EW}(j) & [0] & \sigma^e_{EN}(j) & [0] & \sigma^e_{ES}(j) \\ \sigma^e_{WE}(i) & [0] & \sigma^e_{WW}(i) & [0] & \sigma^e_{WS}(i) & [0] & \sigma^e_{WN}(i) & [0] \\ [0] & \sigma^e_{WE}(j) & [0] & \sigma^e_{WW}(j) & [0] & \sigma^e_{WN}(j) & [0] & \sigma^e_{WS}(j) \\ [0] & \sigma^e_{SE}(j) & [0] & \sigma^e_{SW}(j) & [0] & \sigma^e_{SN}(j) & [0] & \sigma^e_{SS}(j) \\ \sigma^e_{NE}(i) & [0] & \sigma^e_{NW}(i) & [0] & \sigma^e_{NS}(i) & [0] & \sigma^e_{NN}(i) & [0] \\ [0] & \sigma^e_{NE}(j) & [0] & \sigma^e_{NW}(j) & [0] & \sigma^e_{NN}(j) & [0] & \sigma^e_{NS}(j) \\ \sigma^e_{SE}(i) & [0] & \sigma^e_{SW}(i) & [0] & \sigma^e_{SS}(i) & [0] & \sigma^e_{SN}(i) & [0] \end{bmatrix} \tag{74}$$

where i, j and k respectively stand for b^ℓ_i, b^ℓ_j and $b^{\ell+1}_k$ for the sake of clarity

The formal equivalence between (ℓ)- and $(\ell+1)$-level formulations is provided in appendix.

3.2.3 Recursive formalism

The formulation thus obtained at level $(\ell+1)$ is the same as used in section 3.1.2, but with different inner flows. However they can be put away in the same manner leading to the three fundamental equations (53)-(55). The only difference holds in the definition of the scattering matrix and its four sub-blocks derived from (73) and (74) for respectively horizontal and vertical gathering. The blocks for horizontal gathering are:

$$\Sigma_{ee}(k) = \begin{pmatrix} [0] & \sigma^e_{EW}(j) & [0] & \sigma^e_{ES}(j) & [0] & \sigma^e_{EN}(j) \\ \sigma^e_{WE}(i) & [0] & \sigma^e_{WS}(i) & [0] & \sigma^e_{WN}(i) & [0] \\ \sigma^e_{SE}(i) & [0] & \sigma^e_{SS}(i) & [0] & \sigma^e_{SN}(i) & [0] \\ [0] & \sigma^e_{SW}(j) & [0] & \sigma^e_{SS}(j) & [0] & \sigma^e_{SN}(j) \\ \sigma^e_{NE}(i) & [0] & \sigma^e_{NS}(i) & [0] & \sigma^e_{NN}(i) & [0] \\ [0] & \sigma^e_{NW}(j) & [0] & \sigma^e_{NS}(j) & [0] & \sigma^e_{NN}(j) \end{pmatrix} \quad \Sigma_{ei}(b^\ell_k) = \begin{pmatrix} [0] & \sigma^e_{EE}(j) \\ \sigma^e_{WW}(i) & [0] \\ \sigma^e_{SW}(i) & [0] \\ [0] & \sigma^e_{SE}(j) \\ \sigma^e_{NW}(i) & [0] \\ [0] & \sigma^e_{NE}(j) \end{pmatrix}$$

$$\Sigma_{ie}(k) = \begin{pmatrix} [0] & \sigma^e_{WW}(j) & [0] & \sigma^e_{WS}(j) & [0] & \sigma^e_{WN}(j) \\ \sigma^e_{EE}(i) & [0] & \sigma^e_{ES}(i) & [0] & \sigma^e_{EN}(i) & [0] \end{pmatrix} \quad \Sigma_{ii}(b^\ell_k) = \begin{pmatrix} [0] & \sigma^e_{WE}(j) \\ \sigma^e_{EW}(i) & [0] \end{pmatrix} \tag{75}$$

and for vertical gathering are:

$$\Sigma_{ee}(k) = \begin{pmatrix} \sigma^e_{EE}(i) & [0] & \sigma^e_{EW}(i) & [0] & \sigma^e_{ES}(i) & [0] \\ [0] & \sigma^e_{EE}(j) & [0] & \sigma^e_{EW}(j) & [0] & \sigma^e_{EN}(j) \\ \sigma^e_{WE}(i) & [0] & \sigma^e_{WW}(i) & [0] & \sigma^e_{WS}(i) & [0] \\ [0] & \sigma^e_{WE}(j) & [0] & \sigma^e_{WW}(j) & [0] & \sigma^e_{WN}(j) \\ [0] & \sigma^e_{SE}(j) & [0] & \sigma^e_{SW}(j) & [0] & \sigma^e_{SN}(j) \\ \sigma^e_{NE}(i) & [0] & \sigma^e_{NW}(i) & [0] & \sigma^e_{NS}(i) & [0] \end{pmatrix} \quad \Sigma_{ei}(b^\ell_k) = \begin{pmatrix} \sigma^e_{EN}(i) & [0] \\ [0] & \sigma^e_{ES}(j) \\ \sigma^e_{WN}(i) & [0] \\ [0] & \sigma^e_{WS}(j) \\ [0] & \sigma^e_{SS}(j) \\ \sigma^e_{NN}(i) & [0] \end{pmatrix}$$

$$\Sigma_{ie}(k) = \begin{pmatrix} [0] & \sigma^e_{NE}(j) & [0] & \sigma^e_{NW}(j) & [0] & \sigma^e_{NN}(j) \\ \sigma^e_{SE}(i) & [0] & \sigma^e_{SW}(i) & [0] & \sigma^e_{SS}(i) & [0] \end{pmatrix} \quad \Sigma_{ii}(b^\ell_k) = \begin{pmatrix} [0] & \sigma^e_{NS}(j) \\ \sigma^e_{SN}(i) & [0] \end{pmatrix} \tag{76}$$

The (ℓ)-level system can thus be solved involving a $(\ell + 1)$-level system in 3 phases:

- **Upward phase** : the exchange equivalent source at level $(\ell+1)$ is computed according to (54) from the (ℓ)-level source.
- **Iterative scattering phase** : the propagation equation (45) is solved iteratively according to algorithm 2.3, using $(\ell + 1)$-level exchange flows only.
- **Downward phase** : the inner flows are computed by (55) in each node independently. Leading to the knowledge of all exchange flows at level (ℓ).

Upward and Downward phases are illustrated in Fig.8.

3.3 The MR-FDPF recursion

The MR-FDPF is now obtained by applying recursively the inter-level relationships obtained above in two steps: the bottom-up and the top-down.

- The former is a recursion starting at level 0, aiming to compute at each level the equivalent source node flows from child source node. This step is referred to as the upward phase.
- The later is also a recursion but starting from the head-node. It aims to compute steady-state inner flows in each node having its inward flows, thus yielding steady-state inward flows of both child nodes. This is performed until the ground level is reached. This step is referred to as the downward phase.

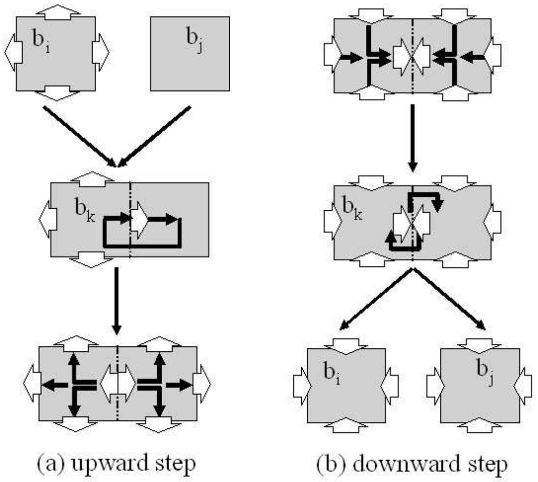


Figure 8: Illustration of the up-and-down propagation. Upward phase (a): i- The source node i is gathered with node j into node k. ii- The inner steady-state is computed. iii- The equivalent source is obtained. Downward phase (b): i- Inward flows are scattered into each child leading to inner flows. ii- The inner steady-state is computed. iii- the node k is now divided into child nodes having their inward flows computed.

3.3.1 Upward phase

The upward phase involves only source nodes. The (0)-level flows associated with the source node b_s^0 are firstly initialized as those of the pixel source. Then the (ℓ)-level outward flows associated with source node (k=s) are computed recursively from (54), in two steps:

- The first step aiming to compute the steady-state inner flows is given by:

$$\breve{f}_0(s) = I(s) \cdot \overrightarrow{S}_0(s) \tag{77}$$

  where

$$I(s) = (Id - \Sigma_{ii}(s))^{-1} \tag{78}$$

  is still the *inner matrix.*
  $\overrightarrow{S}_0(s)$ is obtained depending on which child is the source. For an horizontal gathering, one have

$$\overrightarrow{S}_0(s) = \begin{pmatrix} \overrightarrow{f_E}(i) \\ 0 \end{pmatrix} \quad \text{or} \quad \overrightarrow{S}_0(s) = \begin{pmatrix} 0 \\ \overrightarrow{f_W}(j) \end{pmatrix} \tag{79}$$

  For a vertical gathering E and W subscripts are still replaced by S and N.

- The second step uses these steady-state inner flows to compute the new source vector according to

$$\overrightarrow{S}_e(s) = \overrightarrow{S}_{ex}(s) + U(s) \cdot \breve{f}_0(s) \tag{80}$$

where $\overrightarrow{S}_{ex}(s)$ is initialized with the $(\ell-1)$ outward flows of the source child node (either i or j), according to (71).

The *upward matrix* $U(s)$ associated with node s is given by

$$U(s) = \Sigma_{ei}(s) \tag{81}$$

The recursion ends when the head-node is reached, having its inner flows computed. It should be noticed that the inner steady-state flow vectors $\breve{f}_0(b_s^\ell)$ associated with embedded source nodes are stored.

3.3.2 Downward phase

The downward phase starts at level L down to level 0. At level L, inward flows $\overleftarrow{F_e}(b_0^L)$ (k=0 because level L contains one node only) are set according to the boundary conditions. Indeed, $\overleftarrow{F_e}(b_0^L)$ involves the flows bringing energy from outside. To avoid boundary artefact and spurious reflections, the environment is surrounded in a virtual absorbing material allowing to reduce the amplitude of boundary outer flows. Since b_0^L contains the absorbing layer, inward flows are set to 0. The inner flows are thus unchanged and have to be propagated toward child nodes. This applies down to level 0, over all the nodes. This calculus is done in each node independently, and consists in the computation of the inner flows as function of inward flows according to (29)

$$\breve{f}_0(k) = \begin{cases} I(k)\cdot D(k)\cdot \overleftarrow{F_e}(k) & \text{for } k \neq s \\ I(k)\cdot D(k)\cdot \overleftarrow{F_e}(k) + \breve{f}_0(s) & for\text{k = s} \end{cases} \tag{82}$$

where the *downward matrix* $D(k)$ is given by

$$D(k) = \Sigma_{ie}(k) \tag{83}$$

At the end, the ParFlow linear system is solved exactly, with not any approximation, except for those due to the numerical accuracy.

3.3.3 Preprocessing phase

In this approach, three propagation matrices are associated with each node (Upward, Downward and Inner matrices). Because their computation does'nt require the knowledge of the source, this computation can be prepared in a preprocessing phase. This allows the overall computational load to be reduced when many sources have to be computed. Computing the matrices associated with a given (ℓ)-node requires the knowledge of the scattering matrix of each of its child nodes. Indeed, the Upward and Downward matrices are directly derived from the exchange scattering matrix of child nodes according to (81) and (83) while considering (75) or (76). There is therefore no significant computational load to obtain these matrices.

The first significant part is due to the calculus of the exchange flows scattering matrix , because required by its father node to compute its own propagation matrices. The exchange flows scattering matrix is given by (53) as a function of the scattering matrix (including inner flows). This later scattering matrix is itself obtained from the exchange scattering matrix of its child nodes. Starting from the ground level having the scattering matrix of each (0)-node, all exchange scattering matrices can be recursively computed.

The second part of the computational load is due to the calculus of $I(k)$ according to (78). This matrix can be efficiently computed, for the horizontal gathering under the form

$$I(k) = \begin{pmatrix} I_{EE}(k) & I_{EW}(k) \\ I_{WE}(k) & I_{WW}(k) \end{pmatrix} \tag{84}$$

with

$$\begin{aligned} I_{EE}(k) &= (Id - \sigma^e_{EW}(i) \cdot \sigma^e_{WE}(j))^{-1} \\ I_{WW}(k) &= (Id - \sigma^e_{WE}(j) \cdot \sigma^e_{EW}(i))^{-1} \\ I_{EW}(k) &= \sigma^e_{EW}(i) \cdot I_{WW}(k) \\ I_{WE}(k) &= \sigma^e_{WE}(j) \cdot I_{EE}(k) \end{aligned} \tag{85}$$

For vertical gathering subscripts E and W are still replaced by S and N.

An interesting property should be now emphasized. It is obvious to show that the scattering matrix issued from the ParFlow theory preserves the reciprocity theorem of wave propagation. This theorem thus applies to flows in steady-state. In other words, the scattering matrix exhibits symmetrical properties according to

$$\sigma_{XY}(b^\ell_k) = \sigma_{\overline{YX}}(b^\ell_k)^t \tag{86}$$

where $\overline{X}$ refers to the opposite direction, i.e. $\overline{E} = W$, $\overline{W} = E$, $\overline{S} = N$, $\overline{N} = S$.

This allows to reduce substantially both the memory consumption and the computational load for scattering matrix computations. The computational load dedicated to inner matrix is also reduced because it follows

$$\begin{array}{lc} I_{EE}(b^\ell_k) = I_{WW}(b^\ell_k) & \text{for an horizontal gathering} \\ I_{SS}(b^\ell_k) = I_{NN}(b^\ell_k) & \text{for a vertical gathering} \end{array} \tag{87}$$

The MR-FDPF algorithm is finally obtained according to:

Algorithm 3.2

```
Preprocessing:
  | For $\ell = 1$ to $L-1$, Do
  |     *Compute* $\forall k$; $\Sigma(b^\ell_k)$, $I(b^\ell_k)$, $U(b^\ell_k)$, $D(b^\ell_k)$
Initialization:
  | *Set* $\forall \ell$ ; $\forall k$; $\breve{f}_0(b^\ell_k) = 0$
Upward phase:
    For $\ell = 1$ to $L-1$, Do
        Update source flows
          $S_e(b_s^{\ell-1}) \Rightarrow S_0(b_s^\ell)$ ; $S_{ex}(b_s^\ell)$
        Compute inner flows
          $\breve{f}_0(b_s^\ell) = I(b_s^\ell) \cdot S_0(b_s^\ell)$
        Compute source flows
          $S_e(b_s^\ell) = S_{ex}(b_s^\ell) + U(b_s^\ell) \cdot \breve{f}_0(b_s^\ell)$
Downward phase:
    For $\ell = L-1$ down to 0, Do
        For $k = 0$ to $K_\ell - 1$, Do
          Compute
            $\breve{f}_0(b_s^\ell) = \breve{f}_0(b_s^\ell) + I(b_s^\ell) \cdot D(b_s^\ell) \cdot \overleftarrow{F_e}(b_s^\ell)$
          Update child inward flows
            $\overleftarrow{F}(b_k^\ell) \Rightarrow \overleftarrow{F_e}(b_i^{\ell-1})$ ; $\overleftarrow{F_e}(b_j^{\ell-1})$
```

3.3.4 Complexity study

It should be firstly noted that the preprocessing step is the one needing the most computational load used to process local scattering matrices. Fortunately, the preprocessing is definitely not done for each source but once at all. The upward step is very fast since it concerns only source nodes but the downward step costs more because computed for each node of each level. For the sake of simplicity a square environment is only considered in this complexity study, such as $N_y = N_x = 2^Q$.

Preprocessing loads The main computational load is devoted to the calculus of $I(b_k^\ell)$ according to (78) and of $\Sigma(b_k^\ell)$ according to (31), and equals $19 \cdot 2^{3q}$ for $\ell = 2q$ and to $27 \cdot 2^{3q}$ for $\ell = 2q+1$. Taking the number of nodes into account, the computational load of each level is given by

$$\mathcal{C}(prep, \ell) \propto \begin{cases} O\left(19 \cdot N_x^2 \cdot 2^q\right) & \text{if } \ell = 2q \\ O\left(27 \cdot N_x^2 \cdot 2^{q-1}\right) & \text{if } \ell = 2q+1 \end{cases} \tag{88}$$

The whole pre-processing time is thus given by

$$\mathcal{C}(prep) \propto O\left(52 \cdot N_x^3\right) \tag{89}$$

The memory needs can also be estimated for the storage of the scattering and inner matrices. The memory consumption for a (ℓ)-node is found to be

$$\mathcal{M}(b_k^\ell) = 13 \cdot 2^\ell \cdot mem \tag{90}$$

where *mem* is the memory need to store a complex variable (e.g. 8 bytes for single float).

It follows that the memory need associated with each level is a constant, given by

$$\mathcal{M}(\ell) = 13 \cdot N_x^2 \tag{91}$$

The total memory need then equals

$$\mathcal{M}(prep) = 26 \cdot log_2(N_x) \cdot N_x^2 \tag{92}$$

Both estimations are in fact upper-bounds. Indeed, they are obtained when considering that each scattering matrix is computed independently for each MR-node. as discussed above, the scattering matrix is instead computed for each brick only. The whole free-space is the most favorable case because all pixels are identical and thus at each (ℓ)-level, all (ℓ)-nodes are identical. It follows that only one scattering matrix is computed at each level. The preprocessing loads further decreases, leading to the lower bounds

$$\begin{aligned} \mathcal{C}(prep) &\propto O\left(c_p \cdot N_x^3\right) \\ \mathcal{M}(prep) &\propto O\left(m_p \cdot N_x^2\right) \end{aligned} \tag{93}$$

with $34 < c_p < 52$ and $26 < m_p < 26 \cdot log_2(N_x)$. It can be pointed out that the reuse of scattering matrices allows to reduce much more efficiently the memory needs than the computational load. Indeed, the head-node (last level) consumes itself about 50% of the computational load.

Upward loads Since the upward phase concerns only source nodes, the associated computational load is not significant. The computational load associated with both the inner and the upward steps equals

$$\mathcal{C}(up, \ell) \propto \begin{cases} O\left(6 \cdot 2^\ell\right) & \text{if } \ell = 2q \\ O\left(4 \cdot 2^\ell\right) & \text{if } \ell = 2q+1 \end{cases} \tag{94}$$

Note that for the last level, only the inner flows are computed, leading to a whole cost

$$\mathcal{C}(up) \propto O\left(3 \cdot N_x^2\right) \tag{95}$$

where more than 50% is consumed for the two upper levels.

Downward loads Inward flows are down-propagated inside each MR-node. The computational load associated with one MR-node is $4 \cdot 2^\ell$ if $\ell = 2q+1$ and $6 \cdot 2^\ell$ if $\ell = 2q$. The whole computational load associated with each level is then constant, given by

$$\mathcal{C}(down, \ell) \propto \begin{cases} O\left(6 \cdot N_x^2\right) & \text{if } \ell = 2q \\ O\left(4 \cdot N_x^2\right) & \text{if } \ell = 2q+1 \end{cases} \tag{96}$$

The whole computational load is

$$\mathcal{C}(down) \propto O\left(10 \cdot log_2(N_x) \cdot N_x^2\right) \tag{97}$$

Standard algorithms loads Such a result should be compared with the complexity of the more usual ParFlow time-domain algorithm. Since the time-domain algorithm is iterative, the exact computational load is difficult to assess and depends on the desired accuracy. A first estimation can be found by approximating the number of iterations to *few* times the environment size (i.e. $N_{it} = k \cdot N_x(\ell)$). k should be large enough to simulate multiple reflected waves. Of course, the value of k depends on the expected accuracy in the one hand, and on the loss factor of obstacles in the other hand. A second approximation valid only for large environments would consider that waves vanish before traveling the whole space. The number of iterations would therefore be estimated from the range of emitters. The first approximation applies better in the context of short-size Indoor environments. The reference computational load is then estimated as

$$\mathcal{C}(ref) \propto O\left(16 \cdot k \cdot N_x^3\right) \tag{98}$$

It can be concluded that MR-FDPF outperforms the standard algorithm. Concerning the preprocessing, the computational load is proportional to $O(N_x^3)$ and thus is comparable to the standard algorithm computational load for one source. But the computational load of a source propagation is much more lower in our approach, being in $O(\log_2(N_x) \cdot N_x^2)$. It means that after preprocessing, a full coverage is obtained with a computational load equals to few TDPF iterations only.

Let us now compare the complexity of the MR-FDPF approach to the one of the MWM approach. The complexity of the MWM is roughly proportional to the number of receiving points. To estimate the load of each point to point link budget, two computational phases can be distinguished: the wall intersection search and the path-loss computation. The former is often said more consuming than the later but in fact depends on the number of walls to manage. To obtain the full resolution, the complexity of the second phase is obviously obtained as $\mathcal{C}(MWM) \propto O\left(k_m \cdot N_x^2\right)$ where k_m is the mean cost of a path-loss computation. Thus, a first insight exhibits that the complexity of the MR-FDPF is comparable to the complexity of a very simple MWM approach.

A comparison with ray-tracing based algorithms would also be interesting but difficult to assess. Indeed ray-tracing is based on a vectorial formalism while MR-FDPF is full-space based. The complexity of ray-tracing is exponentially bound to the number of rays and reflections and not on the environment size. With only a few receiving points and few reflections, ray-tracing is probably the faster approach, but MR-FDPF certainly outperforms ray-tracing for a full resolution study and when the environment contains many obstacles, because ray-tracing is known to be much more time consuming than the MWM approach in this case [1, 32, 56].

3.4 Adaptive tree

3.4.1 Trends of an irregular tree

A regular binary tree is the easiest to be built. It is however neither the only way, nor the most efficient.

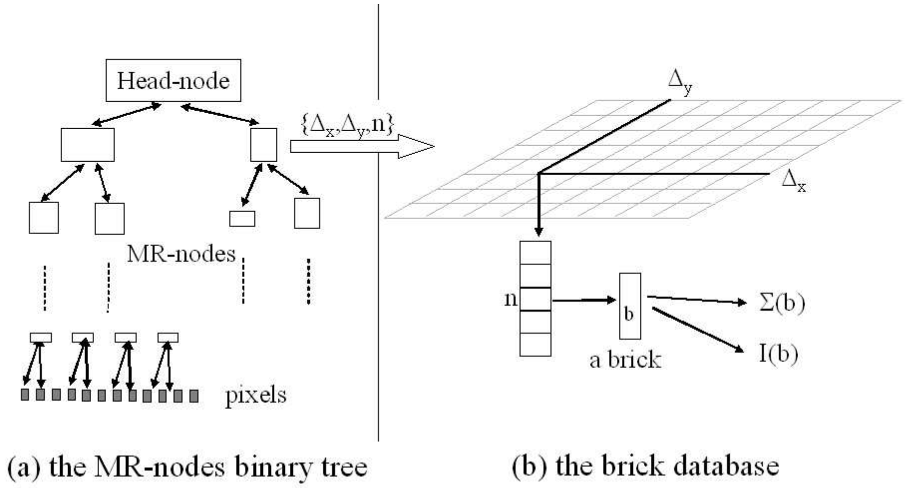


Figure 9: The data structure contains two main objects : the tree of MR-nodes and the brick types.

The efficiency is considered with respect to four criteria:

- The computational time of the pre-processing step, dedicated to the computation of $\Sigma()$ and $I()$ associated with each MR-node.
- The memory needed to store both matrices.
- The computational time needed to compute the coverage of a source over the whole environment at the pixel resolution.
- The computational time, needed to compute the coverage of a source but at a rough resolution.

This rough resolution is introduced to further reduce the computational time by ending the downward propagation at MR-nodes corresponding to free-space homogeneous nodes. A free-space homogeneous MR-node refers to a node containing only air cells. For these nodes, the mean received power is evaluated directly from the inward flows according to

$$P(b_i) = \frac{\|\overleftarrow{F}(b_i)\|^2}{2 \cdot (N_x + N_y)} \qquad (99)$$

This reduces the computational load because the downward propagation is stopped in each branch as soon as an homogeneous node is reached. The resolution thus obtained is therefore adaptive, depending on the desired accuracy. Such a proposal suits well to the problem of coverage predictions: in wide free space areas, the mean received power is computed over wide MR-nodes, while in dense areas (numerous obstacles) the received power is computed over smaller MR-nodes. This further allows to reduce the variability of predictions due to short-time fading if MR-nodes over which the mean power is computed are larger than a wavelength. This definitely calls for a dividing procedure privileging large homogeneous nodes.

Concerning the preprocessing step, it appears clearly that the main computational load and memory consumption are due to the computation of scattering matrices. Note that specific computational and memory loads are associated with each MR-node and the larger the MR-node is, the higher the loads are. The case where two MR-nodes are identical needs to be further investigated. Two MR-nodes are said identical if they have the same size and the same child nodes. In this case, their matrices are also identical and a reference model can be computed and stored for both. It is therefore advantageous to try to build a binary tree having many identical MR-nodes. In order to exploit nodes similarity, the brick concept is defined, referring to a type of MR-nodes. A MR-node has thus a father, two children, flow vectors and a reference to its type : the brick. The brick refers itself to his two child bricks(to compute the scattering matrices) and to its scattering matrix. A peculiar data structure is used to optimize both the memory consumption and the computational load. This structure is illustrated in Fig.9. The MR-nodes tree is identified by the head node. A branch is terminated when a unitary node (pixel) is reached. Each MR-node is defined by its size and the index of the brick it refers to, e.g. a brick of size $N \times M$ having bricks A and B as children, gathered in a specified sens. The aim of the preprocessing phase is twofold: building the binary tree (very fast) and building the brick database with the calculus of associated scattering matrices. The brick database and the binary tree are both built during the pre-processing phase.

3.4.2 A top-down approach

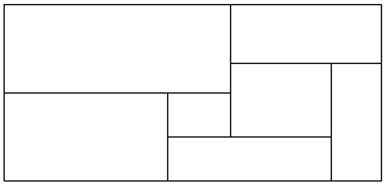


Figure 10: A bottom-up irregular gathering of MR-nodes could lead to this kind of arrangement: in this figure, MR-nodes cannot be brought together two by two, into rectangular higher level MR-nodes.

To built the binary tree, a bottom-up approach would consist to start from the bottom level where the nodes are the pixels, and to gather them recursively until all are gathered into the head-node of the binary tree. The advantage would be to take more accurately the local arrangement of walls and other obstacles into account. However, an ascending approach

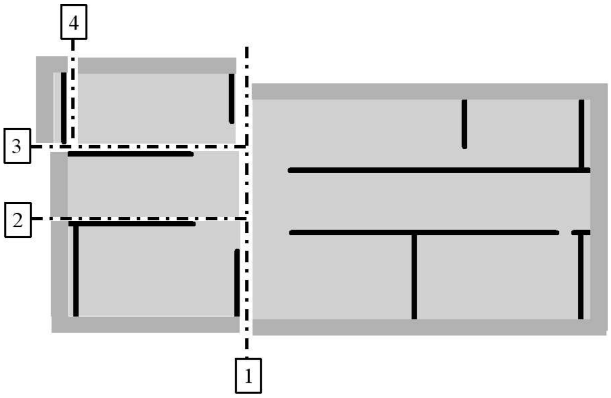


Figure 11: The test environment is compounded of free space (light grey), walls (black) and surrounding absorbing material (dark grey). The top-down dividing procedure is illustrated for the fourth first levels on one branch: the head-node is firstly divided along line 1, then left-hand child along line 2, then top-child along line 3, and finally left-hand child along line 4.

would not preserve both the rectangular shape of MR-nodes and the binary nature of the tree as illustrated in Fig.10. The management of other than rectangular structures would increase drastically the computational load of the tree structure leading globally to overhead computational and memory loads. In contrast, a top-down approach tackles the problem from the head node, breaking nodes down recursively into pairs of child MR-nodes. Several empirical algorithms can be developed to decide how each MR-node is divided. Keeping in mind the aim of reducing the memory needs, a first idea could be to try to minimize the number of MR-nodes. It is in this case obvious to show that the best approach consists in cutting each MR-node along a line in the middle of its higher length. This leads to the regular binary-tree. However this is probably not the most efficient cutting. Indeed, memory and computational loads depend rather on the number of bricks than on the number of MR-nodes. It appears indeed more efficient to try not to reduce the number of MR-nodes but rather the number of bricks.

For this purpose it is proposed to try to align cuts in MR-nodes along walls and obstacles, to reach as soon as possible homogeneous MR-nodes. The head-node is divided along its main discontinuity line and the same process is repeated recursively on its children nodes until unitary nodes are reached. This process is illustrated in Fig. 11. To align cuts along main discontinuities the following rule is followed :

- Select the longer side of a MR-node, (N pixels).

- Compute the number of discontinuities $D(i)$ for each possible splitting line l_i; $\forall i \in [1; N-1]$. In the case of a vertical cut, $D(i)$ is given by the number of lines j for which the material indices $n(i,j)$ and $n(i+1,j)$ are different

$$D(i) = \sum_{j=0}^{M-1} C\left(n(i-1,j) \neq n(i,j)\right) \tag{100}$$

For an horizontal cut, swap i and j.

- Split the block at the index i_m such as $D(i)$ is maximum

$$i_m = \underset{i\in[1;N-1]}{argmax} D(i) \tag{101}$$

Note however that a pure discontinuity based approach would lead to a wide overhead computational load when cutting a node near one side. To make a trade-off between the discontinuity based approach and the center based one, the following rules are chosen

$$i_m = \underset{i\in[1;N-1]}{argmax} D(i) \cdot C(i) \tag{102}$$

where

$$C(i) = 1 - \left|\frac{i-h_0}{h_0}\right|^p \tag{103}$$

with h_0 the half-length of the MR-node. p is given by

$$p = \begin{cases} 0 & \text{if } N < L, \\ K & \text{if } N \geq L \end{cases} \tag{104}$$

where K and L are settings parameters. Their meaning is the following. For MR-nodes having sides smaller than L, the splitting is discontinuity-based only. For larger MR-nodes the splitting line is chosen as a trade-off between discontinuity-based and center-based criteria.

3.4.3 Implementation

The method has been implemented on a JAVA virtual machine for portability purposes. Matrix computations are down using the COLT library developed by the CERN, Geneva. However, because of the high number of matrix inversion and multiplications done during the pre-processing phase, and because JAVA is not optimized for matrix computation, a Java interface (JNI) has been developed to perform calls to the BLAS (Basic Linear Algebra Subprogram)library. The BLAS library is a freely distributed library including procedures for intense matrix calculus which can be compiled specifically for each system. This approach has allowed us to save computational time. The whole computational time for the preprocessing phase decreased by a factor of 8 for a 500×500 pixels. Furthermore, higher is the environment size, higher is the saving time factor. The following results have been obtained on a PC pentium 2GHz.

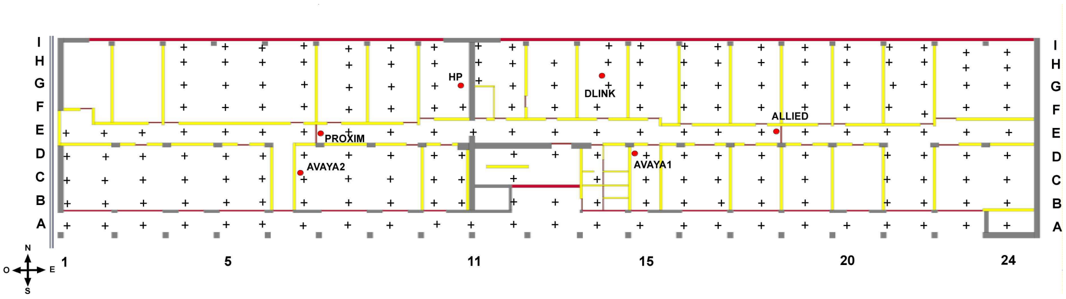


Figure 12: The reference environment (our lab) of 25mx100m is represented. black crosses represent measurement points.

4 Efficiency and applications

4.1 Fine wave propagation simulations

In this section, our laboratory is used as a typical environment representing a building floor of about $100m \times 25m$ in which a WiFi LAN is deployed, as illustrated in Fig.. Fig.13(a) shows the resulting field strength prediction at a frequency of $2.4GHz$, with a resolution step $dr = 2cm$, verifying (14). The preprocessing lasted $30mn$ while the full propagation only $45s$. As exhibited in Fig.13(b), narrow-band channel properties can be studied in depth. The local analyze of the field (amplitude and phase) may yield the spatial correlation of the channel response providing a way to obtain fine channel modeling (Rayleigh, Rice,...). DOA algorithms could be furthermore applied to predict the local channel angle spreading in each room. The only unaccessible thing is the time spreading because of the narrow-band estimation. It should be however emphasized that this could be overcome by using multiple simulations at near frequencies to build the wideband channel response. However, the higher the bandwidth is, the longer the computational time is. If it's probably not efficient to use the frequency domain formalism for UWB channel modeling, it is however possible to use it efficiently to estimate a WiFi impulse time response. Indeed, the full bandwidth of a WiFi pulse is in the order of $20MHz$. Since a realistic time spreading even in severe conditions is lower than $1\mu s$, the knowledge of the full time response requires only less than 20 spectrum lines and even less than 10 in fair conditions.

MR-FDPF is also a good candidate to be included in any wireless network simulator. The current increase of ad-hoc network studies calls indeed for accurate simulations of Indoor propagation [50, 4]. MR-FDPF allows to compute quickly radio links between nodes moving in the environment. Another interesting application concerns the simulation of multi-antennas mobile systems [37]. The accurate nature of MR-FDPF based simulations permit the computation of local field variations and therefore could yield easily MIMO channel simulations.

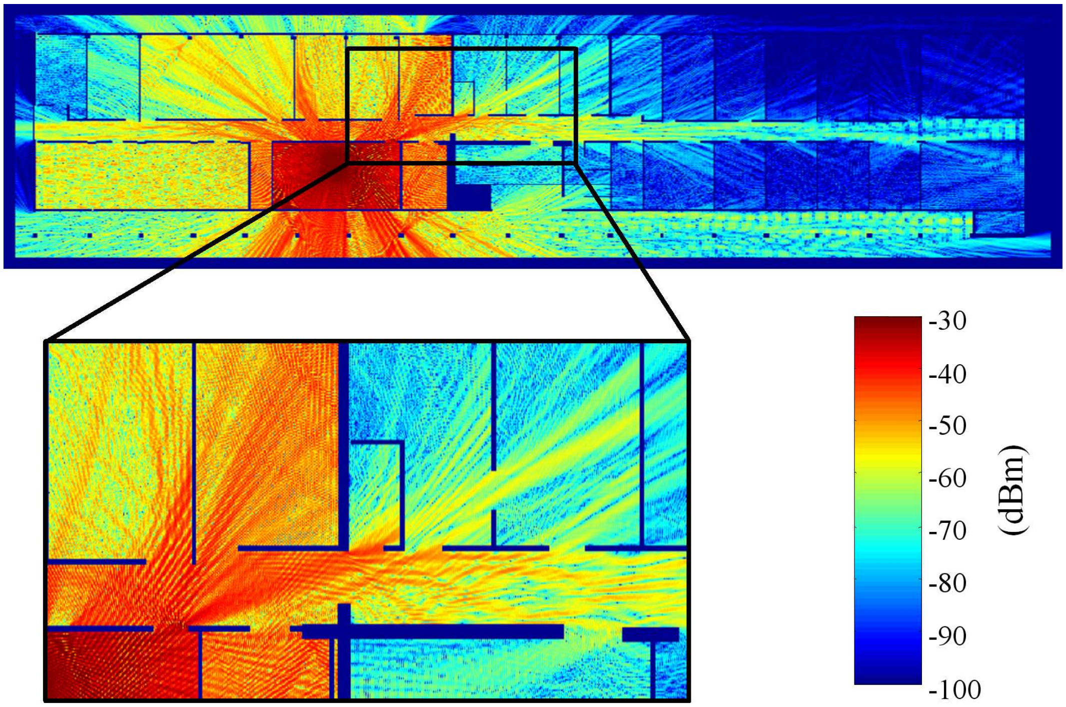


Figure 13: Field strength prediction for a 2.4Ghz source at a resolution of 2cm and a detailed area. Different effects such as reflection, diffraction, stationary waves, corridor effect, are shown.

4.2 Engine optimization for wLAN planning

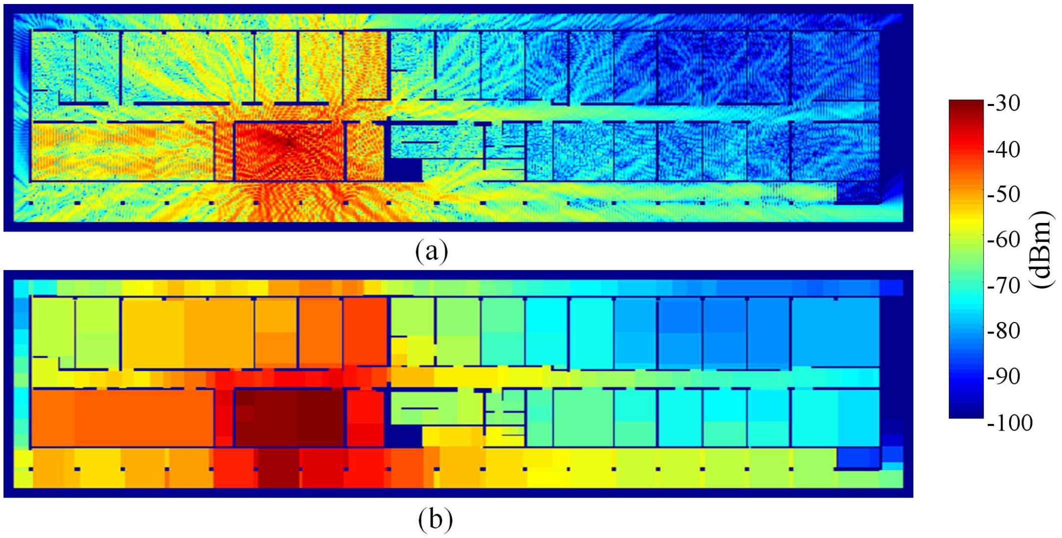
Among all of these applications, let us focusing now on the usual wLAN planning problem. Many efforts have been recently devoted to the development of efficient propagation tools in this context. As discussed in the introduction, standard approaches face three challenging problems:

- Numerous diffractions and reflections make empirical approaches not efficient while increasing severely the computational time of ray-tracing based approaches.
- The planning task implies dense field computation (many potential receivers).
- The planning task implies also to test numerous potential source locations.

The MR-FDPF approach is really adapted to counterbalance these hard points. Firstly, the computational time doesn't depend on the number of reflections. Furthermore, all diffractions and reflections are taken into account up to infinity as the inverse steady-state problem is solved in an exact way. Lastly, the most time-consuming phase, i.e. the preprocessing, is done once for all possible sources, allowing to test efficiently many sources. However, for a typical exemple as described above, the fine resolution requirement still leads to a computational time higher than expected. To reduce further the computational time, we propose the use of a simulation frequency lower than the true frequency. This was in fact already proposed for TDPF, by Chopard et al [11] for GSM network planning.

To choose the simulation frequency, the discretization step is firstly set according to the desired resolution of field predictions, involving the size of rooms and obstacles. For instance, assuming both the desired resolution is about $1m^2$ and obstacles are longer than $1m$ in length, the main constraint becomes that the wavelength should be lower than $1m$ to allow robust mean power estimations. Therefore a simulation frequency of 480MHz is chosen requiring a resolution step of $10cm$ to fulfill (14). Fig.14(a) exhibits the field strength estimation at the pixel level ($10cm$) for the previously described Indoor environment. It is obvious that exact positions of fading holes and peaks are not realistic because of the use of the intermediate frequency. But other reasons hold, which are the 2D approximation as well as the lack of knowledge of furniture, peoples, and wall constitutive materials. This approach can however provide a good estimation of the mean field strength, if a calibration process is used. Full details about the calibration process and measurement procedure are out of the scope of this paper. As an indication, the root mean square error (RMSE) was found of about $5dB$, computed over 250 measurement points, and after a calibration process described in [40]. For this purpose the mean strength value was estimated over homogeneous nodes, according to (99). The resulting coverage map is provided in Fig.14, for both the fine (a) and the homogeneous node (b) resolutions.

Concerning the computational and memory loads, final results are summarized in table A, for three binary trees: a regular one, a full discontinuity-based tree, and a trade-off between both approaches. The lower memory use is obtained with the regular tree but involving a large number of bricks (1009). It furthermore leads to very small homogeneous nodes since only 23% of the surface area is included in homogeneous blocks larger than 400



(a)

(b)

Figure 14: Field strength prediction with an intermediate frequency of 480MHz, at a resolution of 10cm: full resolution in (a) and homogeneous nodes resolution in (b). The color scale is the same as in the previous figure.

Table 1: Computational and memory loads

	method		
	regular	disc.	optimal (L=32,K=6)
memory	57.8 MO	86.9 MO	63.4 MO
CPU time	8.1 s	27 s	8.6 s
hom nodes; $S > 400pix$	23%	79%	70%
nb of bricks	1009	218	226
propag hom	361 ms	288 ms	203 ms
propag pix	1.65 s	1.77 s	1.64 s

pixels. The discontinuity based splitting algorithm in the opposite leads to much less bricks (218) and much more nodes larger than 400 pixels (79% of the surface area) while the CPU time increases by a factor of 3 and the memory use by a factor of 2. The trade-off approach provides a good compromise concerning the preprocessing. Moreover, the trade-off approach provides less propagation times than both other approaches for the pixel level propagation (in $1.64s$) and for the homogeneous level propagation (in $203ms$). It is difficult to compare these results with standard methods because environment sizes and discretization steps are always different. However in [56], Wolfle et al. gave results obtained on 2D simulations for a building floor of a comparable size. Their simulations lasted more than 20s with ray-tracing and less than one second for single path based approaches, either for a MWM approach or their dominant path method. This confirms our complexity study stating that the MR-FDPF complexity is comparable to the one of any direct path method. Fig.15 exhibits the impact of the resolution step on the computational time and memory needs. Memory needs curve fits accurately wit the theoretical model and the adaptive algorithm behaves as the regular one. The pre-processing time fits the asymptotic law except for last value corresponding to a $2cm$ resolution. In this case, the computational increase is due to the memory swap associated with large matrices. Concerning the propagation at level 1, all curves are identical and fit well with the theoretical asymptote. Indeed, a deeply study of the downward propagation would show that this complexity is not dependent on the way the tree is built. On the opposite, the homogeneous level propagation is sensitive to the tree. The reference curve (red) corresponds to the computational time for the pixel level propagation. For fine resolutions, the homogeneous level estimation associated with the criteria based tree allow to reduce by ten the computational time.

5 Conclusion and future works

This report described a new method called MR-FDPF, embedded in a propagation engine for Indoor propagation called WILDE (Wireless Design Tool). This engine has been integrated in a commercially available software, WIPLAN.

This method was based on the ParFlow formalism introduced by Luthi in its PhD thesis [30]. In his work, Luthi derived a general flow equation modeling. However, in the context of electromagnetism and radio-wave propagation, this approach is equivalent to the well-known TLM (Transmission Line Matrix) [22, 28, 38] model with the SCN (Super Condensed Node) restricted to a 2D modeling. We have proposed to develop the TLM system in the frequency domain. This has been already proposed (for instance see [7, 27, 21]) but only for the purpose of singular values and similar analysis of the propagation matrix. We obtained in the frequency domain a wide linear system to solve taking advantage of the harmonic model. This implies that our approach deals only with a steady-state computation of the electromagnetic field and cannot estimate the time-spread response of the environment. For Indoor WiFi like networks, it has been shown that the time-spreading is often shorter than the chip time of emitted signals.

To solve this wide linear system we developed an original approach exploiting the specific structure of the propagation matrix provided by the flows. The first step was to define the concept of a MR-node which is a generalization of the usual TLM node. The second step was to develop recursive relations between these MR-nodes, providing an efficient way to compute the propagation inside the environment. Finally, the last step was to propose a recursive dividing method of the head-node in order to obtain a binary tree of MR-nodes in which each branch is terminated by a conventional TLM node. However the dividing method was chosen to favor homogeneous nodes, which can be considered as terminal nodes if we want to accelerate the coverage computation.

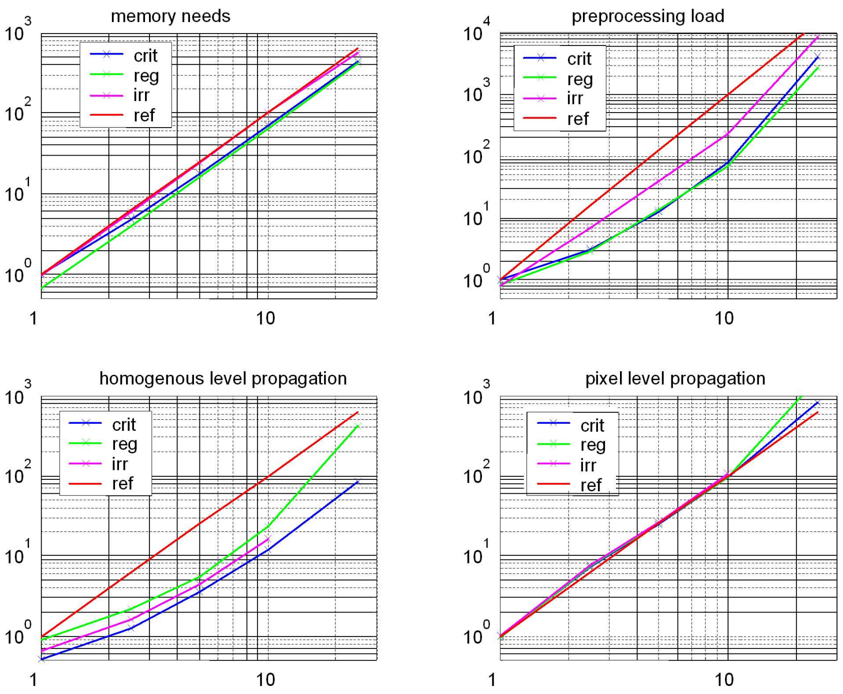


Figure 15: In these figures the computational requirements are plotted as function of the resolution step and for three kinds of tree : optimal (blue), regular (green), discontinuity based (magenta). The theoretical asymptotic functions are plotted in red. In each plot the abscissa corresponds to the ratio between the reference and the actual resolution. The ordinate corresponds to the relative cost (either time or memory) the reference being the one obtained with the adaptive tree for a resolution step of $50cm$ Discretization steps of $\{50cm, 20cm, 10cm, 5cm, 2cm\}$ haven been tested with the reference step of 50cm (x=1). The memory needs are plotted in (a), and the computational times in (b-d) respectively for preprocessing, propagation at homogeneous level and propagation at pixel level.

As a conclusion, it can be emphasized that the wide linear system obtained in the FDPF formalism is solved with no approximation by our approach, with a computational time equals to the computational time of $log_2(min(N_y, N_x))$ iterations in the time domain. This is a significant improvement, leading to the first finite difference base approach allowing to compute coverage of more than $10000m^2$ in less than on second, at our knowledge. The computational time is even probably lower than the computational time required by ray-tracing. However, our approach cannot gives the full time-spread response in Indoor environments, which remains the main advantage of ray-tracing like approaches. In the opposite, our approach allows to take into account all reflections and diffractions with no limit.

6 Appendix

In this appendix recursive bounds allowing to develop the MR-FDPF algorithm are revisited using a global formalism. It gives the proof that the proposed algorithm solved exactly the initial linear system.

6.1 Global formulation of the MR-node based approach

The reference formulation is that obtained at level 0 with exchange flows only as provided by equation (37) noted herein as:

$$\left(I_d - \underline{\Omega}_e^0\right) \overleftarrow{\underline{F}}_e^0 = \overleftarrow{\underline{S}}_e^0 \tag{105}$$

The reference method to solve this system is the geometric series derivation according to (40) and rewritten as

$$\overleftarrow{\underline{F}}_e^0 = \sum_{k=0}^{\infty} \left(\underline{\Omega}_e^0\right)^k \cdot \overleftarrow{\underline{S}}_e^0 = \overleftarrow{\underline{S}}_e^0 + \underline{\Omega}_e^0 \cdot \overleftarrow{\underline{S}}_e^0 + \left(\underline{\Omega}_e^0\right)^2 \cdot \overleftarrow{\underline{S}}_e^0 + \dots \tag{106}$$

In the non recursive MR-node based formulation (see section 3.1, (ℓ)-level flows are firtstly ordered row per row, node after node. They are then divided into two subsets: exchange and inner flows according to

$$\begin{aligned} \overleftarrow{\underline{F}}_e^\ell &= \underline{R}_0^\ell \cdot \overleftarrow{\underline{F}}_e^0 \\ \overleftarrow{\underline{F}}_i^\ell &= \overline{\underline{R}_0^\ell} \cdot \overleftarrow{\underline{F}}_e^0 \end{aligned} \tag{107}$$

where $\underline{R}_0^\ell$ holds for the projection of the (0)-level flows to the (ℓ)-level exchange flows.

By introducing (107) into (105), one obtains

$$\begin{pmatrix} Id - \underline{\Omega}_{0,ee}^\ell & -\underline{\Omega}_{0,ei}^\ell \\ -\underline{\Omega}_{0,ie}^\ell & Id - \underline{\Omega}_{0,ii}^\ell \end{pmatrix} \cdot \begin{pmatrix} \overleftarrow{\underline{F}}_e^\ell \\ \overleftarrow{\underline{F}}_i^\ell \end{pmatrix} = \begin{pmatrix} \overleftarrow{\underline{S}}_{0,e}^\ell \\ \overleftarrow{\underline{S}}_{0,i}^\ell \end{pmatrix} \tag{108}$$

$\overleftarrow{\underline{S}}_{0,e}^\ell$ and $\overleftarrow{\underline{S}}_{0,i}^\ell$ are respectively the projection of the (0)-level source on the (ℓ)-level exchange and inner flows spaces.

The four blocks of the propagation matrix are detailed below.

$$\underline{\Omega}^{\ell}_{0,ee} = \left(\begin{array}{cccccccc} [0] & \Omega_W(1) & [0] & & [0] & \Omega_N(k) & [0] & \\ \Omega_E(0) & [0] & \Omega_W(2) & & & & \ddots & \\ [0] & \ddots & \ddots & \ddots & & & & \Omega_N(.) \\ & & \ddots & \ddots & \ddots & & & [0] \\ [0] & & & \ddots & \ddots & \ddots & & \\ \Omega_S(0) & & & & \ddots & \ddots & \ddots & [0] \\ [0] & \ddots & & & & \Omega_E(K_{-}3) & [0] & \Omega_W(K_{-}1) \\ & [0] & \Omega_S(.) & [0] & & [0] & \Omega_E(K_{-}2) & [0] \end{array} \right) \tag{109}$$

with

$$\Omega_E(k) = \begin{pmatrix} \sigma_{EE}(k) & \sigma_{EW}(k) & \sigma_{ES}(k) & \sigma_{EN}(k) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad \Omega_W(k) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \sigma_{WE}(k) & \sigma_{WW}(k) & \sigma_{WS}(k) & \sigma_{WN}(k) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$
$$\Omega_S(k) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \sigma_{SE}(k) & \sigma_{SW}(k) & \sigma_{SS}(k) & \sigma_{SN}(k) \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad \Omega_N(k) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \sigma_{NE}(k) & \sigma_{NW}(k) & \sigma_{NS}(k) & \sigma_{NN}(k) \end{pmatrix}; \tag{110}$$

$\underline{\Omega}^{\ell}_{0,ei}$ has the same structure of $\underline{\Omega}^{\ell}_{0,ee}$ but with the following blocks:

$$\Omega_E(k) = \begin{pmatrix} \sigma_{E0}(k) \\ 0 \\ 0 \\ 0 \end{pmatrix}; \quad \Omega_W(k) = \begin{pmatrix} 0 \\ \sigma_{W0}(k) \\ 0 \\ 0 \end{pmatrix}; \quad \Omega_S(k) = \begin{pmatrix} 0 \\ 0 \\ \sigma_{S0}(k) \\ 0 \end{pmatrix}; \quad \Omega_N(k) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{N0}(k) \end{pmatrix}; \tag{111}$$

The two other propagation matrices are block-diagonal, the k^{th} block being defined according to:

$$\begin{array}{l} \underline{\Omega}^{\ell}_{0,ii}(k,k) = \sigma_{00} \\ \underline{\Omega}^{\ell}_{0,ie}(k,k) = \left(\sigma_{0E}k \; \sigma_{0W}k \; \sigma_{0S}k \; \sigma_{0N}k \right) \end{array} \tag{112}$$

At this point the (0)-level formulation has been expressed at level (ℓ) involving both inner and exchange flows. To obtain the exchange flows only based formulation, (128) is firstly solved with respect to $\overleftarrow{\underline{F}}^{\ell}_{e}$ leading to

$$\left(Id - \underline{\Omega}^{\ell}_{e} \right) \cdot \overleftarrow{\underline{F}}^{\ell}_{e} = \overleftarrow{\underline{S}}^{\ell}_{e} \tag{113}$$

with

$$\underline{\Omega}^{\ell}_{e} = \underline{\Omega}^{\ell}_{0,ee} + \underline{\Omega}^{\ell}_{0,ei} \cdot \left(Id - \underline{\Omega}^{\ell}_{0,ii} \right)^{-1} \cdot \underline{\Omega}^{\ell}_{0,ie} \tag{114}$$

and

$$\overleftarrow{\underline{S}}^{\ell}_{e} = \overleftarrow{\underline{S}}^{\ell}_{0,e} + \underline{\Omega}^{\ell}_{0,ei} \cdot \left(Id - \underline{\Omega}^{\ell}_{0,ii} \right)^{-1} \cdot \overleftarrow{\underline{S}}^{\ell}_{0,i} \tag{115}$$

Finally, solving (128) with respect to $\overleftarrow{\underline{F}}_i^\ell$ leads to

$$\overleftarrow{\underline{F}}_i^\ell = \left(Id - \underline{\Omega}_{0,ii}^\ell\right)^{-1} \cdot \overleftarrow{\underline{S}}_i^\ell + \left(Id - \underline{\Omega}_{0,ii}^\ell\right)^{-1} \cdot \underline{\Omega}_{0,ie}^\ell \cdot \overleftarrow{\underline{F}}_e^\ell \tag{116}$$

These 4 equations can be easily related to the local relations found in section 3.1.2.

- **The Inner step**. It corresponds to the propagation of the elementary source inside its MR-node as if it was isolated from the rest of the world. Let the global inner propagation matrix be defined as

  $$\underline{I}^\ell = \left(Id - \underline{\Omega}_{0,ii}^\ell\right)^{-1} \tag{117}$$

  The diagonal blocks of $\underline{I}^\ell$ are nothing else than $I(b_k^\ell)$ provided in (57). This matrix computes independently the steady-state when MR-nodes are isolated. This local solution is written :

  $$\overleftarrow{\underline{F}}_i^\ell = \underline{I}^\ell \cdot \overline{\underline{R}_0^\ell} \cdot \underline{S}_e^0 \tag{118}$$

  It applies node per node, on inner flows only. At this point, $\overleftarrow{\underline{F}}_i^\ell$ is nothing else than the solution of the ParFlow equation solved on each MR-node independently.

- **The Upward step**. Outward flows of the source MR-node are computed, providing the equivalent source node. Let the upward matrix be defined by

  $$\underline{U}^\ell = \underline{\Omega}_{0,ei}^\ell \tag{119}$$

  Introducing this matrix, the steady-state equivalent source is given by:

  $$\overleftarrow{\underline{S}}_e^\ell = \left(\underline{R}_0^\ell + \underline{U}^\ell \underline{I}^\ell \cdot \overline{\underline{R}_0^\ell}\right) \cdot \overleftarrow{\underline{S}}_e^0 \tag{120}$$

  The steady-state node source is obtained as the sum of the source flows naturally located on the bounds of the node with the outward flows resulting from the propagation of the other source flows inside the MR-node itself.

- **The Iterative scattering step**. The iterative process involves now the propagation matrix restricted to exchange flows according to (114). Therefore, inner flows are naturally withdrawn during the iterative step. The propagation equation (105) is thus replaced by (113) and the system is solved iteratively according to:

  $$\overleftarrow{\underline{F}}_e^\ell = \sum_{k=0}^{\infty} \left(\underline{\Omega}_e^\ell\right)^k \cdot \overleftarrow{\underline{S}}_e^\ell \tag{121}$$

- **The Downward step**. It corresponds to the computation of inner flows in each MR-node, when the steady-state is reached with exchange flows. Let the downward matrix be defined by:

  $$\underline{D}^\ell = \underline{\Omega}_{0,ie}^\ell \tag{122}$$

In this step, the incoming flows are propagated toward inner flows in each MR-node, independently. Including the inner step directly into the equation, the downward equation becomes

$$\overleftarrow{\underline{F}}_i^\ell = \underline{I}^\ell \cdot \left(\overleftarrow{\underline{S}}_i^\ell + \underline{D}^\ell \cdot \overleftarrow{\underline{F}}_e^\ell\right) \tag{123}$$

Finally, the (0)-level exchange flows can be expressed as a function of (ℓ)-level exchange flows and (0)-level source only as

$$\overleftarrow{\underline{F}}_e^0 = \overline{\underline{R}_0^\ell}^t \cdot \underline{I}^\ell \cdot \overline{\underline{R}_0^\ell} \cdot \overleftarrow{\underline{S}}_e^0 + \left(\underline{R}_0^{\ell\,t} + \overline{\underline{R}_0^\ell}^t \cdot \underline{D}^\ell\right) \cdot \overleftarrow{\underline{F}}_e^\ell \tag{124}$$

The equivalence between usual and node based formulation can be synthesized with the following equivalence

$$\left(I_d - \underline{\Omega}_e^0\right) \overleftarrow{\underline{F}}_e^0 = \overleftarrow{\underline{S}}_e^0 \Longleftrightarrow \left\{ \begin{array}{l} \overleftarrow{\underline{S}}_e^\ell = \underline{\Omega}_{Up}^\ell \cdot \overleftarrow{\underline{S}}_e^0 \\ \left(I_d - \underline{\Omega}_e^\ell\right) \cdot \overleftarrow{\underline{F}}_e^\ell = \overleftarrow{\underline{S}}_e^\ell \\ \overleftarrow{\underline{F}}_e^0 = \underline{\Omega}_{In}^\ell \cdot \overleftarrow{\underline{S}}_e^0 + \underline{\Omega}_{Dn}^\ell \cdot \overleftarrow{\underline{F}}_e^\ell \end{array} \right. \tag{125}$$

with

$$\begin{aligned} \underline{\Omega}_{Up}^\ell &= \underline{R}_0^\ell + \underline{U}^\ell \underline{I}^\ell \cdot \overline{\underline{R}_0^\ell} \\ \underline{\Omega}_{In}^\ell &= \overline{\underline{R}_0^\ell}^t \cdot \underline{I}^\ell \cdot \overline{\underline{R}_0^\ell} \\ \underline{\Omega}_{Dn}^\ell &= \underline{R}_0^{\ell\,t} + \overline{\underline{R}_0^\ell}^t \cdot \underline{D}^\ell \end{aligned} \tag{126}$$

6.2 Global formulation of the MR-FDPF recursion

Starting now from the (ℓ)-level formulation the ($\ell+1$)-level formulation is established providing the MR-FDPF recursion. The principle is the same as used in previous section. Let starting from (113). The (ℓ)-nodes are gathered per two in ($\ell+1$)-nodes. Then, the (ℓ)-level flow vector are divided into two subsets according to

$$\begin{aligned} \overleftarrow{\underline{F}}_e^{\ell+1} &= \underline{R}_\ell^{\ell+1} \cdot \overleftarrow{\underline{F}}_e^\ell \\ \overleftarrow{\underline{F}}_i^{\ell+1} &= \overline{\underline{R}_\ell^{\ell+1}} \cdot \overleftarrow{\underline{F}}_e^\ell \end{aligned} \tag{127}$$

where $\underline{R}_\ell^{\ell+1}$ holds for the projection from (ℓ) to ($\ell+1$)-level exchange flows.

Eq.(113) now expends as :

$$\begin{pmatrix} Id - \underline{\Omega}_{\ell,ee}^{\ell+1} & -\underline{\Omega}_{\ell,ei}^{\ell+1} \\ -\underline{\Omega}_{\ell,ie}^{\ell+1} & Id - \underline{\Omega}_{\ell,ii}^{\ell+1} \end{pmatrix} \cdot \begin{pmatrix} \overleftarrow{\underline{F}}_e^{\ell+1} \\ \overleftarrow{\underline{F}}_i^{\ell+1} \end{pmatrix} = \begin{pmatrix} \overleftarrow{\underline{S}}_{\ell,e}^{\ell+1} \\ \overleftarrow{\underline{S}}_{\ell,i}^{\ell+1} \end{pmatrix} \tag{128}$$

$\overleftarrow{\underline{S}}_{\ell,e}^{\ell+1}$ and $\overleftarrow{\underline{S}}_{\ell,i}^{\ell+1}$ are the projections of the (ℓ)-level source on respectively the exchange and inner flows spaces of level ($\ell+1$).

Following the development of the previous section, the recursion relations are obtained as

$$\begin{aligned}
\overleftarrow{\underline{S}}_e^{\ell+1} &= \left(\underline{R}_\ell^{\ell+1} + \underline{U}^{\ell+1}\underline{I}^{\ell+1} \cdot \overline{\underline{R}_\ell^{\ell+1}}\right) \cdot \overleftarrow{\underline{S}}_e^{\ell} \\
\overleftarrow{\underline{F}}_e^{\ell} &= \overline{\underline{R}_\ell^{\ell+1}}^t \cdot \underline{I}^{\ell+1} \cdot \overline{\underline{R}_\ell^{\ell+1}} \cdot \overleftarrow{\underline{S}}_e^{\ell} + \left(\underline{R}_\ell^{\ell+1t} + \overline{\underline{R}_\ell^{\ell+1}}^t \cdot \underline{D}^{\ell+1}\right) \cdot \overleftarrow{\underline{F}}_e^{\ell+1}
\end{aligned} \tag{129}$$

The up-and-down recursion works then with these two equations, having initial and boundary conditions imposed by

$$\begin{aligned}
\overleftarrow{\underline{S}}_e^{0} &= \overleftarrow{\underline{S}}_0 \\
\overleftarrow{\underline{F}}_e^{L-1} &= 0
\end{aligned} \tag{130}$$

At the end, the exact solution of the linear system is solved, according to the following equations

$$\left(I_d - \underline{\Omega}_e^{\ell}\right) \overleftarrow{\underline{F}}_e^{\ell} = \overleftarrow{\underline{S}}_e^{\ell} \Longleftrightarrow \begin{cases} \overleftarrow{\underline{S}}_e^{\ell+1} = \underline{\Omega}_{\ell,Up}^{\ell+1} \cdot \overleftarrow{\underline{S}}_e^{\ell} \\ \\ \overleftarrow{\underline{F}}_e^{\ell} = \underline{\Omega}_{\ell,In}^{\ell+1} \cdot \overleftarrow{\underline{S}}_e^{\ell} + \underline{\Omega}_{\ell,Dn}^{\ell+1} \cdot \overleftarrow{\underline{F}}_e^{\ell+1} \end{cases} \tag{131}$$

which can be solved with

Algorithm 6.1

```
Upward phase:
|  For $\ell = 0$ to $L-2$, Do
|      $\overleftarrow{\underline{S}}_e^{\ell+1} = \underline{\Omega}_{\ell,Up}^{\ell+1} \cdot \overleftarrow{\underline{S}}_e^{\ell}$
Downward phase:
|  For $\ell = L-2$ down to 0, Do
|      $\overleftarrow{\underline{F}}_e^{\ell} = \underline{\Omega}_{\ell,In}^{\ell+1} \cdot \overleftarrow{\underline{S}}_e^{\ell} + \underline{\Omega}_{\ell,Dn}^{\ell+1} \cdot \overleftarrow{\underline{F}}_e^{\ell+1}$
```

References

[1] *Digital Mobile Radio towards Future Generation Systems, COST 231 Final Report*, chapter Chap 4 : Propagation Prediction Models, pages 115–208. European Union, Brussels, 1996.

[2] F. Aguado Agelet, F.P. Fontan, and A. Formella. Fast ray tracing for microcellular and indoor environments. *IEEE transactions on Magnetics*, 33(2):1484–1487, march 1997.

[3] Fernando Aguado Agelet and al. Efficient ray-tracing acceleration techniques for radio propagation modeling. *IEEE transactions on Vehicular Technology*, 49(6):2089–2104, november 2000.

[4] Amit P. Jardosh ans Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Real-world environment models for mobile network evaluation. *IEEE Journal on selected areas in communications*, 23(3):622–632, march 2005.

[5] Georgia E. Athanasiadou and Andrew R. Nix. Investigation into the sensitivity of the power predictions of a microcellular ray tracing propagation model. *IEEE transactions on Vehicular Technology*, 49(4):1140–1151, July 2000.

[6] Georgia E. Athanasiadou and Andrew R. Nix. A novel 3D indoor ray-tracing propagation model: The path generator and evaluation of narrow-band and wide-band predictions. *IEEE transactions on Vehicular Technology*, 49(4):1152–1168, July 2000.

[7] P. Berini and K. Wu. A new frequency domain symmetrical condensed TLM node. *IEEE Microwave Guided Wave Lett.*, 4:180–182, 1994.

[8] H. L. Bertoni, W. Honcharenko, L. R. Maciel, and H. H. Xia. UHF propagation prediction for wireless personal communications. *Proceedings of the IEEE*, 82(9):1333–1359, 1994.

[9] Zhongqiang Chen, Henry L. Bertoni, and Alex Delis. Progressive and approximate techniques in ray-tracing-based radio wave propagation prediction models. *IEEE transactions on Antennas and Propagation*, 52(1):240–251, January 2004.

[10] K-W. Cheung, J. H-M Sau, and R.D. Murch. A new empirical model for indoor propagation prediction. *IEEE transactions on Vehicular Technology*, 47(3):996–1001, August 1998.

[11] B. Chopard, P.O. Luthi, and J.F. Wagen. A lattice Boltzmann method for wave propagation in urban microcells. In *IEE Proceedings - Microwaves, Antennas and Propagation*, volume 144, pages 251–255, 1997.

[12] S.J. Fortune, D.M. Gay, B.W. Kernighan, O. Landron, R.A. Valenzuela, and M.H. Wright. Wise design of indoor wireless systems : practical computation and optimization. *Computer Science and Engineering; IEEE*, 2(1):58–68, spring 1995.

[13] Steven Fortune. A beam tracing algorithm for prediction of indoor radio propagation. Technical report, February 1996.

[14] Steven Fortune. Algorithms for prediction of indoor radio propagation. Technical report, AT& T Bell Laboratories report, January 1998.

[15] Massimo Franceschetti, Jehoshua Bruck, and Leonard J. Schulman. A random walk model of wave propagation. *IEEE transactions on Antennas and Propagation*, 52(5):1304–1317, May 2004.

[16] Jean-Marie Gorce, S. Frenot, V. Crespo, and S. Ubeda. Modélisation de la propagation en environnement indoor dans la bande de fréquences UHF. In *International Conference on Image and Signal Processing (ICISP)*, Agadir, 2001.

[17] Jean-Marie Gorce, E. Jullo, and K. Runser. An adaptative multi-resolution algorithm for 2D simulations of indoor propagation. In *Proceedings of the 12th International Conference on Antennas and Propagation*, pages 216–219, Exeter, UK, April 2003. IEE.

[18] Jean-Marie Gorce, K. Runser, and G. de la Roche. FDTD based efficient 2D simulations of indoor propagation for wireless LAN. In *IMACS, World Congress Scientific Computation, Applied Mathematics and Simulation*, Paris, France, July 2005.

[19] Jean-Marie Gorce and S. Ubeda. Propagation simulation ith the parflow method : fast computation using a multi-resolution scheme. In *IEEE 54th Vehicular Technology Conference*, pages 1304–1307, Atlantic City, October 2001.

[20] Mudhafar Hassan-Ali and Kaveh Pahlavan. A new statistical model for site-specific indoor radio propagation prediction based on geometric optics and geometric probability. *IEEE transactions on Wireless Communications*, 1(1):112–124, January 2002.

[21] J. Hesselbarth and R. Vahldieck. Accuracy of the frequency-domain TLM method and its application to microwave circuits. *International journal of numerical modelling : electronic networks, devices and fields*, 15:371–383, may 2002.

[22] W.J.R. Hoefer. The transmission line matrix method - theory and applications. *IEEE Transactions on Microwaves Theory Technique.*, 33(4):882–893, 1985.

[23] W. Honcharenko, H.L. Bertoni, .L. Dailing, J. Qian, and H.D. Yee. Mechanisms governing UHF propagation on single floors in modern office buildings. *IEEE transactions on Antennas and Propagation*, 41(6):787–790, 1993.

[24] R. Hoppe, P. Wertz, F.M. Landstorfer, and G. Wölfle. Advanced ray-optical wave propagation modelling for urban and indoor scenarios including wideband properties. *European Transactions on Telecommunications*, 14:61–69, 2003.

[25] Tetsuro Imai and Teruya Fujii. Fast algorithm for indoor microcell area prediction system using ray-tracing method. *Electronics and Communications in Japan, part 1*, 85(6):41–52, 2002.

[26] Zhong Ji, Bin-Hong Li, Hao-Xing Wang, Hsing-Yi Chen, and Tapan K. Sarkar. Efficient ray-tracing methods for propagation prediction for indoor wireless propagation. *IEEE Antennas and Propagation Magazine*, 43(2):41–49, April 2001.

[27] H. Jin and R. Vahldieck. The frequency-domain transmission line matrix method : a new concept. *IEEE Trans. Microwave Theory Techniques*, 40:2207–2218, 1992.

[28] PB. Johns. A symmetrical condensed node for the TLM method. *IEEE trans. on Microwave theory Techniques*, 35:370–377, April 1987.

[29] J.W.H. Lee and A. K. Y. Lai. FDTD analysis of indoor radio propagation. In *IEEE Antennas Propagation Society International Symposium*, volume 3, pages 1664–1667, Atlanta, GA, June 1998.

[30] Pascal Olivier Luthi. *Lattice Wave Automata : from radiowave to fracture propagation.* PhD thesis, University of Geneva, Geneva, Switzerland, march 1998.

[31] J.W. McKnown and R.L. Hamilton. Ray tracing as design tool for radio networks. *IEEE Network Magazine*, 5:27–30, November 1991.

[32] A. Neskovic, N. Neskovic, and G. Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. *IEEE communication surveys, http://www.comsoc.org/pubs/surveys*, pages 2–12, October 2000.

[33] K. Pahlavah and A. H. Lavesque. *Wireless information networks*, pages 195–196. Wiley Series in Telecommunications and Signal Processing. Wiley, New York, USA, 1995.

[34] Manish A. Panjwani, A. Lynn Abbott, and Theodore S. Rappaport. Interactive computation of coverage regions for wireless communication in multifloored indoor environments. *IEEE Journal on Selected Areas in Communications*, 14(3):420–430, April 1996.

[35] P. Pechac and M. Klepal. Effective indoor propagation predictions. In *IEEE Vehicular Technology Conference VTC-2001*, Atlantic City, NJ, October 2001.

[36] A. Rajkumar, B.F. Naylor, F. Feisullin, and L. Rogers. Predicting RF coverage in large environments using ray-beam tracing and partitioning tree represented geometry. Technical report, Technical Memorendum, AT&T Bell Laboratories, 1995.

[37] Ram Ramanathan, Jason Redi, Cesar Santivanez, David Wiggins, and Stephen Polit. Ad-hoc networking with directional antennas: a complete system solution. *Wireless Networks*, 23(3):496–506, march 2005.

[38] Jurgen Rebel. *On the foundation of the transmission Line Matrix Method*. PhD thesis, Universitat Munchen, Munchen, Deutchland, dec 1999.

[39] R. Rebeyrotte, G. de la Roche, K. Runser, and Jean-Marie Gorce. A new strategy for indoor propagation fast computation. In *IASTED, Antennas and Radar Propagation*, Banff, Canada, July 2005.

[40] K. Runser and Jean-Marie Gorce. Assessment of a new indoor propagation prediction model based on a multi-resolution algorithm. In *IEEE Vehicular Technology Conference, spring*, Stockholm, May 2005.

[41] R. Sato and H. Shirai. Simplified analysis for indoor propagation of a WLAN channel. In *IEEE Topical Conference on Wireless Communication Technology*, volume CDrom, Honolulu, Hawai, USA, October 2003.

[42] S. Y. Seidel, K. R. Schaubach, T.T. Tran, and T.S. Rappaport. Research in site-specific propagation modeling for pcs system design. In *Proceedings of the 43rd IEEE Vehicular Technology Conference*, pages 261–264, Secaucus, NJ, USA, May 1993.

[43] S.Y. Seidel and T.S. Rappaport. 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE transactions on Antennas and Propagation*, 40(2):207–217, 1992.

[44] S.Y. Seidel and T.S. Rappaport. Site specific propagation prediction for wireless in-building personal communication system design. *IEEE transactions on Vehicular Technology*, 43(4), 1994.

[45] Yuriy Olegovich Shlepnev. Trefftz-type brick finite elements for electromagnetics. In *proc. of the 17 annual review of progress in applied computational electromagnetics*, Monterey, March 2001.

[46] Yuriy Olegovich Shlepnev. Trefftz finite elements for electromagnetics. *IEEE trans. on Microwave theory Techniques*, 50(5):1328–1339, May 2002.

[47] Hajime Suzuki and Ananda S. Mohan. Ray tube tracing method for predicting indoor channel characteristics map. *Electronics Letters*, 33:1495–1496, August 1997.

[48] Hajime Suzuki and Ananda S. Mohan. Frustum ray tracing technique for high spatial resolution channel characteristic map. In *RawCon'98 proceedings*, pages 253–256, august 1998.

[49] Hajime Suzuki and Ananda S. Mohan. Measurement and prediction of high spatial resolution indoor radio channel characteristic map. *IEEE Transactions on Vehicular Technology*, 49(4):1321–1333, July 2000.

[50] Mineo Takai, Rrajive Bagrodia, Ken Tang, and Mario Gerla. Efficient wireless network simulations with detailed propagation models. *Wireless Networks*, 7:297–305, 2001.

[51] L. Talbi. FDTD characterization of the indoor propagation. *Journal of electromagnetic waves and applications*, 10(2):243–247, 1996.

[52] R.P. Torres, L. Valle, and M. Domingo. Cindoor : an engineering tool for planning and design of wireless systems in enclosed spaces. *IEEE Antennas and Propagation Magazine*, 41(4):11–22, 1999.

[53] R.A. Valenzuela. A ray tracing approach to predicting indoor wireless transmission. In *Proceedings of the 43rd IEEE Vehicular Technology Conference*, pages 214–218, May 1993.

[54] G. Wölfle and F.M. Landstorfer. Dominant path for the field strength prediction. In *IEEE Vehicular Technology Conference VTC*, pages 552–556, Ottawa, Ontario, May 1998.

[55] G. Wölfle, R. Wahl, P. Wertz, P. Wildbolz, and F. Landstorfer. Dominant path prediction model for indoor scenarios. In *German Microwave Conference (GeMIC)*, Ulm, Germany, April 2005.

[56] Gerard Wölfle, René Wahl, Pascal Wildbolz, and Philipp Wertz. Dominant path prediction model for indoor and urban scenarios. In *11th COST 273*, Germany, September 2004.

[57] Chang-Fa Yang, Boau-Cheng Wu, and Chuen-Jyi Ko. A ray-tracing method for modeling indoor wave propagation and penetration. *IEEE transactions on Antennas and Propagation*, 46(6):907–919, June 1998.

Contents

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399