

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:

2019-03-28

Deposited version:

Post-print

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Acebron, J. A. & Ribeiro, M. A. (2016). A Monte Carlo method for solving the one-dimensional telegraph equations with boundary conditions. *Journal of Computational Physics*. 305, 29-43

Further information on publisher's website:

[10.1016/j.jcp.2015.10.027](https://doi.org/10.1016/j.jcp.2015.10.027)

Publisher's copyright statement:

This is the peer reviewed version of the following article: Acebron, J. A. & Ribeiro, M. A. (2016). A Monte Carlo method for solving the one-dimensional telegraph equations with boundary conditions. *Journal of Computational Physics*. 305, 29-43, which has been published in final form at <https://dx.doi.org/10.1016/j.jcp.2015.10.027>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Accepted Manuscript

A Monte Carlo method for solving the one-dimensional telegraph equations with boundary conditions

Juan A. Acebrón, Marco A. Ribeiro

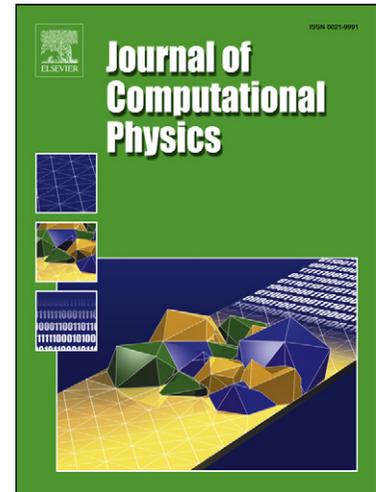
PII: S0021-9991(15)00691-9
DOI: <http://dx.doi.org/10.1016/j.jcp.2015.10.027>
Reference: YJCPH 6186

To appear in: *Journal of Computational Physics*

Received date: 27 February 2015
Revised date: 12 October 2015
Accepted date: 18 October 2015

Please cite this article in press as: J.A. Acebrón, M.A. Ribeiro, A Monte Carlo method for solving the one-dimensional telegraph equations with boundary conditions, *J. Comput. Phys.* (2015), <http://dx.doi.org/10.1016/j.jcp.2015.10.027>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



A Monte Carlo Method for solving the One-Dimensional Telegraph Equations with Boundary Conditions

Juan A. Acebrón^a

^a*Department of Information Science and Technology, ISCTE-University Institute of Lisbon, Av. Forças Armadas, 1649-026 Lisboa, Portugal, and INESC-ID/IST, Technical University of Lisbon, Rua Alves Redol, 9, 1000-029 Lisboa, Portugal*

and

Marco A. Ribeiro^b

^b*Department of Information Science and Technology, ISCTE-University Institute of Lisbon, Av. Forças Armadas, 1649-026 Lisboa, Portugal, and Instituto de Telecomunicações, Av. Rovisco Pais 1, 1049-001, Lisboa, Portugal*

Abstract

A Monte Carlo algorithm is derived to solve the one-dimensional telegraph equations in a bounded domain subject to resistive and non-resistive boundary conditions. The proposed numerical scheme is more efficient than the classical Kac's theory because it does not require the discretization of time. The algorithm has been validated by comparing the results obtained with theory and the Finite-difference time domain (FDTD) method for a typical two-wire transmission line terminated at both ends with general boundary conditions. We have also tested transmission line heterogeneities to account for wave propagation in multiple media. The algorithm is inherently parallel, since it is based on Monte Carlo simulations, and does not suffer from the numerical dispersion and dissipation issues that arise in finite difference-based numerical schemes on a lossy medium. This allowed us to develop an efficient numerical method, capable of outperforming the classical FDTD method for large scale problems and high frequency signals.

Key words: Monte Carlo methods, telegrapher's equation, finite-difference time domain (FDTD).

PACS: 65C05, 65C30, 65M55, 65N55

Email addresses: juan.acebron@iscte.pt (Juan A. Acebrón), marco.ribeiro@iscte.pt (Marco A. Ribeiro).

1 Introduction

Probabilistic methods based on Monte Carlo simulations have been used already to solve problems in Science and Engineering modeled by partial differential equations. The most important difference compared with the classical methods used so far rests on the possibility of computing the solution at a single point, being therefore essentially a meshless-type method. From the computational point of view this feature can be exploited advantageously over the classical methods. In fact, since no computational mesh is required, the well known memory constraints for solving large scale and high dimensional problems are minimized. Moreover, since the solution is computed by taking the average of independent calculations, the underlying algorithms are well suited for parallel computing [1]. However, unless one is interested to compute the solution at single points, as happens in some specific applications on the analysis of systems and networks, they are typically not competitive enough compared with classical numerical algorithms, when used to compute the solution at every point inside a given computational domain. This is basically due to the well-known slow convergence rate of the Monte Carlo method. An alternative consists in combining the Monte Carlo method with other classical techniques, such as the domain decomposition method, computing merely the solution in a few points along some chosen interfaces inside the domain. This method is called probabilistic domain decomposition method (PDD), and was successfully used to solve a variety of problems modeled by elliptic, and linear and semilinear parabolic partial differential equations [2,3].

A key ingredient to implement a PDD method is to find a probabilistic representation of the solution. For linear elliptic equations, and parabolic equations, the probabilistic representation consists on the celebrated Feynman-Kac formula. In the specific field of electromagnetism, several accelerating techniques have been proposed in the literature to solve probabilistically electrostatic problems, such as the floating random walk [6], and walking on spheres [24,25], capable to speed up the computation of the solution by means of a variable time step size. For time-dependent electromagnetic problems, in particular for the time harmonic solution of the wave equation described by the scalar Helmholtz equation, a probabilistic representation was proposed in [5]. This representation is based on a suitable transformation, which allows to transform the original problem into two set of equations, one of them amenable to be solved using the Feynman-Kac formula.

In the general case of arbitrary hyperbolic partial differential equations, which model transport phenomena in Science and Engineering, a general probabilistic representation does not exist. However, there are two important exceptions. The first one is the Vlasov-Poisson system of equations, which appears to be of unquestionable interest in Plasma Physics. For such system of equations it was

proposed already in [22] a probabilistic representation in the Fourier space, and a corresponding numerical method [4]. The second exception consists on the one-dimensional telegraph equations. For these equations the pioneering work by Kac [16,17], and Golstein [12], showed that a probabilistic representation in terms of a Poisson random walk can be readily derived. Basically, this was due to the fact that the telegraph equations can be seen as a wave equation with dissipation, and hence amenable to be described as an expected value of a suitable functional of a given stochastic process. Originally, the method proposed by Kac was derived exclusively to deal with problems in unbounded domains, as well as, zero initial velocity. More recently, in [15], it has been conveniently generalized to tackle the problem of arbitrary initial velocity. By exploiting the link between the wave equation and the telegraph equations, through a suitable random time, some useful numerical schemes for computing multidimensional fields in dispersive media have been introduced as well.

It is missing however a general probabilistic method capable of dealing with bounded domains, and moreover, from a computational point of view, of being competitive to be used as an efficient alternative to the widespread FDTD numerical method for electromagnetics. The goal of this paper is to fill these two gaps by proposing a novel Monte Carlo method to solve the telegraph equations governing the evolution of voltage and current in a two-wire transmission line. We show in this article how the method can be extended with suitable boundary conditions, such as, feeding the line with a non-ideal voltage source with internal resistance at one end, and terminating it with a non-resistive load at other.

It is worth to remark that the telegraph equations are formally the same type of equations as the Maxwell equations in 1D, therefore the method proposed in this paper can be trivially extended to deal with such a system of equations.

Here it is the outline of the paper. The probabilistic algorithm for solving the telegraph equations is presented in Sec. 2 for the unbounded case. In particular, we discuss the computational advantages offered by such an algorithm compared with the classical one. In Sec. 3 the method is extended to cope with bounded domains subject to suitable boundary conditions. Sec. 4 is devoted to numerical examples, where the results obtained with our new algorithm are compared with the results obtained with the classical FDTD method and theoretical solutions based on Fourier analysis. Finally, to close the paper, in Sec.5 we summarize the more relevant findings and possible extensions of this work.

2 Probabilistic Formulation of Telegraph Equations

The system of first-order partial differential equations governing the evolution of the voltage $u(x, t)$ and current $i(x, t)$ in a general two-wire transmission line (also known as telegraphist's equations) are given by

$$\begin{aligned}\frac{\partial u}{\partial x} &= -L \frac{\partial i}{\partial t} - R i, \\ \frac{\partial i}{\partial x} &= -C \frac{\partial u}{\partial t} - G u,\end{aligned}\quad (1)$$

where R , L , G , and C , are the resistance, inductance, conductance, and capacitance per unit length of the line, respectively. For simplicity, let us assume that no boundary conditions are imposed and that we choose arbitrary initial conditions $u(x, 0) = u_0(x)$, and $i(x, 0) = i_0(x)$. Recall that the Maxwell equations for the electric and magnetic fields \mathbf{E} , and \mathbf{H} in a lossy media, assuming a x-directed z-polarized TEM wave ($H_x = E_x = 0$), with no variations in the y and z direction, are given by the one-dimensional equations

$$\begin{aligned}\frac{\partial E_y}{\partial x} &= -\mu \frac{\partial H_z}{\partial t}, \\ \frac{\partial H_z}{\partial x} &= -\varepsilon \frac{\partial E_y}{\partial t} - \sigma E_y,\end{aligned}\quad (2)$$

where ε , and μ are the electric permittivity and magnetic permeability, respectively, and σ the electric conductivity. Note that both equations, the telegraph equations and the Maxwell equations, are formally the same type of equations, obtaining one from the other by simply replacing the voltage u by the electric field E_y , the current i by the magnetic field H_z , and the parameters L by μ , C by ε , G by σ , and finally setting $R = 0$. Through the transformation $f(x, t) = (u + R_0 i)/2$, and $b(x, t) = (u - R_0 i)/2$, where $R_0 = \sqrt{L/C}$ is the constant characteristic resistance, Eqs. (1) can be rewritten as

$$\begin{aligned}\frac{\partial f}{\partial t} &= -c \frac{\partial f}{\partial x} + \lambda b - (\lambda + \mu) f, \\ \frac{\partial b}{\partial t} &= c \frac{\partial b}{\partial x} + \lambda f - (\lambda + \mu) b,\end{aligned}\quad (3)$$

where $c = 1/\sqrt{LC}$ is the constant phase velocity, $\mu = G/C$, and $\lambda = (R/L - G/C)/2$ is the distortion parameter, which reduces to zero in the special case of a distortion-less line.

Using the characteristic curves [19], the system equations (1) can be transformed into the following coupled integral equations:

$$\begin{aligned}
f(x, t) &= f_0(x - ct) e^{-(\lambda+\mu)t} + \lambda \int_0^t ds e^{-(\lambda+\mu)s} b(x - cs, t - s), \\
b(x, t) &= b_0(x + ct) e^{-(\lambda+\mu)t} + \lambda \int_0^t ds e^{-(\lambda+\mu)s} f(x + cs, t - s),
\end{aligned} \tag{4}$$

where $f_0(x) = f(x, 0)$, and $b_0(x) = b(x, 0)$.

Let us assume S to be an absolutely continuous random variable defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with $S : \Omega \rightarrow \mathbb{R}_0^+$, and governed by the exponential density function $p(s) = \frac{d}{ds} P[s < S] = (\lambda + \mu)e^{-(\lambda+\mu)s}$. Thus, the following probabilistic representation of Eqs. (3) are obtained

$$\begin{aligned}
f(x, t) &= f_0(x - ct)P[S > t] + \rho \mathbb{E} \left[b(x - cS, t - S) \mathbb{1}_{[S \leq t]} \right], \\
b(x, t) &= b_0(x + ct)P[S > t] + \rho \mathbb{E} \left[f(x + cS, t - S) \mathbb{1}_{[S \leq t]} \right],
\end{aligned} \tag{5}$$

where $\rho = \lambda/(\lambda + \mu)$, and \mathbb{E} denotes the expected value. Here $\mathbb{1}_A$ denotes the indicator function, being 1 or 0 depending on whether the event A occurs. Note that this system of equations is an implicit system in which to evaluate f is required to evaluate b and vice-versa. This can be readily solved resorting to Picard iteration, which has been already used successfully in a more general setting with nonlinear equations, see [23] e.g.. There it was shown that the coefficients of the so obtained Picard series can be probabilistically sampled since they satisfy a normalization condition. Here a similar procedure can be applied straightforwardly, and to implement it in practice we propose the following recursive algorithm: Generate a first random time S_0 obeying the exponential density function. Then, depending on whether $S_0 < t$ or not, two different alternatives are taken. If $S_0 > t$, the algorithm is stopped, and the contribution to the solution $f(x, t)$ and $b(x, t)$ of this particular realization of the random variable is obtained by evaluating the initial condition f_0 and b_0 at the points $x - ct$, and $x + ct$, respectively. If, on the contrary, $S_0 < t$, then generate a second random number S_1 distributed according to the same density function. Once again two different alternatives follow depending on whether $S_1 > t - S_0$ or not. If $S_1 > t - S_0$, the algorithm is stopped, and the contribution to the solution $f(x, t)$ and $b(x, t)$ is obtained by multiplying ρ by the initial conditions b_0 and f_0 at the points $x - cS_0 + c(t - S_0)$, and $x + cS_0 - c(t - S_0)$, respectively. If $S_1 < (t - S_0)$ the algorithm proceeds repeating the same elementary rules. In Fig. 1 we present a flux diagram of a practical implementation of the algorithm.

The overall solution is constructed summing up the different partial contributions obtained from each trajectory. In fact, Eqs. (5) can be solved recursively resorting to Picard iteration, as was mentioned above, by replacing the right-hand side, which depends on the functions $f(x, t)$, and $b(x, t)$, with the solutions itself. This in practice gives rise to the following expansion in terms of random exponential times,

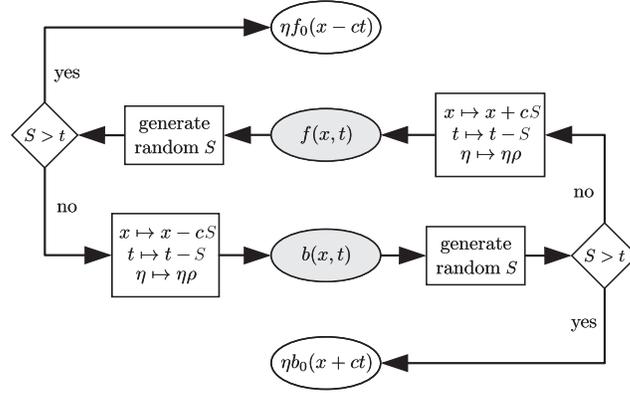


Fig. 1. Recursive algorithm for solving the unbounded problem. Initially, η is set to one, and the initial states are $f(x, t)$, or $b(x, t)$, depending on whether one is interested to compute the solution f or b , respectively.

$$\begin{aligned}
 f(x, t) &= f_0(x - ct) P[S_0 > t] \\
 &+ \rho \mathbb{E} \left[b_0(x - cS_0 + c(t - S_0)) \mathbf{1}_{[S_0 \leq t]} \mathbf{1}_{[S_1 > t]} \right] \\
 &+ \rho^2 \mathbb{E} \left[f_0(x - cS_0 + cS_1 - c(t - S_0 - S_1)) \mathbf{1}_{[S_0 \leq t]} \mathbf{1}_{[S_1 \leq t]} \mathbf{1}_{[S_2 > t]} \right] + \dots, \\
 b(x, t) &= b_0(x + ct) P[S_0 > t] \\
 &+ \rho \mathbb{E} \left[f_0(x + cS_0 + c(t - S_0)) \mathbf{1}_{[S_0 \leq t]} \mathbf{1}_{[S_1 > t]} \right] \\
 &+ \rho^2 \mathbb{E} \left[b_0(x + cS_0 - cS_1 + c(t - S_0 - S_1)) \mathbf{1}_{[S_0 \leq t]} \mathbf{1}_{[S_1 \leq t]} \mathbf{1}_{[S_2 > t]} \right] + \dots \quad (6)
 \end{aligned}$$

Note that for a given realization ω of the random variables S_i , the corresponding arguments used so far to evaluate the initial condition f_0 and b_0 are given in general as $x - cS_0 + cS_1 - \dots - cS_N + c(t - S_0 - S_1 - \dots - S_N)$, and $x + cS_0 - cS_1 + \dots + cS_N - c(t - S_0 - S_1 - \dots - S_N)$. Those starting with $x - cS_0 + \dots$ correspond to arguments associated with the function $f(x, t)$, while those starting with $x + cS_0 + \dots$ are associated with the function $b(x, t)$. Regarding the arguments of the initial conditions f_0 and b_0 , the following general rule holds: the number of random times s_i appearing as the argument of b_0 is odd when used for evaluating the function $f(x, t)$ and even when used for evaluating the function $b(x, t)$. For the initial condition f_0 the rule is exactly the opposite, being an odd number for evaluating the function $b(x, t)$, while an even number for evaluating the function $f(x, t)$. Therefore, using such sequences of exponential random times S_i , the system equations (5) can be reformulated in the following compact form,

$$\begin{aligned}
 f(x, t) &= \mathbb{E} \left[\rho^{n(\omega)} f_0(x - \beta_\omega(t)) \mathbf{1}_{[n(\omega) \equiv 0 \pmod{2}]} \right] \\
 &+ \mathbb{E} \left[\rho^{n(\omega)} b_0(x - \beta_\omega(t)) \mathbf{1}_{[n(\omega) \equiv 1 \pmod{2}]} \right] \\
 b(x, t) &= \mathbb{E} \left[\rho^{n(\omega)} f_0(x + \beta_\omega(t)) \mathbf{1}_{[n(\omega) \equiv 1 \pmod{2}]} \right] \\
 &+ \mathbb{E} \left[\rho^{n(\omega)} b_0(x + \beta_\omega(t)) \mathbf{1}_{[n(\omega) \equiv 0 \pmod{2}]} \right] \quad (7)
 \end{aligned}$$

where

$$\beta_\omega(t) = c \left(\sum_{i=0}^{n(\omega)-1} (-1)^i S_i + \left(t - \sum_{i=0}^{n(\omega)-1} S_i \right) \right), \quad (8)$$

$n(\omega)$ is the random length of the sequence. Note that in practice $x \pm \beta_\omega(t)$ can be seen as a random trajectory in time, playing a similar role that the characteristic curves play for the first-order wave equation. In fact, taking λ and μ equal to zero, the classical solution for the wave equation is recovered, since the probability of $S > t$ goes to one, and then $n(\omega) = 0$ with probability one.

A numerical method can be readily proposed, since the solution $f(x, t)$, and $b(x, t)$ can be approximated numerically replacing the expected value by the corresponding estimator, the arithmetic mean, thus

$$f(x, t) = \frac{1}{N} \sum_{j=1}^N f_{\omega_j}, \quad b(x, t) = \frac{1}{N} \sum_{j=1}^N b_{\omega_j}, \quad (9)$$

where N is the sample size, and f_{ω_j} , and b_{ω_j} corresponds to the partial contribution to the solution $f(x, t)$, and $b(x, t)$, respectively, of the random path $\beta_{\omega_j}(t)$. Such a contribution can be computed as follows,

$$\begin{aligned} f_{\omega_j} &= \rho^{n(\omega_j)} \{ f_0(x - \beta_{\omega_j}(t)) \mathbb{1}_{[n(\omega_j) \equiv 0 \pmod{2}]} \\ &\quad + b_0(x - \beta_{\omega_j}(t)) \mathbb{1}_{[n(\omega_j) \equiv 1 \pmod{2}]} \}, \\ b_{\omega_j} &= \rho^{n(\omega_j)} \{ f_0(x + \beta_{\omega_j}(t)) \mathbb{1}_{[n(\omega_j) \equiv 1 \pmod{2}]} \\ &\quad + b_0(x + \beta_{\omega_j}(t)) \mathbb{1}_{[n(\omega_j) \equiv 0 \pmod{2}]} \}, \end{aligned} \quad (10)$$

Note that such a finite sample size unavoidably introduces a numerical error, which is statistical in nature, and known to be of order of $O(1/\sqrt{N})$ when N goes to infinity [9], provided the variance of the random process is finite. Although a general proof for any arbitrary initial data may not exist, however for all the examples studied so far, numerical results have confirmed that the variance remains finite in all cases. Intuitively, this can be explained due to the fact that physically meaningful chosen initial and boundary data decays exponentially fast, assuring in such a way the convergence of the numerical method. Therefore, the accuracy of the computed numerical solution will be determined exclusively by the sample size N . Furthermore, it is important to remark here that unlike classical methods such a probabilistic numerical method does not suffer of any dispersion and dissipation issues, since no approximation schemes of the spatial and time derivatives of the solution are made.

From the computational point of view, our numerical method is more efficient

than any classical Monte Carlo method based on the theory of Kac regarding the second order telegraph equation, as in [15]. This is so because our numerical scheme does not require any discretization to obtain the random time. On the other hand, when applying the classical theory of Kac for numerical purposes, the definite time integral given by

$$\int_0^t (-1)^{N(s)} ds, \quad (11)$$

where $N(t)$ is a Poisson random process, has to be numerically approximated by discretizing conveniently the time variable s . This introduces a new source of error and the computational time spent to compute the solution will be higher when compared with our algorithm, especially because in the method of Kac an accurate solution can only be found if the time step is kept reasonably small. The distinguishing feature of our new method is that, contrarily to the theory of Kac, it generates the corresponding random paths based only on those random times for which there is a change of path direction. A graphical illustration of the operation of the probabilistic algorithm is given in Fig. 2 which shows three possible random paths – identified as (1), (2) and (3) – corresponding to different realizations of $x + \beta(\omega)$, used to evaluate the solution at the point $x = 0$ and time $t = 20$ ns. The parameters considered in the simulation were $c = 2 \times 10^8$ m/s, $\mu = 5.0 \times 10^6$ s⁻¹ and $\lambda = 22.5 \times 10^6$ s⁻¹. The partial contributions to the solution are $\rho^4 f_0(x - c s_1 + c s_2 - c s_3 + c s_4 + c(t - s_1 - s_2 - s_3 - s_4))$ for the random path marked as (1) $\rho^3 f_0(x + c s'_1 - c s'_2 + c s'_3 - c(t - s'_1 - s'_2 - s'_3))$ for that marked as (2) and $\rho b_0(x - c s''_1 + c(t - s''_1))$ for path (3). Note that the solid lines (1) and (3) correspond to trajectories that contribute to the solution of f , whereas the dotted line (2) contributes to the solution of b .

The computational cost of both algorithms can be readily derived, and is given by

$$C_{\text{Kac}} \approx N \frac{t}{\Delta t}, \quad C_{\text{New algorithm}} \approx N \frac{t}{\langle s \rangle}, \quad (12)$$

where t is the final time, N the number of trajectories, and $\langle s \rangle$ denotes the time average related with the random time when trajectories change their direction of movement. This time average is equal to the inverse of the mean of the exponential probability density $p(s)$, which is given by $1/(\lambda + \mu)$. The computational cost of both algorithms grows linearly on the final time t . In the classical algorithm of Kac the computational cost is also inversely proportional to the discretized time step Δt , being greater when Δt gets smaller. The computational cost of our new probabilistic method depends exclusively on the parameters λ and μ . In Table 1 we present the computational time and the numerical error of our method as a function of the final time t for the

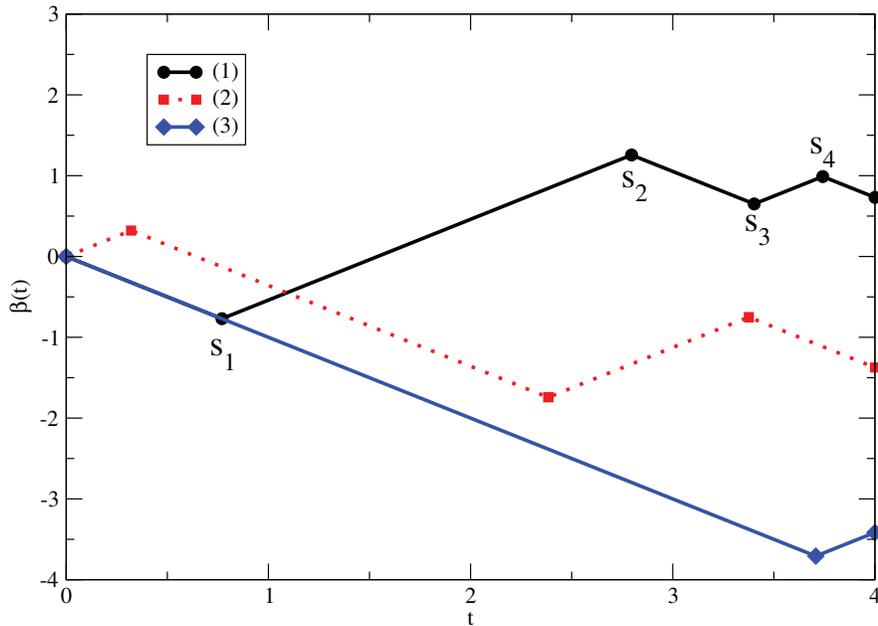


Fig. 2. Random paths for three different realizations of $x + \beta_\omega(t)$. The trajectories start at position $x = 0$ and run until $t = 20$ ns.

case of a transmission line fed by a wave packet signal and terminated by a resistive load. The specific values of the parameters used for this simulation are found in Fig. (5). The number of trajectories was kept fixed to $N = 10^5$ and, as can be seen, the computational cost grows linearly with t , while the numerical error remains almost constant. The discrete L^∞ -error norm has been obtained, as usual, as $\max(|u_{MC}(x_i, t) - u(x_i, t)| : 1 \leq i \leq N)$ (see [14], e.g.), where u_{MC} corresponds to the numerical solution computed with our Monte Carlo method for single spatial points x_i equally spaced between 0 and 2, and u is the theoretical solution in Eq. (21) evaluated at such points.

In the simple case of initial conditions given by $f_0(x) = b_0(x) = \delta(x)$, an analytical solution was found in [21], and it was shown in Fig. 3 to validate the proposed algorithm. Note that the agreement is perfect for both solutions, $f(x, t)$, and $b(x, t)$.

Table 1

CPU time and numerical error in the norm L^∞ obtained when the voltage source signal is a pulse for different values of the final time t . The parameters are as in Fig. 5.

Final Time (ns)	CPU Time (s)	Error L^∞
15	13.003	0.00017015
20	16.944	0.00010262
25	20.304	0.00015178
30	25.801	0.00018977

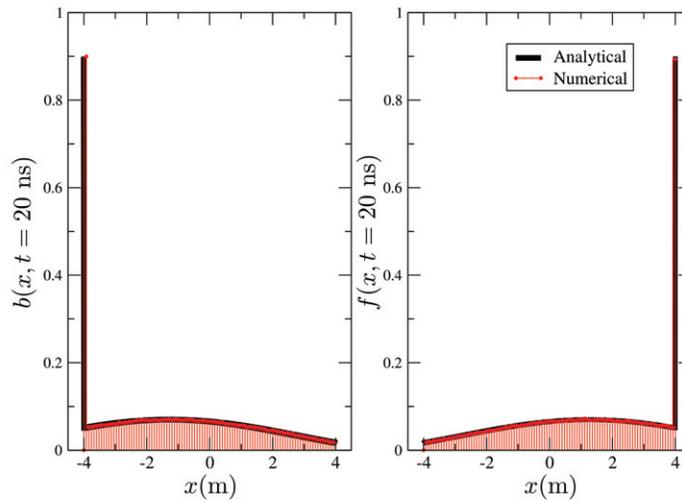


Fig. 3. Comparison between the analytical solution and the numerical solution obtained by the algorithm in Eq. (5) at time $t = 20$ ns. The number of trajectories is kept fixed to $N = 10^5$.

3 Probabilistic Formulation of Telegraph Equations with boundary conditions

One of the interesting features of the algorithm is how it can be easily generalized to deal with boundary conditions. To illustrate how this is done, let consider a non-ideal voltage source $v_g(t)$ with series internal resistance R_g connected to the input terminals of a general two-wire transmission line terminated with a linear time invariant network of lumped elements. This network is generally described by a linear differential equation for the voltage, current and their time derivatives (to several orders) at the end of the line. The number of independent energy-storage elements of the network determine the

order of this equation. As it is well known, the time domain relation between the voltage and current in such a system can be characterized by a convolution operation. Thus, assuming that the line has a length l , the voltage $u(x, t)$ and current $i(x, t)$ at the boundaries $x = 0$, and $x = l$, are given by

$$u(0, t) = v_g(t) - i(0, t) R_g, \quad i(l, t) = \int_{-\infty}^{\infty} u(l, \lambda) h(t - \lambda) d\lambda, \quad (13)$$

where $h(t)$ is the impulse response of the terminal network connected to the transmission line at $x = l$. In terms of $f(x, t)$, and $b(x, t)$, the corresponding boundary conditions are

$$\begin{aligned} f(0, t) &= \zeta v_g(t) + \Gamma_g b(0, t), \\ b(l, t) &= f(l, t) - R_0 \int_{-\infty}^{\infty} [f(l, \lambda) + b(l, \lambda)] h(t - \lambda) d\lambda, \end{aligned} \quad (14)$$

where

$$\zeta = \frac{R_0}{R_0 + R_g}, \quad \Gamma_g = \frac{R_g - R_0}{R_g + R_0}. \quad (15)$$

Consider for example that an inductive circuit with resistance R_L and a inductance L_L (series circuit) is connected to the end of the line at $x = l$. The impulse response of the network is then given by

$$h(t) = \frac{1}{L_L} e^{-\frac{R_L}{L_L} t} H(t), \quad (16)$$

Thus, according to Eq. (14), the value of b at the boundary $x = l$ for a given time t depends on the value of f for that time but also on the values of f and b for previous instants of time between 0 and t . This is consistent with the fact that inductive elements store energy and therefore have memory. Furthermore, according to the exponential decay characterized by the time constant $\tau_L = L_L/R_L$ in Eq. (16), the values of f and b at the boundary in earlier times contribute less for the values of b at time t , as theoretically expected. For a purely resistive circuit ($L_L = 0$) the impulse response is the following:

$$h(t) = \frac{1}{R_L} \delta(t), \quad (17)$$

and from Eq. (14) it yields

$$b(l, t) = \Gamma_L f(l, t), \quad \Gamma_L = \frac{R_L - R_0}{R_L + R_0}. \quad (18)$$

Now, at the load terminal, the values taken by f and b at previous times have no influence on the values of b at the current time t , a clear manifestation of the fact that the circuit has no memory. The solution of the transmission line equations with these general boundary conditions is given by the following system of integral equations,

$$\begin{aligned}
f(x, t) &= f_0(x - ct) e^{-(\lambda+\mu)t} \mathbb{1}_{[x>ct]} \\
&+ \lambda \int_0^t ds e^{-(\lambda+\mu)s} b(x - cs, t - s) \mathbb{1}_{[x \geq cs]} \\
&+ \int_0^t ds e^{-(\lambda+\mu)s} \delta(cs - x) f(0, t - s) \mathbb{1}_{[x \leq ct]}, \\
b(x, t) &= b_0(x + ct) e^{-(\lambda+\mu)t} \mathbb{1}_{[l-x>ct]} \\
&+ \lambda \int_0^t ds e^{-(\lambda+\mu)s} f(x + cs, t - s) \mathbb{1}_{[l-x \geq cs]} \\
&+ \int_0^t ds e^{-(\lambda+\mu)s} \delta(cs + x - l) b(l, t - s) \mathbb{1}_{[l-x \leq ct]}, \tag{19}
\end{aligned}$$

which is a generalization of Eqs. (4) to deal now with load and generator constraints.

A probabilistic method can be derived as in the unbounded case, defining a random time S exponentially distributed according to the same probability density $p(s)$, and yields,

$$\begin{aligned}
f(x, t) &= f_0(x - ct) P[S > t] \mathbb{1}_{[x>ct]} + \rho \mathbb{E} \left[b(x - cS, t - S) \mathbb{1}_{[S \leq t]} \mathbb{1}_{[x \geq cS]} \right] \\
&+ \rho \lambda^{-1} \mathbb{E} [f(0, t - S) \delta(cS - x)] \mathbb{1}_{[x \leq ct]} \\
b(x, t) &= b_0(x + ct) P[S > t] \mathbb{1}_{[l-x>ct]} \\
&+ \rho \mathbb{E} \left[f(x + cS, t - S) \mathbb{1}_{[S \leq t]} \mathbb{1}_{[l-x \geq cS]} \right] \\
&+ \rho \lambda^{-1} \mathbb{E} \left[b(l, t - S) \delta(cS + x - l) \mathbb{1}_{[l-x \leq ct]} \right] \tag{20}
\end{aligned}$$

Note that contrary to what happens in the unbounded case, when the generated random time S is greater than the time t , two different actions takes place. In fact, to compute the solution $f(x, t)$, the function b has to be evaluated at the points $x - cS$, and the function f at the point $x = 0$, provided that the random time S coincides with x/c . This should be done, as it was for the unbounded case, recursively. However, for the bounded case independently of the generated random time s being or not larger than t , the algorithm should check whether or not the random path crosses the boundaries at 0 or l . In case of crossing for any of both boundaries, the random path is reflected back into the domain. Further, for the boundary at $x = 0$, and according to the boundary conditions defined in (14), the voltage source $v_g(t)$ has to be evaluated at the time the random path crossed such a boundary. To illustrate how the algorithm can be implemented in practice, a pseudocode is given in Algorithm 1.

When the general boundary conditions in Eq. (14) are considered, the procedure is the following for the boundary at l :

- (1) Generate a two-point discrete random variable ξ_1 taking the values 0, and

Algorithm 1 Telegraph equations with boundary conditions

Require: $x, t, N, l, \Gamma_L, \Gamma_g, \rho, v_g, \zeta$

```

for  $i = 1, 2$  do
   $sign = (-1)^i$ 
  for  $n = 1, \dots, N$  do
     $\tau = 0, \eta = 1, X = x$ 
    while  $\tau < T$  do
      repeat: generate(S)
       $\tau = \tau + S, X' = X + sign\ c\ S$ 
      if  $X' < 0$  then
        call UPDATE LEFT
        goto repeat
      else if  $X' > l$  then
        call UPDATE RIGHT
        goto repeat
      end if
       $sign = -sign, \eta = \eta\rho, X = X'$ 
    end while
     $X' = X + sign\ c[T - (\tau - S)]$ 
    if  $X' < 0$  then
      call UPDATE LEFT
      goto repeat
    else if  $X' > l$  then
      call UPDATE RIGHT
      goto repeat
    end if
     $X = X'$ 
  end for
   $solution[i] = solution[i]/N$ 
end for
 $f(x, t) = solution[1], b(x, t) = solution[2]$ 

procedure UPDATE LEFT( $X, sign, \tau, solution$ )
   $\tau = \tau - S - sign\ x/c, sign = -sign, X = 0$ 
   $solution[i] = solution[i] + \eta\ \zeta\ v_g(t - \tau), \eta = \eta\ \Gamma_g$ 
end procedure

procedure UPDATE RIGHT( $X, sign, \tau$ )
   $\tau = \tau - S + sign\ (l - x)/c, sign = -sign, X = l$ 
   $\eta = \eta\ \Gamma_L$ 
end procedure

```

- 1, with probability 1/2.
- (2) If $\xi_1 = 0$, then the first term in the right-hand-side of Eq. (14) is chosen, which corresponds to evaluate the function f at $x = l$, and time t , otherwise if $\xi_1 = 1$ the second term should be chosen.
 - (3) When $\xi_1 = 1$, a new two-point discrete random variable ξ_2 with the same properties as ξ_1 is generated. Such a new random variable is used to choose between evaluating the function f or b according whether ξ_2 is 1, or 0 respectively. Both functions are evaluated at the time $t - S$, where S is obtained randomly from a exponential probability density given by $p(s) = (R_L/L_L)e^{-(R_L/L_L)s} \mathbb{1}_{[s \geq 0]}$. Note that in case of a purely resistive load S is 0 with probability one, and therefore the functions f and b are evaluated always at the time t .
 - (4) Repeat recursively the previous steps replacing conveniently the arguments of the functions f and b .

In practice, the bounded case entails different scenarios for the random paths, and in Fig. 4 some simulated random paths are depicted for the purpose of illustration for the more simple purely resistive circuit. Let suppose for simplicity that $f_0(x) = b_0(x) = 0$, then the partial contributions to the solution $f(x, t)$, and $b(x, t)$ of the random paths are: (1) $\zeta v_g(t - S_1) + \Gamma_g \rho \zeta v_g(t - S_1 - S_2 - S_3)$; (2) 0; and (3) $\zeta v_g(t - S'_1) + \Gamma_g \rho \zeta v_g(t - S'_1 - S'_2 - S'_3)$.

Concerning the numerical method for bounded problems, it is worth to remark that the probabilistic methods proposed in literature for solving boundary value problems for elliptic or parabolic partial differential equations are based on the well-known Feynman-Kac formula [18]. When this formula is used for numerical purpose, to obtain accurate solutions requires to compute precisely the first exit time of a diffusion process (or a Brownian motion in particular for a Laplace operator). Numerical experiments show that the error made estimating the first exit time may be dominant over the other sources of numerical errors and is therefore of paramount importance in providing an accurate value of such a quantity [2]. Rather, the corresponding probabilistic method for the telegraph equations does not exhibit this issue, mainly because the nature of the generated random process is completely different. Indeed the underlying random trajectories can be seen as piecewise-deterministic, being characterized by a switch of deterministic straight lines of slope c or $-c$ at random times, as shown in Fig. 4. Moreover, the density function of the first exit time is a delta function, and thus no error is done to evaluate it numerically.

Summarizing, the statistical error is the unique source of errors of the proposed numerical method for solving unbounded and bounded problems. The method does not require any computational mesh, being essentially meshless, and it allows to compute the solution at arbitrary point in space and time through a recursive procedure which is probabilistic in nature. Moreover, computing the solution at any given time does not require to compute and keep the solution

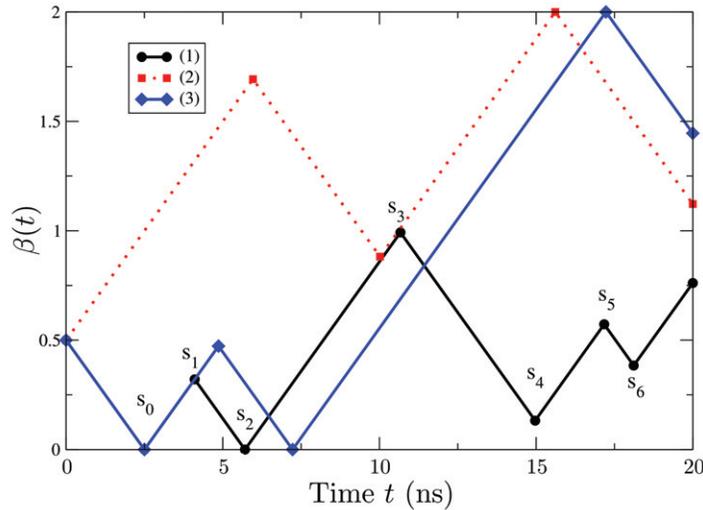


Fig. 4. Three random trajectories for three different realizations of $x + \beta_\omega(t)$ in a bounded domain $[0, 2]$ m. The paths start at $x = 0.5$, and are stopped at time $t = 20$ ns. The solid curves (1) and (3) correspond to paths belonging to the solution f , while the dotted line (2) to paths for the solution of b .

at intermediate times. Indeed, the probabilistic method generates the whole history of the underlying random process, and the solution is computed using exclusively the initial and boundary data.

4 Numerical Results

In this section, we present numerical examples to illustrate the probabilistic method introduced in the previous section. In order to assess the validity of our method, a comparison was made solving the problems by the classical FDTD method. In the following, by theoretical solution it is meant the solution obtained solving analytically the telegraph equations in Fourier space. The voltage $u(x, t)$ can be obtained from its counterpart $u(x, \omega)$ through the Fourier transform,

$$u(x, t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} u(x, \omega) e^{j\omega t} d\omega, \quad (21)$$

where $u(x, \omega)$ is given by

$$u(x, \omega) = \sum_{i=0}^{\infty} u_i(x, \omega). \quad (22)$$

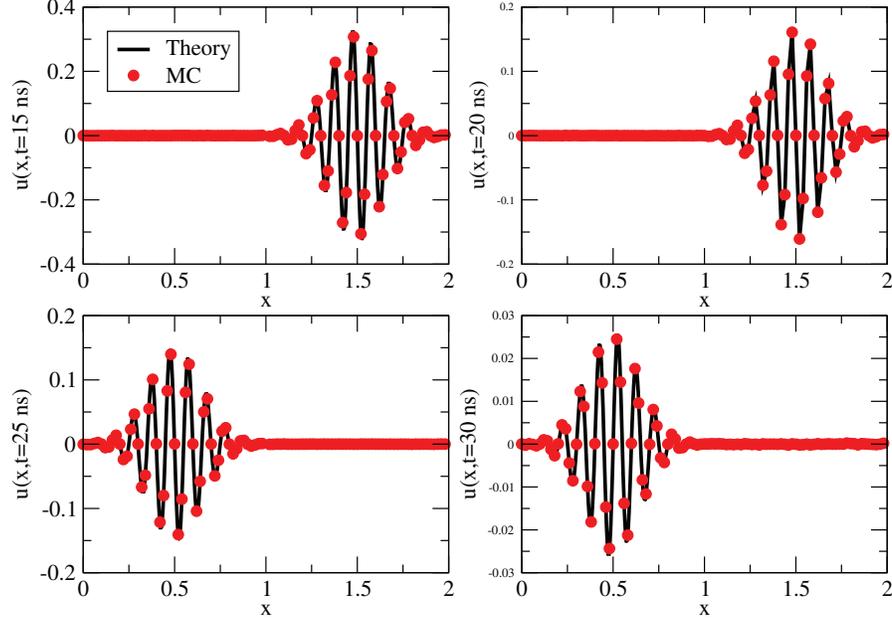


Fig. 5. Comparison between the numerical results obtained with the Monte Carlo method (MC), and the theoretical solution, for four different time values.

The index i identifies a pair of forward and backward waves. For i equal to zero, $u_0(x, \omega)$ is composed by the wave coming from the generator, which travels forward from $x = 0$ to $x = l$, and by the backward wave from $x = l$ to $x = 0$. The backward wave is reflected on the generator, giving birth to a new forward wave which will later reflect on the load, thus originating a backward wave. These two waves represent $u_1(x, \omega)$. In general $u_i(x, \omega)$ is obtained following the same constructive procedure, and yields

$$u_i(x, \omega) = u_0(x, \omega) \Gamma_g^i(\omega) \Gamma_L^i(\omega) e^{-2j k(\omega) l i}, \quad (23)$$

where $k(\omega)$ is the complex wave number given by

$$k(\omega) = -\frac{j}{c} \sqrt{(j\omega + R/L)(j\omega + G/C)}, \quad (24)$$

and $c = 1/\sqrt{LC}$. Note that for the lossy case k is a complex number, and its dependence with ω is not linear, and therefore the medium is lossy and dispersive. In Eq. (23), u_0 , is given by

$$u_0(x, \omega) = \zeta(\omega) V_g(\omega) \left[e^{-j k(\omega) x} + \Gamma_L(\omega) e^{-j k(\omega) (2l-x)} \right], \quad (25)$$

where

$$\begin{aligned}\Gamma_L(\omega) &= \frac{Z_L(\omega) - Z_0(\omega)}{Z_L(\omega) + Z_0(\omega)}, & \Gamma_g(\omega) &= \frac{R_g - Z_0(\omega)}{R_g + Z_0(\omega)}, \\ \zeta(\omega) &= \frac{Z_0(\omega)}{Z_0(\omega) + R_g}, & Z_0(\omega) &= \sqrt{\frac{R + j\omega L}{G + j\omega C}}.\end{aligned}\quad (26)$$

It is worth to mention that the complex coefficients above are not in general those given in Eq. (15). In fact, they coincide only for the case of a lossless line. Remarkably, the Monte Carlo method is capable of solving the telegraph equations encoding all the information regarding losses and dispersion in the system of equations (20), rather than in the coefficients characterizing the line, as it happens in the frequency domain theory described above.

In Fig. 5 it is depicted the numerical results obtained when solving a transmission line with a purely resistive load subject to a voltage source given by a wavepacket, $v_g(t) = \sin(2\pi ft) \exp[-(t - t_0)^2/(2s^2)]$, where $t_0 = 7.5$ ns, $s_0 = 0.75$ ns and $f = 1.0$ GHz. The parameters of the line are $l = 2$ m, $L = 0.250$ $\mu\text{H}/\text{m}$, $C = 100.0$ pF/m, $R = 12.5$ Ω/m , $G = 0.5$ mS/m, $R_L = 12.5$ Ω , $R_g = 75$ Ω . For the Monte Carlo method N was 10^5 . The theoretical solution in the figure was obtained from Eqs. (21)-(26) introducing the Fourier transform of the wavepacket in Eq. (25). The four figures correspond to instants of time (a) before reaching any boundary; (b) immediately after reaching the right boundary (load) located at $x = 2$; (c) exactly before reaching the left boundary (source) located at $x = 0$, and (d) immediately after reaching the left boundary. Note the very good agreement between the Monte Carlo method and the theoretical solution for every case, which validates the Monte Carlo method with boundary conditions on the load and source.

It is worth to remark that the numerical solution computed with the Monte Carlo method has been obtained pointwise, since the numerical method allows to compute the solution at a single point without the need of any computational mesh, as rather it happens for the FDTD method. Specifically for this example, since the frequency f is a free parameter, it can be used to test the accuracy of the method for arbitrarily large frequency values.

Indeed this has been done, and the numerical error in the L^∞ norm summarized in Table 2 for different number of spatial and temporal cells. Here the numerical experiments were done to mimic experimental setups built to monitoring in time the voltage and current of the transmission line at a given point inside the line. Thus, the final time were kept fixed to 2 in all simulations, and the solution computed at temporal points equally distributed between 0 and 2. Concerning the spatial coordinate, for the Monte Carlo method the simulations were run at the point x equal to 0.5. Recall that for the Monte Carlo method, a computational mesh is not needed, and therefore the solution

Table 2

Numerical error in the norm L^∞ obtained with the wavepacket source as the carrier frequency increases. All simulations are run until $t = 10$ ns.

Frequency [GHz]	FDTD					MC
	16384	32768	65536	131072	16384($c\Delta t = \Delta x/2$)	$N = 1000$
3	0.002088	0.001016	0.000471	0.000202	0.000932	0.014497
19	0.013234	0.006401	0.002988	0.001285	0.012136	0.014648
40	0.027896	0.013449	0.006291	0.002701	0.312132	0.013457
90	0.063151	0.030387	0.013171	0.006077	0.370000	0.015282
200	0.146617	0.068182	0.031572	0.013520	0.732208	0.014351

can be obtained at a single point. On the contrary for the FDTD method, the solution has to be computed in any point of the computational domain, and this regardless of only needing the solution in time for a single spatial point. Column 2, 3, 4, and 5 corresponds to the numerical error done when computed numerically the solution by the FDTD method with 16,384, 32,768, 65,536, and 131,072 spatial and temporal discretization cells, respectively. Note that in terms of the discretization parameters, Δt , and Δx , this corresponds to choose both to be equal. Rather in column 6, the number of temporal cells were chosen to be the double of the spatial cells, or equivalently $c\Delta t = \Delta x/2$. This has been done to show that the minimum error can be obtained, when the time step is chosen to be the well known "magic" time step. In fact, this was reported already in literature, see [13] e.g. Concerning the last column of the table, this corresponds to the numerical error obtained by the Monte Carlo method, being the number of paths chosen to be $N = 1000$, the solution computed at a single spatial point, and 16,384 points equally distributed in time between 0, and 2.

It is worth to remark that contrarily to the FDTD method, where the numerical error seems to grow linearly with the value of the frequency, the numerical error done with the Monte Carlo method turns out to be independent of it. Therefore, to keep constant the numerical error for increasingly large values of the frequency, this requires for the FDTD method increasing proportionally the number of cells. This has been explicitly pointed out in Table 2 marking in boldface the numerical error of order of 0.013. Moreover, since the computational time depends linearly on the number of cells, this means that such a time should increase accordingly whenever the value of the frequency grows. On the contrary, the computational time spent by the Monte Carlo method will remain constant independently of the value of the frequency. Such a behavior is indeed observed, being the results depicted in Fig. 6. To measure properly the CPU time, all simulations were carried out in sequential mode at an standard Linux Cluster equipped with Intel Xeon processors running at 2.27 GHz under the same computational conditions. Note that for sufficiently large values of the frequency, the Monte Carlo method clearly outperforms the FDTD method, being the differences even more striking when the length of the line increases. This is because the solution obtained by the Monte Carlo

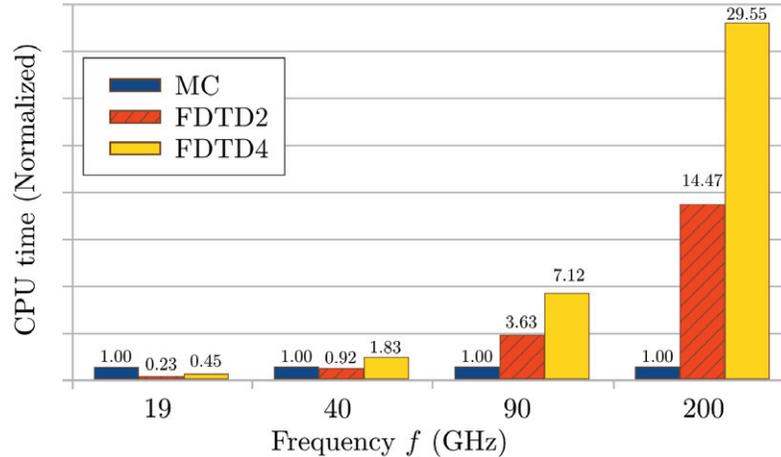


Fig. 6. Comparison between the CPU simulation times of the MC and FDTD methods, when solving the telegraph equations for several frequency values of the wavepacket pulse. Values are normalized to the MC CPU time. Here two different values of the length of the line, $l = 2$ m (denoted as FDTD2), and $l = 4$ m (FDTD4) are shown, coloured as orange and yellow, respectively, for the FDTD method. For both numerical methods the numerical error has been kept fixed to 10^{-2} .

method is computed at a single spatial point, and as it was explained above, the only source of error is statistical in nature. Therefore, the numerical error is completely independent of the length of the line, and consequently no more discretization cells are needed to keep constant the numerical error, as it happens on the contrary for the the FDTD method.

To illustrate the potential of the algorithm for solving non-resistive transmission line, the same type of simulations were done now for the case of a gaussian pulse, $v_g(t) = \exp[-(t-t_0)^2/(2s^2)]$. The transmission line is terminated with a resistance R_L , and an inductance L_L . The results are plotted in Fig. 7. Again the agreement between the results obtained with the Monte Carlo method and the theoretical solution is perfect.

As an example for multiple propagation mediums, in Fig. 8 it is plotted the results obtained when solving a transmission line composed of three segments connected in series with two different medium properties. The middle segment has an inductance nine times larger than the other two segments. To solve numerically this problem it is required to impose continuity for both, the voltage $u(x, t)$, and the current $i(x, t)$, at the interfaces separating the different segments. In terms of the functions $f_j(x, t)$ and $b_j(x, t)$ for a general multiple mediums transmission line composed by n segments, where $j = 1, \dots, n$ denotes the j th segment, is given by

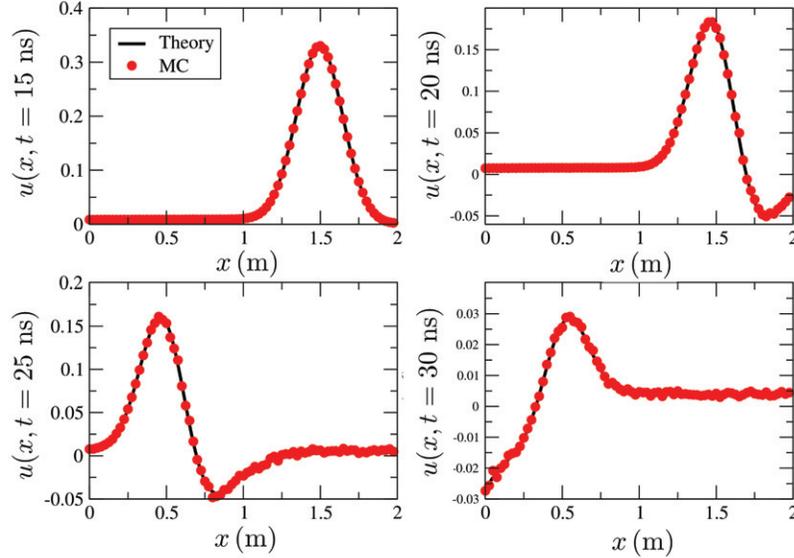


Fig. 7. Comparison between the numerical results obtained with the Monte Carlo method (MC), and the theoretical solution, for four different time values. The parameters at the load are $R_L = 100 \Omega$, $L_L = 0.0625 \mu\text{H/m}$, and the other parameters as in previous figures. The voltage source is a gaussian pulse, with $t_0 = 7.5 \text{ ns}$, $s_0 = 0.75 \text{ ns}$.

$$b_j(x_j, t) = \frac{\frac{2}{R_0^{j+1}} b_{j+1}(x_j, t) + f_j(x_j, t) \left(\frac{1}{R_0^j} - \frac{1}{R_0^{j+1}} \right)}{\left(\frac{1}{R_0^j} + \frac{1}{R_0^{j+1}} \right)},$$

$$f_{j+1}(x_j, t) = \frac{\frac{2}{R_0^j} f_j(x_j, t) - b_{j+1}(x_j, t) \left(\frac{1}{R_0^j} - \frac{1}{R_0^{j+1}} \right)}{\left(\frac{1}{R_0^j} + \frac{1}{R_0^{j+1}} \right)}, \quad (27)$$

where x_j corresponds to the spatial position of the j th interface, and R_0^j is the constant characteristic resistance of the segment j . The Monte Carlo method for solving this problem is implemented now as follows: Whenever a random path crosses the interfaces at x_j , a two-point discrete random variable ξ taking the values 0, and 1, with probability 1/2 is generated. For a random path crossing the interface at x_j , and coming from the segment j to $j+1$, if $\xi = 0$ then the function $b_{j+1}(x_j, t)$ is evaluated. This in practice requires generating a random trajectory according to the properties of segment $j+1$, evolving now through the segment $j+1$. On the contrary, if $\xi = 1$ then the function $f_j(x_j, t)$ is evaluated. For a random path this corresponds to be reflected back to the same segment j . The same holds for the other condition regarding the function $f_{j+1}(x_j, t)$, but now the scenario corresponds to random paths crossing the interface at x_j , and coming from the segment $j+1$ to the segment j . The Fig. 8 shows again the perfect agreement between the solution obtained with the

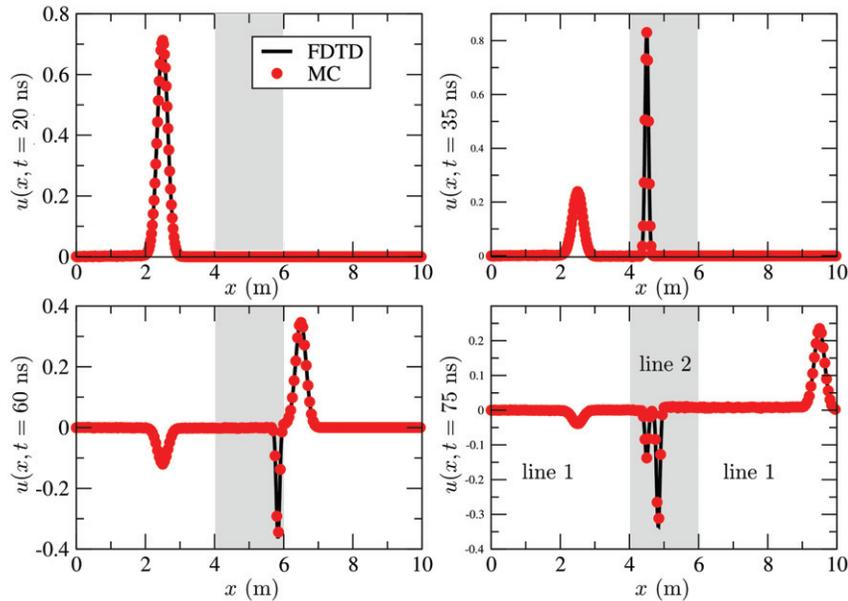


Fig. 8. Transmission line composed of three segments with different electromagnetic characteristics. The middle segment (in grey) has a characteristic inductance $L = 2.250 \mu\text{H}/\text{m}$ (line 2) and the two other segments have characteristic inductance $L = 0.250 \mu\text{H}/\text{m}$. The other simulation parameters are the same as in the previous simulation. The line is terminated with a resistance of value $R_L = 100 \Omega$, and the voltage source is a gaussian pulse, with $t_0 = 7.5 \text{ ns}$, $s_0 = 0.75 \text{ ns}$.

Monte Carlo method, and the FDTD method for any values of time.

As it was already anticipated, the Monte Carlo method seems to be free of any numerical dispersion and dissipative issues. This is clearly different from any finite-difference based numerical schemes, such as the classical FDTD method. The main reason is found in how the continuous partial differential equations are discretized.

Recall that for the Monte Carlo method no discretization in the temporal and spatial variables is done, being the number of paths the only discretization parameter. To illustrate this remarkable feature, in Fig. 9 it is shown the results corresponding to the numerical simulation of the line with a raised cosine voltage source given by

$$v_g(t) = \begin{cases} 1, & \text{if } |t - T| \leq \xi_- \\ \frac{1}{2} \left[1 + \cos \left(\frac{\pi}{\beta T} [|t - T| - \xi_-] \right) \right], & \text{if } \xi_- < |t - T| < \xi_+ \\ 0, & \text{if } |t - T| > \xi_+, \end{cases} \quad (28)$$

where $\xi_{\pm} = (1 \pm \beta)T/2$. The parameter β controls the roughness of the on/off transitions of the signal in the time domain and can take values between zero and one.

The raised cosine signal turns out to be specially suited to observe a known phenomena in FDTD simulations called "ringing" due to dispersion which can be easily observed when β goes to zero. For the FDTD simulations in Fig. 9 we have considered two values of the time step: 1) $c\Delta t = 0.99\Delta x$, which is close to the so called "magic time step", and 2) $c\Delta t = 0.5\Delta x$. The magic time step was avoided on purpose because the numerical method becomes unstable for this particular signal and parameters. Moreover, it has been proved in [13] that for the telegraph equations, there is no "magic time step size" for lossy media discretized by the Yee's FDTD method. The lack of numerical stability, even satisfying theoretically the CFL condition, was already pointed out in [20] for the case of the one-dimensional Maxwell equations in free space. Moreover, according to the practical suggestions mentioned in [7], as a rule-of-thumb, it is recommended to choose values of the time step close to the CFL limit as possible. Apart from the results obtained with the Monte Carlo method, and the FDTD method, the theoretical solution obtained by the analytical procedure described above is shown in Fig. 9. Note the very good agreement between the Monte Carlo method and the theoretical solution. On the contrary, the FDTD method (for both time step values considered) exhibits the mentioned "ringing", being more pronounced for $c\Delta t = 0.5\Delta x$, as theoretically expected. For the Monte Carlo method, the number of points where the solution was computed, and the number of paths, were chosen to make approximately equal the computational time spent by the FDTD method with $c\Delta t = 0.99\Delta x$. In closing, it holds that for about the same computational cost, the Monte Carlo method seems to be more accurate than the FDTD method.

5 Conclusion

A new probabilistic method to solve the one-dimensional telegraph equations has been presented. Mathematically, what distinguishes this algorithm is the fact that the classical Kac's theory, and subsequent works based on this theory, were developed analyzing merely the single second order partial differential equation, while the method presented here is based on the corresponding

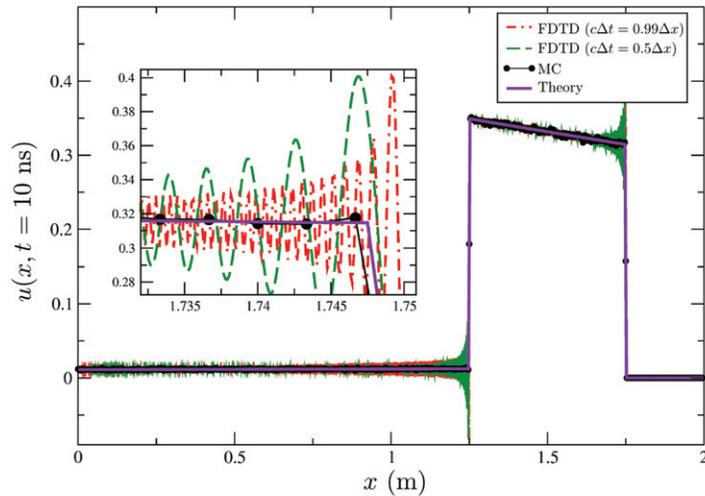


Fig. 9. Comparison between the numerical results obtained with the Monte Carlo method (MC), the FDTD method, and the theoretical solution, for a raised cosine pulse as a voltage source. For the FDTD method two different values of Δt has been used, one close to the well known "magic time step", and other far from it. Parameters of the line are $l = 2$ m, $L = 0.250$ $\mu\text{H}/\text{m}$, $C = 100$ pF/m, $R = 12.5$ Ω/m , $G = 0.5$ mS/m, $R_L = 12.5$ Ω , $R_g = 75$ Ω , and for the pulse, $T = 5$ ns, and $\beta = 10^{-6}$. For the MC method, $N = 10^5$.

system of first order differential equations. One of the main advantages of this new approach is that it can be used to obtain a numerical algorithm more efficient compared with that obtained with classical Kac's theory. In fact, the direct use of the classical theory seems to be somewhat unsuitable from a computational point of view, since it requires a further discretization in time, which introduces a new source of error, and consequently degrades the overall performance of the underlying numerical algorithm. Furthermore, this new approach has been shown to be particularly suitable to solve bounded problems and we have shown how to incorporate general boundary conditions in the method.

The algorithm was validated successfully through the simulation of electromagnetic wave propagation in transmission lines for voltage sources with different spectral characteristics. The results were compared with those obtained using the classical FDTD method and theoretical solutions using Fourier analysis. We have successfully demonstrated the applicability of our algorithm to the case of boundary conditions involving non-resistive loads and transmission line heterogeneities to model wave propagation in multiple mediums.

Apart from the well-known advantages of probabilistic methods, such as be-

ing naturally parallel and meshless, the probabilistic method proposed here does not exhibit numerical dispersion and dissipation features. This is because no approximation of any derivatives of the solution was done and so the method is not based on any sort of spatial and temporal discretization, as in finite difference methods. The meshless nature of the method allows to compute the solution at single points in space and time efficiently, which is specially advantageous for monitoring the signal evolution at specific places of the transmission line, without needing the values of the voltage and current in all the intermediate points. However, in view of the slow rate of convergence of any Monte Carlo method, it may be not so convenient to be used for computing the solution in the whole line. As an alternative the method can be combined with the FDTD method through a domain decomposition method. This idea, called probabilistic domain decomposition method [2], was already proposed in literature for different problems, but can also be applied, adapting conveniently, for the telegraph equations.

Important is also to remark that for high frequency and large scale problems, the computational performance of our Monte Carlo-based method overcomes the performance of the classical FDTD method for one-dimensional problems. This is highly unexpected given that Monte Carlo methods are considered rather inefficient for low dimensions. On the other hand, for high dimensional problems, Monte Carlo methods typically outperform deterministic mesh-based methods in view of the well known "curse of dimensionality". In fact, the computational cost of such methods grows exponentially with the dimension of the problem while is independent of the dimensions for the Monte Carlo methods, see e.g.[10]. This encourages to think that the generalization of the algorithm for multidimensional problems will perform even better than the algorithm proposed here for one-dimensional problems. Such an extension is planned to be accomplished using the *random flights* proposed recently in [8,11]. In fact, here it was shown that there exists indeed a probabilistic solution of the n -dimensional telegraph equation, where now the random trajectories are obtained generating a Poisson random time, along with a uniform distribution. For each Poisson event the trajectory follows a direction in space choosing randomly the orientation from a uniform distribution. Generalizing this approach to deal with boundary conditions is currently a work in progress to be submitted elsewhere.

Acknowledgments

This work was partially supported by Fundação para a Ciência e a Tecnologia under Grants No. PEst-OE/EEI/LA0021/2013 and PEst-OE/EEI/LA0008/2013. The authors gratefully acknowledge fruitful discussions with Paulo Nunes and Luís Ducla Soares.

References

- [1] J.A. Acebrón, and R. Spigler, Supercomputing applications to the numerical modeling of industrial and applied mathematics problems, *J. Supercomput.*, 40 (2007) 67-80.
- [2] J.A. Acebrón, M.P. Busico, P. Lanucara, and R. Spigler, Probabilistically induced domain decomposition methods for elliptic boundary-value problems *Journal of Computational Physics*, 210 (2005) 421-438.
- [3] J.A. Acebrón, and A. Rodríguez-Rozas, A new parallel solver suited for arbitrary semilinear parabolic partial differential equations based on generalized random trees, *Journal of Computational Physics*, 230 (2011) 7891-7909.
- [4] J.A. Acebrón, and A. Rodríguez-Rozas, Highly efficient numerical algorithm based on random trees for accelerating parallel Vlasov-Poisson simulations, *Journal of Computational Physics*, 250 (2013) 224-245.
- [5] B.V. Budaev, and D.V. Bogy, Novel solutions of the Helmholtz equation and their application to diffraction, *Proc. R. Soc. A*, 463 (2007) 1005-1027.
- [6] Y.L.L. Coz, H.J. Greub, and R.B. Iverso, Performance of random walk capacitance extractors for IC interconnects: A numerical study, *Solid-State Electron.*, 42 (1998) 581-588.
- [7] D.B. Davidson, Computational electromagnetics for RF and microwave engineering, *CRC press*, 2005.
- [8] A. De Gregorio, and E. Orsingher, Flying randomly in \mathbf{R}^d with Dirichlet displacements, *Stochastic Processes and their Applications*, 122 (2011) 676-713.
- [9] I.T. Dimov, Monte Carlo Methods for Applied Scientists, World Scientific, 2008.
- [10] M. Evans, and T. Swartz, Approximating Integrals Via Monte Carlo and Deterministic Methods, Oxford University Press, 2000.
- [11] R. Garra, and E. Orsingher, Random flights governed by Klein-Gordon-type partial differential equations, *Stochastic Processes and their Applications*, 124 (2014) 2171-2187.
- [12] S. Golstein, On diffusion by discontinuous movements, and on the Telegraph equation, *Quart. J. Mech. Appl. Math.*, IV (1951) 129-156.
- [13] S. Guilin, and C.W. Trueman, Numerical Dispersion and Numerical Loss in Explicit Finite-Difference Time-Domain Methods in Lossy Media, *IEEE Transactions on Antennas and Propagation*, 53 (2005) 3684-3690 .
- [14] M.T. Heath, Scientific Computing: An Introductory Survey, McGraw Hill, 1997.

- [15] R. Janaswamy, On random time and on the relation between wave and telegraph equations, *IEEE Transactions on Antennas and Propagation*, 61 (2013) 2735-2744.
- [16] M. Kac, Random walk and the theory of Brownian motion, *Math. Month.*, 54 (1947) 369-391.
- [17] M. Kac, A stochastic model related to the telegrapher's equation, *Rocky Mountain J. Math*, 43 (1974) 479-509.
- [18] I. Karatzas, and S.E. Shreve, Brownian Motion and Stochastic Calculus, *Springer*, 1991.
- [19] R.M.M. Mattheij, S.W. Rienstra, and J.H.M. ten Thije Boonkamp, Partial Differential Equations: Modeling, Analysis, Computation, *SIAM monographs*, 2005.
- [20] M.S. Min, and C.H. Teng, The instability of the Yee scheme for the magic time step, *Journal of Computational Physics*, 166 (2001) 418-424.
- [21] E. Orsingher, Probability law, flow function, maximum distribution of wave governed random motions and their connections with Kirchoff's laws, *Stochastic Processes and their Applications*, 34 (1990) 49-66.
- [22] R. Vilela Mendes, Poisson-Vlasov in a strong magnetic field: A stochastic solution approach, *Journal of Mathematical Physics*, 51(4) (2010) 043101
- [23] R. Vilela Mendes, Stochastic solutions of some nonlinear partial differential equations, *Stochastics An International Journal of Probability and Stochastic Processes*, 81 (2009) 279-297.
- [24] M.N.O. Sadiku, Monte Carlo methods for electromagnetics, *CRC press*, 2009.
- [25] M.N.O. Sadiku, Numerical Techniques in electromagnetics, *CRC Press*, 2001.