Deterministic Approach for Fast Simulations of Indoor Radio Wave Propagation

Jean-Marie Gorce, *Member, IEEE,* Katia Runser, *Member, IEEE,* and Guillaume de la Roche, *Student Member, IEEE,*

***Abstract*— This paper describes the multi-resolution frequency domain ParFlow (MR-FDPF) approach for simulating radio wave propagation in indoor environments. This method allows for a better understanding of indoor propagation and hence greatly assists the development of WiFi-like network planning tools. The efficiency of such wireless design tools is strongly impacted by the quality of the coverage predictions which have to be estimated with a limited computational load. The usual approaches are based either on an empirical modeling relying on measurement campaigns or on geometrical optics leading to ray-tracing. While the former approach suffers from a lack of accuracy, the later one needs to balance accuracy with computational load requirements. The new approach proposed herein is based on a finite difference formalism, i.e. the transmission line matrix (TLM). Once the problem is developed in the frequency domain, the linear system thus obtained is solved in two steps: a pre-processing step which consists of an adaptive multi-resolution (multi-grid) pre-conditioning and a propagation step. The first step computes a multi-resolution data structure represented as a binary tree. In the second step the coverage of a point source is obtained by up-and-down propagating through the binary tree. This approach provides an exact solution for the linear system whilst significantly reducing the computational complexity when compared with the time domain approach.**

***Index Terms*— indoor propagation, wave propagation, simulation, ParFlow, frequency domain, wLAN planning, TLM.**

I. INTRODUCTION

THE simulation of decimetric wave propagation in indoor environments is an important task for wireless networks. Although having been studied for more than fifteen years [1]–[4], the problem of finding the right balance between the computational load and accuracy remains an open issue. New applications and technologies such as ad hoc or sensor networking, multiple antennas transmission, etc., call for more specific propagation models. Two kinds of propagation models are widely used for indoor pico-cells, i.e. empirical and deterministic models (see [1], [5] for a brief survey). Due to the presence of multiple obstacles the simplest empirical model (the one-slope model) fails to provide accurate predictions. The multi-wall model (MWM) [1] was proposed to take walls into account but only along the direct path. Even with the modified MWM later proposed [6], multiple paths are not considered and the path-loss remains a function of the transmitter-receiver distance. With respect to deterministic approaches, many works have been devoted to techniques based on geometrical optics since the beginning of the nineties [2], [4], [7]–[12]. These methods are very attractive but accuracy is obtained at the price of a high computational load. Indeed, this computational load is proportional to the number of launched rays and increases exponentially with the number of reflections each ray undergoes. The large bulk of recent papers focused either on improving the accuracy [13]–[15] or on reducing the computational complexity [16]–[20]. For computational purposes the number of reflections is often limited to about five per ray. This is definitely not enough for indoor environments because of some peculiar effects such as wave guiding in corridors. Wölfe *et. al.* [16] overcame this limit by introducing the dominant path model. As also proposed for other modern ray-tracers [13], [17], [18], their approach is based on the computation of a visibility graph aiming at reducing the time-consuming search of rays and walls intersections. The dominant path model is then obtained by removing all paths excepted the one providing the most power.

Hassan-Ali *et. al.* [21] have also proposed an empirical approach enhanced by exploiting geometric optics. An ellipsoid having its centroids at the emitter and receiver locations is traced (like the Fresnel ellipsoid). The probability of multi-effects in each room having an intersection with this ellipsoid is then used to compute a mean path-loss.

Compared to approaches based on empirical and geometric optics, only few works tackled this problem using a finite elements modeling [22]–[24]. The reason is of course, the high computational load usually required by these approaches. However Luthi *et. al.* [25], [26] have proposed in 1998 a new discrete approach referred to as *ParFlow*, based on the cellular automaton formalism, and applied to 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; however, the required spatial resolution for this to work well is theoretically very high. More details about ParFlow are presented and discussed in Section II.

This method appears to suit well the indoor environment. Firstly, the computational load does not increase with the number of reflections. Secondly, any shape of obstacle can be easily handled. To reduce the computational load, the ParFlow theory is firstly transposed in the frequency domain leading to the Frequency Domain ParFlow (FDPF) approach as described in [27]. In Section III an original way to solve this type of problem is developed by exploiting a multi-

Manuscript received October, 2005. Revised July 2006. Accepted November 2006. This work was supported by the French government (ACI) and the Rhône-Alpes Regional Council.

The authors are with the National Institute of Applied Sciences, Lyon, France, and the ARES project of INRIA Rhône-Alpes France (e-mail: jean-marie.gorce@insa-lyon.fr).

resolution structure. The main computational load is gathered into a pre-processing stage happening before the placement of the transmitters. Implementation choices and complexity are discussed in Section IV. In Section V applications and results are discussed.

II. PARFLOW THEORY

This section presents an overview of the Frequency-Domain ParFlow (FDPF) method. Although the theoretical background is different, the ParFlow algorithm is similar to the Transmission Line Matrix method [28]–[30] used for circuit design. The implementation in time-domain outlined in section II-A leads to a cellular automaton modeling. In this approach, the computational time can be kept as low as possible thanks to grid computing but also to the use of an intermediate frequency lower than the true RF frequency [25], [26]. In order to reduce further the computational load we propose to exploit a frequency domain formulation leading us to restrict the study to a narrowband estimation around the carrier frequency. The exact frequency domain formulation is provided in section II-B. This formulation is the starting point of the new multi-resolution approach described in section III. Note that for the sake of simplicity the problem is addressed in two dimensions (2D) only.

A. Time domain ParFlow formulation

As a finite difference approach ParFlow (Partial Flows) [25], [26] relies on the first order approximation of the wave equation on a $2D$ regular grid according to

$$\begin{aligned}&\Psi(r,t-dt)-2\cdot\Psi(r,t)+\Psi(r,t+dt)\\&=-\frac{c_0^2\cdot dt^2}{n_r^2\cdot dr^2}\cdot\left(4\cdot\Psi(r,t)-\sum_i\Psi(r+dr_i,t)\right),\end{aligned}\quad(1)$$

where $\Psi(r,t)$ is the electrical field in r at time t, c_0 the speed of light and n_r the refraction index. dr and dt are respectively the space and time steps. $r+dr_i$ refers to a neighbor pixel, $i\in\{E,W,S,N\}$ and E,W,S,N relate to the flows' directions as described below.

The specificity of ParFlow relies on the decomposition of the electrical field in flows: 4 directive outward flows and an additional stationary flow [26] as depicted in Fig.1 and named respectively $\overrightarrow{f_d}$ and $\breve{f}_0$. The stationary (or inner) flow is used for modeling a dielectric media having a relative permittivity $\epsilon_r\neq 1$. The electrical field $\Psi(r,t)$ is expressed in terms of the local flows as

$$\begin{aligned}\Psi(r,t)=n_r^{-2}\cdot\Big(&\overrightarrow{f_E}(r,t)+\overrightarrow{f_W}(r,t)+\overrightarrow{f_S}(r,t)\\&+\overrightarrow{f_N}(r,t)+Y_r\cdot\breve{f}_0(r,t)\Big),\end{aligned}\quad(2)$$

where $Y_r=4n_r^2-4$ is the local admittance.

Outward flows (over right-arrows) of a given pixel are also considered as inward flows (over left-arrows) for adjacent pixels according to

$$\overleftarrow{f_i}(r+dr_i,t)=\overrightarrow{f_i}(r,t);\quad i\in\{E,W,S,N\}.\quad(3)$$

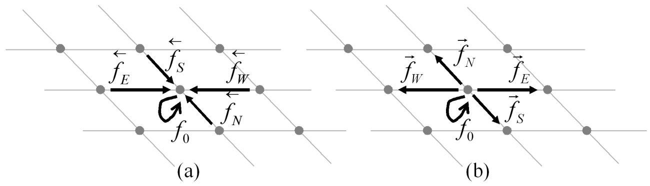


Fig. 1. *Inward (a) and outward (b) flows are associated with each pixel.*

At each pixel a local scattering equation describes the discrete time evolution of the flows, i.e.

$$\overrightarrow{F}(r,t)=\Sigma(r)\cdot\overleftarrow{F}(r,t-dt)+\overrightarrow{S}(r,t),\quad(4)$$

where $\overrightarrow{S}(r,t)$ contains source flows (null if the source is not in r) and where

$$\begin{aligned}\overleftarrow{F}(r,t)&=\begin{pmatrix}\overleftarrow{f_E}&\overleftarrow{f_W}&\overleftarrow{f_S}&\overleftarrow{f_N}&\breve{f}_0\end{pmatrix}^t,\\\overrightarrow{F}(r,t)&=\begin{pmatrix}\overrightarrow{f_E}&\overrightarrow{f_W}&\overrightarrow{f_S}&\overrightarrow{f_N}&\breve{f}_0\end{pmatrix}^t.\end{aligned}\quad(5)$$

In this equation, the r and t variables in flows are omitted for the sake of clarity. To make $\Psi(r,t)$ the exact solution of (1), the local scattering matrix is defined by

$$\Sigma(r)=\frac{1}{2n_r^2}\cdot\begin{pmatrix}1&\alpha_r&1&1&Y_r\\\alpha_r&1&1&1&Y_r\\1&1&1&\alpha_r&Y_r\\1&1&\alpha_r&1&Y_r\\1&1&1&1&\beta_r\end{pmatrix},\quad(6)$$

where $\alpha_r=1-2n_r^2$; $\beta r=2n_r^2-4$.

In [25], Eq. (4) is solved by a cellular automaton and the instantaneous electric field is computed with (2). As pointed out by Luthi himself [26] this approach is a generalization of the usual TLM method used for electronic circuits and antennas design [30]. The main drawback of Parflow is that it is slow. Indeed, a high computational time is required 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 to avoid simulation artifacts; e.g. in $2D$, we have [26]

$$dr=c_0\sqrt{2}\cdot dt,\quad(7a)$$

$$dr\ll\lambda.\quad(7b)$$

A resolution $dr=\lambda/6$ is suggested in [26] as a good trade-off value. Moreover, the number of iterations should be at least equal to several times the digital size of the simulated area in order to take multi-path into account. However, the ParFlow capability for modeling natural propagation compensates for its high computational load. Moreover, the algorithm allowing to implement this approach is very simple.

B. Frequency domain formulation

The transposition of TDPF into the frequency domain simply relies on a Fourier transform of the local equation (4) leading to

$$\overrightarrow{F}(r,\nu)=\Sigma(r,\nu)\cdot\overleftarrow{F}(r,\nu)+\overrightarrow{S}(r,\nu),\quad(8)$$

where $\Sigma(r,\nu)=\Sigma(r)\cdot e^{-j2\pi\nu dt}$ and, for notational simplicity, we attain the same notation for the frequency transformed flows.

As shown below, the set of local scattering equations (8) for all pixels amounts to an inverse linear problem when considering only the carrier frequency ν_0, thus providing the narrow band response. This would be inappropriate for time-dispersive environments. However, 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 most of the time. The radio channel is therefore assumed not to be time dispersive. However, this important limit may be relaxed by computing the propagation at different frequencies regularly spaced around the carrier. Based on these predictions, the time response of the channel may be obtained by an inverse Fourier transform. Although certainly of great interest to other type of systems and more dispersive channels, this is out of the scope of this paper. Note furthermore that in the time-domain, Chopard *et. al.* [25] have also used harmonic sources to reduce the complexity arising with a pulsed excitation.

An important consequence of the FD formulation is that the local inner flows can be removed from the formulation. The proof is easily obtained when (8) is written as

$$\begin{pmatrix} \overrightarrow{F}_b(r) \\ \breve{F}(r) \end{pmatrix} = \Sigma(r) \cdot \begin{pmatrix} \overleftarrow{F}_b(r) \\ \breve{F}(r) \end{pmatrix} + \begin{pmatrix} \overrightarrow{S}_b(r) \\ 0 \end{pmatrix}, \quad (9)$$

where border flows and inner flows are respectively given by

$$\overleftarrow{F}_b(r) = \begin{pmatrix} \overleftarrow{f_E}(r) \\ \overleftarrow{f_W}(r) \\ \overleftarrow{f_S}(r) \\ \overleftarrow{f_N}(r) \end{pmatrix}, \quad \overrightarrow{F}_b(r) = \begin{pmatrix} \overrightarrow{f_E}(r) \\ \overrightarrow{f_W}(r) \\ \overrightarrow{f_S}(r) \\ \overrightarrow{f_N}(r) \end{pmatrix}, \quad (10)$$

and

$$\breve{F}(r) = \breve{f}_0(r). \quad (11)$$

In these equations and below, the variable ν_0 has been removed for the sake of clarity.

Note that the inner flow, $\breve{F}(r)$, is not connected to any other pixel and it is used in one local equation only. This local equation can thus be solved with respect to it, leading to the reduced FD formulation (see details in the Appendix):

$$\overrightarrow{F}_b(r) = \Sigma_0(r) \cdot \overleftarrow{F}_b(r) + \overrightarrow{S}_b(r), \quad (12)$$

where $\Sigma_0(r)$ is detailed in the Appendix as a function of $\Sigma(r)$.

To derive the frequency domain algorithm, let (12) be written by using a global formulation concatenating all flows into a unique vector $\underline{F}$, leading to a linear system:

$$\underline{F} = \underline{\Omega}_0 \cdot \underline{F} + \underline{S}, \quad (13)$$

where $\underline{\Omega}_0$ is the propagation matrix involving both the local scattering and neighborhood relationships. Equation (13) leads to a linear inverse problem given by

$$(I_d - \underline{\Omega}_0) \cdot \underline{F} = \underline{S}. \quad (14)$$

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 matrix of $(5 \cdot 10^6)^2$ elements. Therefore, the development of a fast dedicated algorithm is very challenging. A conjugate gradient or other usual iterative techniques may be used, exploiting the sparse nature of the propagation matrix. A more efficient approach exploits the form of $(I_d - \underline{\Omega}_0)$ in (14) seen as the inverse of the sum of a matrix geometric series, leading to

$$\underline{F} = \sum_{k=0}^{\infty} (\underline{\Omega}_0)^k \cdot \underline{S} = \underline{S} + \underline{\Omega}_0 \cdot \underline{S} + (\underline{\Omega}_0)^2 \cdot \underline{S} + \dots \quad (15)$$

For the purpose of computational efficiency, this equation is solved by the use of the local scattering algorithm:

$$\begin{aligned} \overrightarrow{F}_k(r) &= \Sigma_0(r) \cdot \overleftarrow{F}_{k-1}(r) + \overrightarrow{S}(r), \\ \overrightarrow{F}(r) &= \sum_k \overrightarrow{F}_k(r), \end{aligned} \quad (16)$$

where $\overleftarrow{F}_{k-1}(r)$ is updated according to (3).

Note that there are only a few differences compared to the initial TDPF algorithm; for both, computational optimizations are possible. For instance, the outgoing flows can only be computed for pixels having *enough* incoming energy. It is furthermore possible with FDPF to keep and accumulate in memory non-propagated flows until they accumulate enough energy.

III. Multi-resolution (MR) approach

The proposed multi-resolution frequency domain ParFlow (MR-FDPF) algorithm is described in this section. MR-FDPF provides an efficient way for solving (with an exact solution) the FDPF linear system given in (14) by recursively exploiting the concept of inner and border flows outlined in the previous section. This section is organized as follows. Section III-A defines the concept of multi-resolution nodes (MR-node) with their own flows and scattering matrix. Section III-B defines the concept of child MR-node and the relationships between the scattering matrix of a node and those of its children are derived. Section III-C describes how a binary tree is built with a recursive division mechanism starting from the largest MR-Node (referred to as the *head node*) and ending at pixel level.

A. MR-node definition

The MR-FDPF algorithm is based on the block concept derived in [31]. A MR-node (see Fig.2) is defined as a rectangular set of pixels, defined by its size (Δ_x, Δ_y) and the position of its top-left hand corner pixel (p_x, p_y). The entire environment is thus divided into K MR-nodes, referred to as b_k in the following. The flows connecting two pixels belonging to the same MR-node are called inner flows. The flows connecting two pixels belonging to two adjacent MR-nodes are called border flows. From an MR-node, the border flows bringing energy to outside are called outward flows (as illustrated in Fig.2) while those importing energy from outside are called inward flows. Note of course that outward flows of a given MR-node are inward flows for its neighbors.

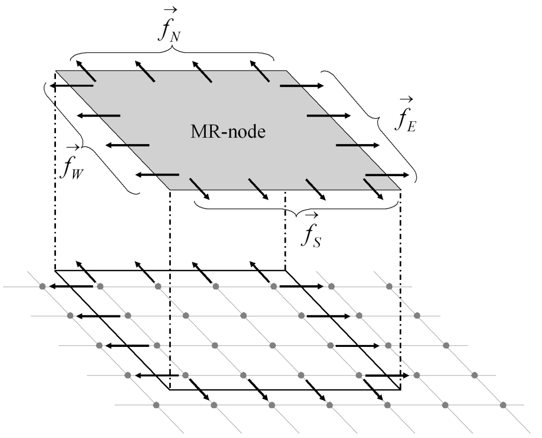


Fig. 2. A MR-node is defined as a 2D rectangular set of unitary nodes. The flows which connect two nodes inside the MR-node are inner flows. The flows which connect an internal node with an external node are border flows. Among the border flows, outward flows (herein illustrated) are those bringing energy to the outside, while inward flows are those importing energy from the outside.

The inward and outward flow vectors are given by

$$\overleftarrow{F}_b(b_k) = \begin{pmatrix} \overleftarrow{f_E}(b_k) \\ \overleftarrow{f_W}(b_k) \\ \overleftarrow{f_S}(b_k) \\ \overleftarrow{f_N}(b_k) \end{pmatrix}, \quad \overrightarrow{F}_b(b_k) = \begin{pmatrix} \overrightarrow{f_E}(b_k) \\ \overrightarrow{f_W}(b_k) \\ \overrightarrow{f_S}(b_k) \\ \overrightarrow{f_N}(b_k) \end{pmatrix}. \tag{17}$$

Let us now define the inner flow vector $\breve{F}(b_k)$ as the vector including all the inner flows for an MR-node b_k. It is straightforward to show that gathering all the local equations (12) of the pixels belonging to b_k leads to a MR-node scattering equation similar to (9) as:

$$\begin{pmatrix} \overrightarrow{F}_b(k) \\ \breve{F}(k) \end{pmatrix} = \Sigma(k) \cdot \begin{pmatrix} \overleftarrow{F}_b(k) \\ \breve{F}(k) \end{pmatrix} + \begin{pmatrix} \overrightarrow{S}_b(k) \\ \overset{o}{S}(k) \end{pmatrix}, \tag{18}$$

where herein and later k stands for b_k for the sake of clarity.

As done previously with the classical FD formulation, the inner flows, which are used in only one local equation, can be removed from the formulation. The equivalent scattering matrix is obtained by solving (18) with respect to $\breve{F}(k)$ as detailed in the Appendix, leading to the local MR-node equation:

$$\overrightarrow{F}_b(k) = \Sigma_b(k) \cdot \overleftarrow{F}_b(k) + \overrightarrow{S}_b(k), \tag{19}$$

where $\Sigma_b(k)$ is the scattering matrix involving *border* flows only. The relationship between $\Sigma_b(k)$ and $\Sigma(k)$ is provided in the Appendix.

It should be noted that Shlepnev has proposed in [32] a similar concept based on the TLM formalism. In his work, Shlepnev has studied the wave propagation inside rectangular bricks using a continuous formulation providing the scattering equations. We show, however, in the next section how we use the discrete formulation to derive a multi-resolution structure leading to a very efficient algorithm.

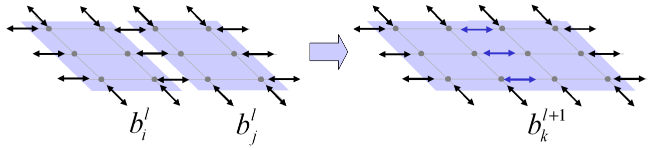


Fig. 3. A father node is obtained from two child nodes. The exchange flows between both child nodes are the inner flows for the father one (blue arrows).

B. Children MR-nodes

From the previous definition of MR-nodes, a forthcoming question concerns the choice of the MR-node size. Complexity indeed increases with *i)* the MR-nodes' size and *ii)* the number of MR-nodes needed to encompass the whole environment. As a matter of fact, the use of larger MR-nodes reduces the number of nodes but in turns increases the size of local scattering matrices. Instead of trying to find an optimal MR-node size, we herein develop a recursion embedding small MR-nodes into larger ones.

As a starting point of the demonstration, let the environment be divided into MR-nodes of a fixed given size. These MR-nodes are referred to as (ℓ)-level MR-nodes, or simply (ℓ)-nodes. It is assumed that the scattering matrices of these nodes are known and the propagation can thus be computed over the whole space with the recursive algorithm (16) using(19) for each MR-node $b_k^{(\ell)}$.

However the final steady-state can be computed more efficiently at level $(\ell+1)$. Let (ℓ)-nodes be merged to provide $(\ell+1)$-nodes. Each $(\ell+1)$-node $b_k^{(\ell+1)}$ is thus made of two adjacent child (ℓ)-nodes $b_i^{(\ell)}$ and $b_j^{(\ell)}$, as illustrated in Fig.3.

To solve the (ℓ)-level equation at level $(\ell+1)$, the results of the Appendix can also be used. Let us first gather the scattering equations (19) associated with both child nodes $b_i^{(\ell)}$ and $b_j^{(\ell)}$ into a unique equation:

$$\begin{pmatrix} \overrightarrow{F}_b(i) \\ \overrightarrow{F}_b(j) \end{pmatrix} = \begin{pmatrix} \Sigma(i) & 0 \\ 0 & \Sigma(j) \end{pmatrix} \cdot \begin{pmatrix} \overleftarrow{F}_b(i) \\ \overleftarrow{F}_b(j) \end{pmatrix} + \begin{pmatrix} \overrightarrow{S}_b(i) \\ \overrightarrow{S}_b(j) \end{pmatrix}. \tag{20}$$

It constitutes the initial equation associated with the father node $b_k^{(\ell+1)}$. The border flows of $b_i^{(\ell)}$ and $b_j^{(\ell)}$ belong to $b_k^{(\ell+1)}$ as either inner or border flows. More precisely, the flows connecting $b_i^{(\ell)}$ with $b_j^{(\ell)}$ are considered as inner flows for $b_k^{(\ell+1)}$ since they are not located at the border of $b_k^{(\ell+1)}$, as illustrated in Fig.3. In the case of an horizontal gathering, the exact relationships between (ℓ)- and $(\ell+1)$-level flows are easily derived. The inward flows relationships are

$$\begin{aligned} &\overleftarrow{f_E}(k) = \overleftarrow{f_E}(i), \quad \overleftarrow{f_W}(k) = \overleftarrow{f_W}(j), \\ &\overleftarrow{f_S}(k) = \begin{pmatrix} \overleftarrow{f_S}(i) \\ \overleftarrow{f_S}(j) \end{pmatrix}, \quad \overleftarrow{f_N}(k) = \begin{pmatrix} \overleftarrow{f_N}(i) \\ \overleftarrow{f_N}(j) \end{pmatrix}, \end{aligned} \tag{21}$$

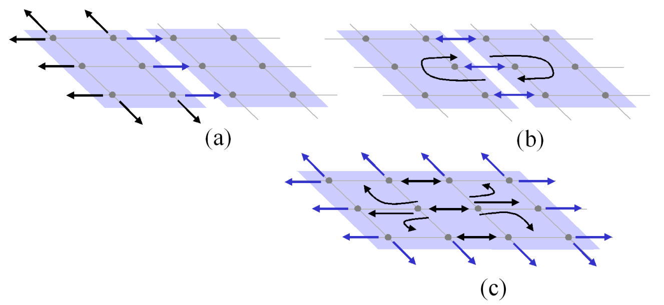


Fig. 4. Bottom-up phase: (a) the source node i is gathered with node j into node k. The inner steady-state (b) and the new source flows (c) are computed.

while the outward flows relationships are

$$\overrightarrow{f}_E(k) = \overrightarrow{f}_E(j), \quad \overrightarrow{f}_W(k) = \overrightarrow{f}_W(i),$$
$$\overrightarrow{f}_S(k) = \begin{pmatrix} \overrightarrow{f}_S(i) \\ \overrightarrow{f}_S(j) \end{pmatrix}, \quad \overrightarrow{f}_N(k) = \begin{pmatrix} \overrightarrow{f}_N(i) \\ \overrightarrow{f}_N(j) \end{pmatrix}. \tag{22}$$

The inner flows $\breve{F}(k)$ associated with $b_k^{\ell+1}$ are the flows connected to both child nodes, as

$$\breve{F}(k) = \begin{pmatrix} \overleftarrow{f}_W(i) \\ \overleftarrow{f}_E(j) \end{pmatrix} = \begin{pmatrix} \overrightarrow{f}_W(j) \\ \overrightarrow{f}_E(i) \end{pmatrix}. \tag{23}$$

By use of (21)-(23), Eq. (20) expands as

$$\begin{pmatrix} \overrightarrow{F}_b(k) \\ \breve{F}(k) \end{pmatrix} = \begin{pmatrix} \Sigma_{ee}(k) \, \Sigma_{ei}(k) \\ \Sigma_{ie}(k) \, \Sigma_{ii}(k) \end{pmatrix} \cdot \begin{pmatrix} \overleftarrow{F}_b(k) \\ \breve{F}(k) \end{pmatrix} + \begin{pmatrix} \overrightarrow{S}_{ex}(k) \\ \overrightarrow{S}_0(k) \end{pmatrix}, \tag{24}$$

which can still be solved with respect to $\breve{F}(k)$ according to the Appendix, providing the exact equivalence between the (ℓ)-level and the $(\ell+1)$-level formulations.

C. The multi-resolution algorithm: MR-FDPF

The MR-FDPF algorithm is derived directly from these inter-level relationships. The first task is to build a binary tree by successively gathering the MR-nodes, starting from the pixels and ending at the head node, i.e. the MR-node encompassing the whole environment. The manner this binary tree can be efficiently built is discussed in the next Section IV-A. Let us now assume that this binary tree exists and has L levels. Thus, having the inter-level relationships defined, the MR-FDPF algorithm is obtained by applying recursively these relationships in two phases.

1) Bottom-up phase: The bottom-up phase aims at computing the $(\ell+1)$-level formulation from the (ℓ)-level one, starting with $\ell = 1$ and ending with $\ell = L$. When a source node is associated with a neighbor providing a father source node, two tasks are needed, as illustrated in Fig.4:

- *inner steady-state computation:* as flows are exchanged between both child nodes, an inner steady-state can be computed using the second right-hand term in (44), leading to
$$\breve{F}(k) = I(k) \cdot \overrightarrow{S}_0(k), \tag{25}$$
where
$$I(k) = \left(Id - \Sigma_{ii}(k)\right)^{-1}. \tag{26}$$
is called the *inner matrix* and has to be computed prior to the propagation for each potential source node. Note that $\Sigma_{ii}(k)$ reflects the flow exchange between both child nodes. The computation of this matrix is discussed below in Section III-C.3.
- *equivalent source flows computation:* to transform the father node into a source node, the outward source flows have to be determined. Introducing (26) into (47) provides
$$\overrightarrow{S}_b(k) = \overrightarrow{S}_{ex}(k) + U(k) \cdot \breve{F}(k), \tag{27}$$
where $U(k)$ denotes the *upward matrix* associated with node k and it is given by
$$U(k) = \Sigma_{ei}(k). \tag{28}$$
$\Sigma_{ei}(k)$ projects the steady-state inner flows towards border flows.

The recursion ends when the head-node is reached, having its inner flows computed. At each level, the steady-state inner flows $\breve{F}(b_s^\ell)$ associated with the source node are stored.

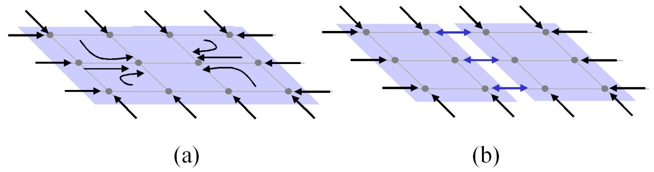


Fig. 5. Top-down phase: (a) The inward flows of the father node import energy toward inner flows. (b) Steady state inner flows are computed leading to inward flows of each child node.

2) Top-down phase: The top-down phase is also a recursion but starting at the head-node (level L) down to level 0. For each node, at each level, the steady-state inner flows are computed as a function of inward flows according to (44) (see the Appendix); this is illustrated in Fig. 5. Let the head node be referred to as b_0^L. L stands for the last level, and 0 is the index of the unique node of L^{th} level. At the beginning, inward flows $\overleftarrow{F}_b(b_0^L)$ associated with the head node are set according to the boundary conditions. Indeed, $\overleftarrow{F}_b(b_0^L)$ involves the flows bringing energy from outside. To avoid boundary artifacts and spurious reflections, the environment is surrounded by a fake 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 of the head-node computed during the bottom-up phase are thus unchanged and have to be propagated towards the child nodes.

The computation is made recursively at each node based on the values of the already processed inward flows. Then, the inner flows are computed according to (44) :

$$\breve{F}(k) = \begin{cases} I(k) \cdot D(k) \cdot \overleftarrow{F}_b(k) & \text{for } k \neq s \,, \\ I(k) \cdot D(k) \cdot \overleftarrow{F}_b(k) + \breve{F}(k) & \text{for } k = s \,, \end{cases} \tag{29}$$

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

$$D(k) = \Sigma_{ie}(k). \tag{30}$$

At the end of the recursion, the ParFlow linear system is solved exactly, with no approximation, except for those due to the numerical accuracy.

3) Preprocessing phase: In this approach, three propagation matrices are associated with each node (Upward, Downward and Inner matrices). Because computing them does not require the knowledge of the source position, this can be handled separately in a preprocessing phase. This further allows to reduce the overall computational load 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. The Upward and Downward matrices are directly related to the exchange scattering matrix of child nodes according to (28) and (30). The computational load for these matrices is thus not significant.

The main computational load is due to the computation of the exchange flows scattering matrix of each node which is required by its father node to compute its own propagation matrices. The exchange flows scattering matrix is given by (46) as a function of the scattering matrices of the 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 involves the computation of $I(k)$ according to (26). An efficient computation of this matrix is proposed in [33]. The MR-FDPF algorithm is finally obtained according to

Algorithm 1

Preprocessing:
- For $\ell = 1$ to $L-1$, Do
  - *Compute* $\forall k$; $\Sigma(b_k^\ell)$, $I(b_k^\ell)$, $U(b_k^\ell)$, $D(b_k^\ell)$

Initialization:
- *Set* $\forall \ell$; $\forall k$; $\breve{F}(b_k^\ell) = 0$

Upward phase:
- For $\ell = 1$ to $L-1$, Do
  - Update source flows: $S_b(b_s^{\ell-1}) \Rightarrow S_0(b_s^\ell)$; $S_{ex}(b_s^\ell)$
  - Compute inner flows: $\breve{F}(b_s^\ell) = I(b_s^\ell) \cdot S_0(b_s^\ell)$
  - Compute source flows: $S_b(b_s^\ell) = S_{ex}(b_s^\ell) + U(b_s^\ell) \cdot \breve{F}(b_s^\ell)$

Downward phase:
- For $\ell = L-1$ down to 0, Do
- For $k = 0$ to $K_\ell - 1$, Do
  - *Compute:* $\breve{F}(b_k^\ell) = \breve{F}(b_k^\ell) + I(b_k^\ell) \cdot D(b_k^\ell) \cdot \overleftarrow{F}_b(b_k^\ell)$
  - *Update child inward flows:* $\overleftarrow{F}(b_k^\ell) \Rightarrow \overleftarrow{F}_b(b_i^{\ell-1})$; $\overleftarrow{F}_b(b_j^{\ell-1})$

IV. Implementation

The general formulation of the MR-FDPF algorithm has been detailed in the previous section. We detail in this section implementation aspects of the method. Section IV-A presents the adaptive multi-resolution (MR)-FDPF algorithm which modifies the regular multi-resolution tree described above to save computational and memory loads. This algorithm is said to be 'adaptive' as the tree is built to fit to the geometry of the environment. Next section IV-B proposes a complexity study of the algorithm and describes why the frequency domain algorithm outperforms the more usual time-domain formulation.

A. Adaptive binary tree

A regular binary tree is the easiest one that can be built but is not optimal. The efficiency is considered with respect to four criteria:

- The computational load of the pre-processing step dedicated to computing the matrices Σ and I associated with each MR-node.
- The memory needed to store all matrices.
- The computational load needed to compute the coverage of a source over the whole environment at the pixel resolution.
- The computational load 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. The mean received power can be evaluated directly from the inward flows according to

$$P(b_i) = \frac{\|\overleftarrow{F}(b_i)\|^2}{2 \cdot (N_x + N_y)}. \tag{31}$$

The computational load is thus minimized because the downward propagation is stopped in each branch as soon as a homogeneous node is reached. The variability of predictions due to short-time fading is also reduced since the received power is spatially averaged.

Regarding the preprocessing step, it is obvious that the main computational load and memory consumption are due to the scattering matrices calculus. The case of two identical MR-nodes is interesting. Two MR-nodes are said to be identical if they have the same size and identical child nodes. In this case, both have identical matrices and a unique reference model can be computed and stored for both. Such a model is called a node's type. The node's type refers itself to its two children's types (to compute the scattering matrices) and to its scattering matrix. The aim of the preprocessing phase is twofold: building the binary tree (very fast), and building the database of MR-node types.

For building the binary tree, a top-down approach is used. Starting from the head node, this approach recursively splits MR-nodes into pairs of child MR-nodes. Several empirical algorithms may be used therefore. A first approach would aim at cutting each MR-node along a line in the middle of its higher length. Another one would be to align cuts in MR-nodes along main walls such as obtaining homogeneous MR-nodes as soon as possible (see Fig. 6). The following rule may be 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; $\forall i \in [1; N-1]$.
- Split the block at the index i_m to maximize $D(i)$.

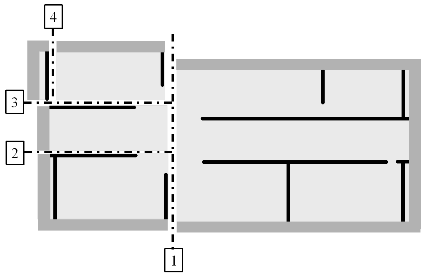


Fig. 6. The test environment is made up of free space (light grey), walls (black) and surrounding absorbing material (dark grey). The top-down dividing procedure is illustrated for the fourth first levels on one branch: the head-node is firstly divided along line 1, then the left-hand child along line 2, the top-child along line 3, and finally, left-hand child along line 4.

Such an approach based on pure discontinuity leads to a large computational overhead when cutting a large node near the border. A trade-off between both approaches is thus proposed with

$$i_m = \arg\max_{i\in[1;N-1]} \left\{D(i)\cdot C(|i-h_0|)\right\}, \tag{32}$$

and h_0 being the half-length of the MR-node and $C(i)$ a monotonic increasing function.

B. Complexity study

The preprocessing step is the most complex as it computes the local scattering matrices. Fortunately, the preprocessing is done only once. The upward step is very fast since it only concerns source nodes. The downward step costs more as being computed for each node of each level. For the sake of simplicity, a square environment is only considered in this complexity study, i.e. $N_y = N_x = 2^Q$. It should be noted, however, that the adaptive tree allows to process any rectangular environment.

1) Preprocessing load: The main computational load is devoted to the calculus of $I(b_k^\ell)$ according to (26) and of $\Sigma(b_k^\ell)$ according to (46). Taking the number of nodes into account, the computational load of each level is given by

$$\mathcal{C}(prep,\ell) \propto \begin{cases} O\left(19\cdot N_x^2\cdot 2^q\right) & \text{if } \ell=2q\ , \\ O\left(27\cdot N_x^2\cdot 2^{q-1}\right) & \text{if } \ell=2q+1\ . \end{cases} \tag{33}$$

The whole pre-processing load is thus given by

$$\mathcal{C}(prep) \propto O\left(52\cdot N_x^3\right). \tag{34}$$

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

$$\mathcal{M}(b_k^\ell) = 13\cdot 2^\ell\cdot mem, \tag{35}$$

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

The memory associated with each level is a constant, given by

$$\mathcal{M}(\ell) = 13\cdot N_x^2\cdot mem. \tag{36}$$

The total memory then equals

$$\mathcal{M}(prep) = 26\cdot log_2(N_x)\cdot N_x^2\cdot mem. \tag{37}$$

Both estimations are in fact upper-bounds because they are obtained when considering each MR-node independently. The use of node's types defined above improves the computational time. A pure free-space would be the most favorable case because all pixels would be identical and thus at each (ℓ)-level, only one scattering matrix should be computedproviding lower bounds. We thus have:

$$\begin{aligned} O\left(26\cdot N_x^2\right) &< \mathcal{M}(prep)/mem < O\left(26\cdot log_2(N_x)\cdot N_x^2\right) \\ O\left(34\cdot N_x^3\right) &< \quad \mathcal{C}(prep) \quad < O\left(52\cdot N_x^3\right). \end{aligned} \tag{38}$$

Note that the head-node (last level) consumes itself about 50% of the computational load.

2) Upward load: Since the upward phase concerns only source nodes, the associated computational load is negligible and given by

$$\mathcal{C}(up) \propto O\left(3\cdot N_x^2\right), \tag{39}$$

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

3) Downward load: 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

$$\mathcal{C}(down,\ell) \propto \begin{cases} O\left(6\cdot N_x^2\right) & \text{if } \ell=2q\ , \\ O\left(4\cdot N_x^2\right) & \text{if } \ell=2q+1\ . \end{cases} \tag{40}$$

The whole computational load is

$$\mathcal{C}(down) \propto O\left(10\cdot log_2(N_x)\cdot N_x^2\right). \tag{41}$$

4) Standard TDPF load: Since the time-domain algorithm is iterative, the exact computational load is difficult to assess and depends on the desired accuracy. An estimation can be found by approximating the number of iterations to *few* times the environment size (i.e. $N_{it} = k\cdot N_x(\ell)$). The parameter k should be large enough to simulate multiple reflected waves. Of course, the value of k depends on the expected accuracy on one hand, and on the loss factor for the obstacles on the other hand. A reference computational load is then estimated as

$$\mathcal{C}(ref) \propto O\left(16\cdot k\cdot N_x^3\right). \tag{42}$$

This computational load is on the same order of magnitude as that of the preprocessing phase of MR-FDPF. However the computational load for computing the coverage of a source is much lower with MR-FDPF, being in $O(\log_2(N_x)\cdot N_x^2)$. It means that after preprocessing, the exact steady-state coverage is obtained within a computational load equal to a few TDPF iterations only.

5) Other algorithms: The complexity of the simple 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 to be more consuming than the later, but in fact depends on the number of walls to be processed. 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 outline exhibits that the

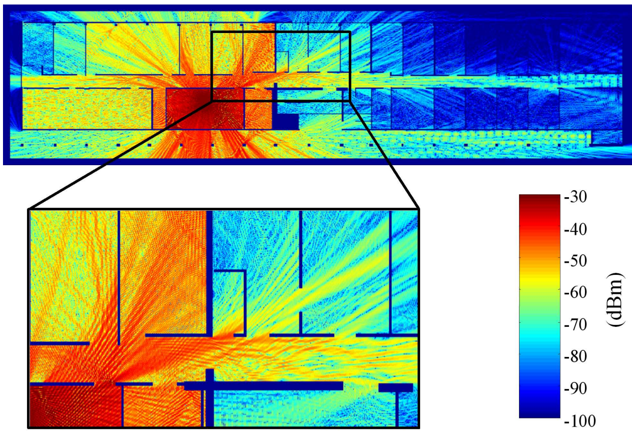


Fig. 7. Field strength prediction for a 2.4GHz source at a resolution of 2cm and a detailed area. Different effects such as reflection, diffraction, stationary waves, corridor effect, are shown.

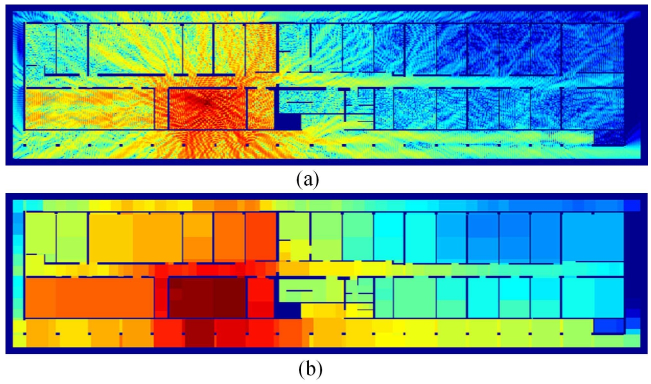


Fig. 8. Field strength prediction with an intermediate frequency of 480MHz, at a resolution of 10cm: full resolution in (a) and homogeneous nodes resolution in (b). The color scale of Fig.7 is still used.

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 depends on the number of rays and reflections and not directly on the environment size. Ray-tracing is probably the fastest approach if only a few receiving points are requested, but MR-FDPF certainly outperforms ray-tracing for a full resolution study and when the environment contains many obstacles. Indeed, ray-tracing is known to be much more time consuming than the MWM approach in this case [1], [5], [16].

V. Results and Discussion

A. 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. Fig.7 shows the resulting field strength prediction at a frequency of $2.4GHz$, with a resolution of $dr = 2cm$. The preprocessing lasted $30min$ while the propagation lasted only $45s$. As shown in Fig.7, the narrow-band channel properties can be studied in depth. The local analysis 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, etc.). DOA algorithms could be furthermore applied to predict the local channel angle spreading in each room.

An attractive field of application concerns networking simulations. For instance, the current craze for *ad hoc* networks calls for accurate simulations of indoor propagation [34], [35]. MR-FDPF should be a good candidate because it allows to quickly compute realistic radio links between nodes moving in the environment. Another interesting application concerns the simulation of multi-antennas mobile systems [36]. The accurate nature of MR-FDPF based simulations permits the computation of local field variations and therefore can easily support MIMO channel simulations.

B. wLAN planning

Many efforts have been recently devoted to the development of efficient propagation tools in the context of wLAN planning [37]. As discussed in the introduction, the standard approaches face three challenging problems:

- Numerous diffractions and reflections make empirical approaches not efficient and increase 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 addresses all of the above mentioned problems. Firstly, the computational time does not depend on the number of reflections. Secondly, all diffractions and reflections are taken into account, as the inverse steady-state problem is solved exactly. Finally, the most time-consuming phase, i.e., the preprocessing, is done just once for all the possible sources, allowing to test efficiently many configurations. Nevertheless, the fine resolution requirement still leads to a computational time higher than expected. To further reduce it, we propose the use of a simulation frequency lower than the true frequency. This assumption was already introduced by Chopard *et. al.* [25] 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. Therefore, a simulation frequency of 480MHz is chosen when a resolution step of $10cm$ is targeted (7).

Fig.8(a) illustrates 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 some approximations: the use of the intermediate frequency, the 2D approximation, the lack of knowledge about furniture, peoples, and the walls' 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 the measurement procedure are out of the scope of this paper. The root mean square error (RMSE) was found to be of about $5dB$ as described in [38]. The resulting coverage map is

TABLE I
COMPUTATIONAL AND MEMORY LOADS

	method		
	regular	disc.	optimal
mem (MO)	57.8	86.9	63.4
CPU time (s)	8.1	27	8.6
nb of nodes	1009	218	226
propag hom (ms)	361	288	203
propag pix (ms)	1650	1770	1640

provided in Fig.8, for both the fine (a) and the homogeneous node (b) resolutions.

Computational and memory loads are summarized in Table I for three binary trees: a regular one, a full discontinuity-based tree, and the optimal trade-off between both approaches.

VI. CONCLUSIONS

In this paper, a novel method called MR-FDPF has been proposed. This method is based on the ParFlow formalism introduced by Luthi and Chopard [25], [26]. In their work, a general flow equation modeling has been derived, which is equivalent to the well-known TLM (Transmission Line Matrix) [28]–[30] model in the context of electromagnetism. In this paper, the ParFlow formalism is developed in the frequency domain, leading to a large linear system. This large linear system is solved herein by means of an original approach exploiting the specific structure of the scattering matrix connecting the flows. In a first step, a MR-node is defined as a generalization of the usual TLM node. In a second step, recursive relationships between MR-nodes are derived using a child-father ties. Finally A recursive splitting method is proposed, starting from the head-node, and providing a multi-resolution structure: the binary tree.

With this approach, the large FDPF linear system is solved without approximations by means of the MR-FDPF algorithm, with a computational load equal to a few iterations of the initial TDPF algorithm. This is a significant improvement, leading to a 2D discrete approach that allows to compute a coverage of more than $1000m^2$ in a few hundreds of milliseconds. It should be pointed out that MR-FDPF takes into account all reflections and all diffractions. The counterpart, which is the narrow-band estimation, may be overcome by estimating multiple spectrum lines.

Although MR-FDPF appears promising, further experimental investigations need to be done to validate the simulations for other environments. Future theoretical works will focus on optimization of matrix based operations in the preprocessing phase in order to permit the 3D generalization of such a method. Indeed, although the theoretical development of the 3D approach is immediate, numerical constraints are difficult to be dealt with and computational and memory requirements increase drastically for this case.

APPENDIX
REMOVING INNER FLOWS

We show herein how to remove the inner flows. To this end, remember that in this paper the formulation of the local scattering is written in terms of inner and exchange variables. Such a formulation allows to remove the inner variables by partially solving the system. A general formulation is given by

$$\begin{pmatrix} \overrightarrow{F}_b(x) \\ \breve{F}(x) \end{pmatrix} = \begin{pmatrix} \Sigma_{ee}(x) & \Sigma_{ei}(x) \\ \Sigma_{ie}(x) & \Sigma_{ii}(x) \end{pmatrix} \cdot \begin{pmatrix} \overleftarrow{F}_b(x) \\ \breve{F}(x) \end{pmatrix} + \begin{pmatrix} \overrightarrow{S}_{ex}(x) \\ \overrightarrow{S}_0(x) \end{pmatrix}, \tag{43}$$

where x stands either for r in section II, or for k in section III. $\overleftarrow{F}_b(x)$ and $\overrightarrow{F}_b(x)$ refer to the exchange variables. While $\overleftarrow{F}_b(x)$ imports energy inside the node, $\overrightarrow{F}_b(x)$ exports energy towards the neighboring nodes. $\breve{F}(x)$ is an inner variable which is not used elsewhere. The scattering matrix $\Sigma(x)$ is herein divided into four blocks, where subscripts e and i stand respectively for *exchange* and *inner*.

Because $\breve{F}(x)$ is not used elsewhere, the system can be solved with respect to this variable, providing the following results:

$$\breve{F}(x) = (Id - \Sigma_{ii}(x))^{-1} \cdot \left(\Sigma_{ie}(x) \cdot \overleftarrow{F}_b(x) + \overrightarrow{S}_0(x) \right). \tag{44}$$

The outward flows are then obtained according to

$$\overrightarrow{F}_b(x) = \Sigma_b(x) \cdot \overleftarrow{F}_b(x) + \overrightarrow{S}_b(x), \tag{45}$$

with

$$\Sigma_b(x) = \Sigma_{ei}(x) \cdot (Id - \Sigma_{ii}(x))^{-1} \cdot \Sigma_{ie}(x), \tag{46}$$

and

$$\overrightarrow{S}_b(x) = \overrightarrow{S}_{ex}(x) + \Sigma_{ei}(x) \cdot (Id - \Sigma_{ii}(x))^{-1} \cdot \overrightarrow{S}_0(x). \tag{47}$$

ACKNOWLEDGMENT

The authors wish to thanks Dr Cristina Comaniciu and Dr Mischa Dohler for their helpful comments and suggestions.

REFERENCES

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

[2] J. McKnown and R. Hamilton, "Ray tracing as design tool for radio networks," *IEEE Network*, vol. 5, pp. 27–30, November 1991.

[3] S. Seidel and T. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings." *IEEE Trans. Antennas Propag.*, vol. 40, no. 2, pp. 207–217, 1992.

[4] W. Honcharenko, H. Bertoni, L. Dailing, J. Qian, and H. Yee, "Mechanisms governing uhf propagation on single floors in modern office buildings," *IEEE Trans. Antennas Propag.*, vol. 41, no. 6, pp. 787–790, 1993.

[5] 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*, pp. 2–12, October 2000.

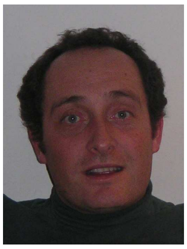
[6] K.-W. Cheung, J. H.-M. Sau, and R. Murch, "A new empirical model for indoor propagation prediction," *IEEE Trans Veh. Technol.*, vol. 47, no. 3, pp. 996–1001, August 1998.

[7] S. Y. Seidel, K. R. Schaubach, T. Tran, and T. Rappaport, "Research in site-specific propagation modeling for PCS system design." in *Proc. of the 43rd IEEE Veh. Technol. conf.*, Secaucus, NJ, USA, May 1993, pp. 261–264.

[8] S. Seidel and T. Rappaport, "Site specific propagation prediction for wireless in-building personal communication system design." *IEEE Trans Veh. Technol.*, vol. 43, no. 4, pp. 879–891, 1994.

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

[10] A. Rajkumar, B. Naylor, F. Feisullin, and L. Rogers, "Predicting RF coverage in large environments using ray-beam tracing and partitioning tree represented geometry," Technical Memorendum, AT&T Bell Laboratories, Tech. Rep., 1995.
[11] S. Fortune, "A beam tracing algorithm for prediction of indoor radio propagation," Tech. Rep., February 1996.
[12] C.-F. Yang, B.-C. Wu, and C.-J. Ko, "A ray-tracing method for modeling indoor wave propagation and penetration," *IEEE Trans. Antennas Propag.*, vol. 46, no. 6, pp. 907–919, June 1998.
[13] H. Suzuki and A. S. Mohan, "Measurement and prediction of high spatial resolution indoor radio channel characteristic map," *IEEE Trans Veh. Technol.*, vol. 49, no. 4, pp. 1321–1333, July 2000.
[14] G. E. Athanasiadou and A. R. Nix, "A novel 3D indoor ray-tracing propagation model: The path generator and evaluation of narrow-band and wide-band predictions," *IEEE Trans Veh. Technol.*, vol. 49, no. 4, pp. 1152–1168, July 2000.
[15] ——, "Investigation into the sensitivity of the power predictions of a microcellular ray tracing propagation model," *IEEE Trans Veh. Technol.*, vol. 49, no. 4, pp. 1140–1151, July 2000.
[16] G. Wolfle, R. Wahl, P. Wildbolz, and P. Wertz, "Dominant path prediction model for indoor and urban scenarios," in *11th COST 273*, Germany, September 2004.
[17] F. A. Agelet and al., "Efficient ray-tracing acceleration techniques for radio propagation modeling," *IEEE Trans Veh. Technol.*, vol. 49, no. 6, pp. 2089–2104, november 2000.
[18] T. Imai and T. Fujii, "Fast algorithm for indoor microcell area prediction system using ray-tracing method," *Electronics and Communications in Japan, part 1*, vol. 85, no. 6, pp. 41–52, 2002.
[19] Z. Ji, B.-H. Li, H.-X. Wang, H.-Y. Chen, and T. K. Sarkar, "Efficient ray-tracing methods for propagation prediction for indoor wireless propagation," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 2, pp. 41–49, April 2001.
[20] Z. Chen, H. L. Bertoni, and A. Delis, "Progressive and approximate techniques in ray-tracing-based radio wave propagation prediction models," *IEEE Trans. Antennas Propag.*, vol. 52, no. 1, pp. 240–251, January 2004.
[21] M. Hassan-Ali and K. Pahlavan, "A new statistical model for site-specific indoor radio propagation prediction based on geometric optics and geometric probability," *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 112–124, January 2002.
[22] L. Talbi, "FDTD characterization of the indoor propagation," *Journal of electromagnetic waves and applications*, vol. 10, no. 2, pp. 243–247, 1996.
[23] J. Lee and A. K. Y. Lai, "FDTD analysis of indoor radio propagation," in *IEEE Antennas and Propag. Society Int. Symposium*, vol. 3, Atlanta, GA, June 1998, pp. 1664–1667.
[24] R. Sato and H. Shirai, "Simplified analysis for indoor propagation of a wlan channel," in *IEEE Topical Conference on Wireless Commun. Tech.*, vol. CDrom, Honolulu, Hawai, USA, October 2003.
[25] B. Chopard, P. Luthi, and J. Wagen, "A lattice boltzmann method for wave propagation in urban microcells," in *IEE Proceedings - Microwaves, Antennas and Propagation*, vol. 144, 1997, pp. 251–255.
[26] P. O. Luthi, "Lattice wave automata : from radiowave to fracture propagation," Ph.D. dissertation, University of Geneva, Geneva, Switzerland, march 1998.
[27] J.-M. Gorce, E. Jullo, and K. Runser, "An adaptative multi-resolution algorithm for 2D simulations of indoor propagation," in *IEE Proc. of the 12th International Conference on Antennas and Propagation*, Exeter, UK, April 2003, pp. 216–219.
[28] W. Hoefer, "The transmission line matrix method - theory and applications," *IEEE Trans. Microw. Theory Tech.*, vol. 33, no. 4, pp. 882–893, 1985.
[29] P. Johns, "A symmetrical condensed node for the TLM method," *IEEE Trans. Microw. Theory Tech.*, vol. 35, pp. 370–377, April 1987.
[30] J. Rebel, "On the foundation of the transmission line matrix method," Ph.D. dissertation, Universitat Munchen, Munchen, Deutchland, dec 1999.
[31] J.-M. Gorce and S. Ubeda, "Propagation simulation with the parflow method : fast computation using a multi-resolution scheme." in *Proc. of the 54th IEEE Veh. Technol. conf.*, Atlantic City, October 2001, pp. 1304–1307.
[32] Y. O. Shlepnev, "Trefftz finite elements for electromagnetics," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 5, pp. 1328–1339, May 2002.
[33] J.-M. Gorce, K. Runser, and G. de la Roche, "The adaptive multi-resolution frequency-domain parflow (MR-FDPF) method for indoor radio wave propagation simulations. Part I : theory and algorithms." RR-5740, INRIA, France, Tech. Rep., June 2005.
[34] M. Takai, R. Bagrodia, K. Tang, and M. Gerla, "Efficient wireless network simulations with detailed propagation models," *Wireless Networks*, vol. 7, pp. 297–305, 2001.
[35] A. P. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Real-world environment models for mobile network evaluation," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 3, pp. 622–632, march 2005.
[36] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, "Ad-hoc networking with directional antennas: a complete system solution," *Wireless Networks*, vol. 23, no. 3, pp. 496–506, march 2005.
[37] K. Jaffrès-Runser, J.-M. Gorce, and S. Ubeda, "Qos constrained wireless lan optimization within a multiobjective framework," *IEEE Wireless Commun.*, december 2006, in press.
[38] K. Runser and J.-M. Gorce, "Assessment of a new indoor propagation prediction model based on a multi-resolution algorithm," in *Proc. of the 61st IEEE Veh. Technol. Conf., spring*, vol. 1, Stockholm, Sweden, May 2005, pp. 35–38.



Jean-Marie Gorce received the Dipl. Ing. (M.Sc.) degree in electrical engineering from the National Institute of Applied Sciences (INSA), Lyon, France, in 1993. He completed his PhD thesis in 1998 on parametric feature extraction from radio-frequency ultrasound signals. After a postdoctoral year at Bracco Research, Geneva, Switzerland, he joined the telecommunications department at INSA Lyon as an associate professor. He is head of the radio modeling axis of CITI Laboratory, and is involved in the ARES project of INRIA Rhone-Alpes, France. His main research field concerns wireless networks focusing on realistic modeling, wireless system optimization, and performance assessment, considering as well architecture-based and ad hoc networks.

Katia Jaffrès-Runser received her MSc. (2002) and PhD. (2005) from the INSA university in Lyon , France. During her PhD, she was a member of the CITI laboratory and of the ARES project team of INRIA. She mainly focused on wireless LANs optimization and Indoor prediction propagation.

In 2006, she joined the Stevens Institute of Technology in Hoboken, NJ, USA as a post-doctoral researcher where she's working with Pr. Comaniciu on wireless ad hoc and sensor networks optimization. She received a 3-year Marie-Curie fellowship from EU to pursue her work in 2007 in both Stevens I.T. and INSA Lyon.

Guillaume de la Roche graduated in telecommunication engineering in 2003 from CPE Lyon and received the MSc. degree from INSA Lyon . He is currently a PhD student in the CITI laboratory of INSA Lyon, under the supervision of Dr. Jean-Marie Gorce. He is working on channel modelling in indoor environments, and his main research is about MR-FDPF algorithm implementation and optimization, especially the 2,5D and 3D extensions.