

# A multiscale numerical method for the heterogeneous cable equation

Alexandre L. Madureira<sup>a,\*</sup>, Daniele Q.M. Madureira<sup>a</sup>, Pedro O. Pinheiro<sup>b</sup>

<sup>a</sup> Departamento de Matemática Aplicada, Laboratório Nacional de Computação Científica, Av. Getúlio Vargas 333, 25651-070 Petrópolis – RJ, Brazil

<sup>b</sup> Institut National des Sciences Appliquées de Lyon 20, avenue Albert Einstein – 69621 Villeurbanne Cedex, France

## ARTICLE INFO

### Article history:

Received 29 October 2010

Received in revised form

3 August 2011

Accepted 13 August 2011

Communicated by I. Bojak

Available online 27 August 2011

### Keywords:

Multiscale

Numerical methods

Cable equation

Dendrites modeling

## ABSTRACT

Several interesting problems in neuroscience are of multiscale type, i.e. possess different temporal and spatial scales that cannot be disregarded. Such characteristics impose severe burden to numerical simulations since the need to resolve small scale features pushes the computational costs to unreasonable levels. Classical numerical methods that do not resolve the small scales suffer from spurious oscillations and lack of precision.

This paper presents an innovative numerical method of multiscale type that ameliorates these maladies. As an example we consider the case of a cable equation modeling heterogeneous dendrites. Our method is not only easy to parallelize, but it is also nodally exact, i.e., it matches the values of the exact solution at every node of the discretization mesh, for a class of problems.

To show the validity of our scheme under different physiological regimes, we describe how the model behaves whenever the dendrites are thin or long, or the longitudinal conductance is small. We also consider the case of a large number of synapses and of large or low membrane conductance.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Among the fields of research that computer models can hope to deliver significant contributions, neuroscience is one of the most demanding and beautiful. Among the main aspects that make neuroscience so challenging from the modeling and computational point of view are the multiple temporal and spatial scales present in most neurological events. Some instances of what is steadily being considered in present research are attempts to increase the size of networks of spiking neurons, as well as incorporation of spatial and heterogeneous aspects of the neuron physiology. Although the ever increasing capabilities of computers facilitate such endeavors, a big chunk of the advances is certainly due to better modeling and computational techniques.

Several facets of multiple scales and neuroscience pervade the literature, from general discussions on modeling [11,25,38], to topics related to numerical and computational aspects [2,3,8,22,27–30,40–42,45,49]. Powerful mathematical tools from asymptotic analysis, like homogenization, matching asymptotics, singular perturbation, and dimension reduction techniques are commonly employed, along with various degrees of neurological based arguments [4,5,7,17,34,36,39,37,44,46].

We focus here on the task of deriving an efficient numerical method for problems of multiscale type. The problem we consider involves modeling dendrites with an arbitrary distribution of synapses. In a recent important paper, Meunier and d'Incamps [37] considered various instances of heterogeneous dendrites modeled by variants of cable equations, and inquired when the process of *mathematical homogenization* was valid. Homogenization [9] is a mathematical technique that, given a detailed model describing all the physical properties of the dendrites, “replaces” it by a homogeneous, simpler to solve, equation. It is like replacing a heterogeneous dendrite by a homogeneous one that has similar physical behavior. There is a catch nevertheless; such substitution is valid only *as the number of heterogeneities approach infinity*. That is not the whole story, since the heterogeneity must have some sort of “pattern”, being periodic or random for instance. Such assumptions are often questionable [3,37].

Motivated by such concerns, we investigate the cable equation considered in [37], this time from a numerical point of view. Such a model is fully motivated and derived in [47,48]. Let the voltage  $\hat{V}$  be the solution of the cable equation

$$c_m \frac{\partial \hat{V}}{\partial t} - \frac{\sigma^\ell d}{4} \frac{\partial^2 \hat{V}}{\partial x^2} + \sigma^m \hat{V} + \hat{\sigma}^{\text{in}}(\hat{V} - V^{\text{in}}) + \hat{\sigma}^{\text{ex}}(\hat{V} - V^{\text{ex}}) = 0$$

$$\text{in } (0, L) \times (0, +\infty),$$

$$\hat{V}(0, t) = \hat{V}(L, t) = 0 \quad \text{for } t \in (0, +\infty),$$

$$\hat{V}(x, 0) = \hat{V}_0(x) \quad \text{for } x \in (0, L). \quad (1)$$

\* Corresponding author. Tel.: +55 24 2233 6116; fax: +55 24 2233 6124.

E-mail addresses: alm@lnc.cbr (A.L. Madureira),

daniele@lnc.cbr (D.Q.M. Madureira), pedro.oliveira-pinheiro@insa-lyon.fr (P.O. Pinheiro).

The reason for choosing the killed-end (Dirichlet) boundary conditions in (1) is because they are harder to solve numerically than the sealed-end (Neumann) condition. This is so since, for problems like the one considered here, killed-end conditions force abrupt changes in the solution, also known as *layers*, that are refractory to computational approximations [43].

Above,  $\hat{V}_0$  is the initial condition,  $d$  denotes the dendrite diameter in centimeters (cm),  $c_m$  is the specific membrane capacitance in farad per square centimeter (F/cm<sup>2</sup>),  $\sigma^\ell$  denotes the longitudinal dendrite specific conductance in siemens per centimeter (S/cm),  $\sigma^m$  denotes the membrane specific conductance in siemens per square centimeter (S/cm<sup>2</sup>), and  $\hat{\sigma}^{\text{ex}}$  and  $\hat{\sigma}^{\text{in}}$  are the excitatory and inhibitory synapse specific conductances, also in siemens per square centimeter (S/cm<sup>2</sup>). We assume that the specific conductances  $\sigma^\ell$  and  $\sigma^m$  are constant. The potentials  $\hat{V}$ ,  $V^{\text{ex}}$  and  $V^{\text{in}}$  are in millivolt (mV), and both reversal potentials  $V^{\text{ex}}$  and  $V^{\text{in}}$  are constant. Finally,  $L$  is the dendrite length in centimeter (cm). The synapses are modeled by

$$\hat{\sigma}^{\text{in}} = \sum_{l=1}^{N^i} g_l^i \delta_{\hat{x}_l^i}, \quad \hat{\sigma}^{\text{ex}} = \sum_{l=1}^{N^e} g_l^e \delta_{\hat{x}_l^e},$$

where  $\delta_{\hat{x}_l^i}$  are Dirac deltas (or delta “functions”) located at the synapses sites  $\hat{x}_l^i$  with strengths  $g_l^i$ , for  $l=1, \dots, N^i$ . Similar notation holds for the excitatory synapses, i.e., the deltas  $\delta_{\hat{x}_l^e}$  are located the synapse sites  $\hat{x}_l^e$  with strengths  $g_l^e$ , for  $l=1, \dots, N^e$ . The definition of a Dirac delta  $\delta_{\hat{x}^*}$  located at a point  $\hat{x}^* \in (0, L)$  is that

$$\int_0^L \delta_{\hat{x}^*} g(\hat{x}) d\hat{x} = g(\hat{x}^*) \quad (2)$$

for any continuous function  $g$ ; see [23].

For equations like (1), several different numerically demanding instances might show up, as big differences in the strengths of the synapses, large number of synapses at arbitrary locations, and even a high ratio between the diameter of the dendrite and its length. Under these circumstances, the computational costs involved in solving such problems can be unacceptable if a raw numerical method is to be considered, in particular when considering a large tree of dendrites where each branch is modeled by (1). The computational costs grow since *any* method employed has to account for the microscale aspects of the dendrite physiology.

In the computational neuroscience literature, discretization of spatial features of partial differential equations traditionally employs compartmental models and difference schemes. On the other hand, finite elements are seldom employed. This is unfortunate since methods based on finite elements are flexible, simple to implement, computationally efficient, and easier to analyze. For “nice” problems, when the solution has a smooth behavior and there are no numerical complications, finite elements and finite differences yield comparable results. However, when standard schemes do not work well, modern variants of finite elements come as a viable option of discretization [1,12,13,18,19,26].

One variant is the multiscale finite element method (MsFEM) [13–16,31,32,35], which we explore here. The idea behind multiscale methods to solve heterogeneous systems is that one has to first solve some *local problems* and extract some *microscale information*. Such information is then *upscaled* into a *homogenized* macroscale problem. Microscale problems depend on refined information of the model, but has to be solved in *small domains*, and parallelization is trivial. Due to this local feature, they are not expensive to solve. In contrast, the homogenized macroscale problem is global and has to be solved in the whole domain. But the microscale data show up averaged, i.e., *homogenized*, and the cost of solving the homogenized macroscale problem is independent of the microscales. The overall result is a method that is not only accurate, but also computationally cheaper in

terms of costs, see [32] for a careful investigation regarding the computational efficiency of the method. As a bonus, for steady problems in one-dimensional domains, the MsFEM yields a *nodally exact solution*, i.e., the numerical solution matches the exact solution at each nodal point (see Section 3 for a proof).

A previous application of the MsFEM include flows in heterogeneous media [13–16,31], a class of problem of major interest for the oil industry. Indeed, oil fields can have the scale of kilometers, while the oil might flow within porous rocks in the centimeter scale. This is an instance where a full discretization of all the physical details is completely out of reach, and techniques based on the MsFEM offered a new tool to tackle such transport problems.

Another application involves modeling equations posed on domains with rough boundaries. This problem is of interest since wrinkles, paradoxically at first sight, can help to minimize the flow drag over surfaces—think of a golf ball or a shark skin. However, the characteristic length of the domain and of the rugosities are of different scales, and refined discretizations become unreasonably expensive. The MsFEM offers a way to reduce computational costs [35].

We consider here a thin dendrite with synapses distributed along its extension. The voltage will jump at the synapses locations, and since traditional methods must use a huge number of grid points to capture such small scale behavior, they quickly become impractical. On the other hand, the MsFEM global problem uses a fixed, possibly small, number of grid points, independent of the number of synapses and the thickness of the dendrite. Between each two consecutive grid points, a subgrid is created and smaller local problems are solved, possibly in parallel, and microscale information are uploaded to the global problem that is solved afterwards. We emphasize that the size of the final global problem is independent of all physiological parameters.

We now describe briefly the contents of the present paper. In the next two sections, we present the basics of the traditional and multiscale finite element method applied to (1). Next, in Section 4 we analyze the behavior of the model under different limit situations, and display some numerical results. In Section 5, we briefly consider two extra applications, namely, a Y-shaped domain and a transient problem. We present our conclusions in Section 6.

## 2. Classical finite element method

To facilitate the dissection of the main properties of the problem under consideration, it is convenient first to define new coordinates  $x = \hat{x}/L$ , and rewrite (1) as

$$\tau_m \frac{\partial V}{\partial t} - \varepsilon \frac{\partial^2 V}{\partial x^2} + V + GV = f \quad \text{in } (0, 1) \times (0, +\infty),$$

$$V(0, t) = V(1, t) = 0 \quad \text{for } t \in (0, +\infty),$$

$$V(x, 0) = V_0(x) \quad \text{for } x \in (0, 1), \quad (3)$$

where  $V(x, t) = \hat{V}(Lx, t)$ ,  $V_0(x) = \hat{V}_0(Lx)$ ,

$$\tau_m = \frac{c_m}{\sigma^m}, \quad \varepsilon = \frac{\sigma^\ell d}{4L^2 \sigma^m}, \quad G = \frac{\sigma^{\text{in}} + \sigma^{\text{ex}}}{\sigma^m}, \quad f = \frac{\sigma^{\text{in}} V^{\text{in}} + \sigma^{\text{ex}} V^{\text{ex}}}{\sigma^m}.$$

We also have

$$\sigma^{\text{in}} = \sum_{l=1}^{N^i} g_l^i \delta_{x_l^i}, \quad \sigma^{\text{ex}} = \sum_{l=1}^{N^e} g_l^e \delta_{x_l^e}, \quad (4)$$

where the Dirac deltas are now located at the sites  $x_l^i = \hat{x}_l^i/L$ , and  $x_l^e = \hat{x}_l^e/L$ .

The finite element method is based on a *variational formulation*, obtained by multiplying both sides of (3) by a sufficiently

smooth function  $w$  that vanishes at 0 and 1 (functions in the Sobolev space  $H_0^1(0, 1)$ , in mathematical terms). After an integration by parts in  $(0, 1)$ , it follows that

$$\int_0^1 \tau_m \frac{\partial V}{\partial t} w \, dx + \int_0^1 \left( \varepsilon \frac{\partial V}{\partial x} \frac{\partial w}{\partial x} + Vw + GVw \right) dx = \int_0^1 fw \, dx. \quad (5)$$

Actually, the problems (5) and (3) are equivalent. Using (4), and the definition of Dirac deltas (2), it follows that

$$\begin{aligned} \int_0^1 GVw \, dx &= \int_0^1 \left( \sum_{l=1}^{N_i} g_l^i \delta_{x_l^i} + \sum_{l=1}^{N_e} g_l^e \delta_{x_l^e} \right) \frac{Vw}{\sigma^m} dx \\ &= \sum_{l=1}^{N_i} \frac{g_l^i V(x_l^i) w(x_l^i)}{\sigma^m} + \sum_{l=1}^{N_e} \frac{g_l^e V(x_l^e) w(x_l^e)}{\sigma^m}, \end{aligned}$$

$$\begin{aligned} \int_0^1 fw \, dx &= \int_0^1 \left( V^{\text{in}} \sum_{l=1}^{N_i} g_l^i \delta_{x_l^i} + V^{\text{ex}} \sum_{l=1}^{N_e} g_l^e \delta_{x_l^e} \right) \frac{w}{\sigma^m} dx \\ &= V^{\text{in}} \sum_{l=1}^{N_i} \frac{g_l^i w(x_l^i)}{\sigma^m} + V^{\text{ex}} \sum_{l=1}^{N_e} \frac{g_l^e w(x_l^e)}{\sigma^m}. \end{aligned}$$

As in finite difference schemes, we partition the domain  $(0, 1)$  in  $N+1$  open intervals (a.k.a. *elements*)  $(x_0, x_1), (x_1, x_2), \dots, (x_{N-1}, x_N), (x_N, x_{N+1})$ , defined by the nodes

$$0 = x_0 < x_1 < x_2 < x_3 < \dots < x_N < x_{N+1} = 1. \quad (6)$$

In a uniform partition,  $x_k = kh$  for  $k=0, \dots, N+1$ , where  $h=1/(N+1)$ .

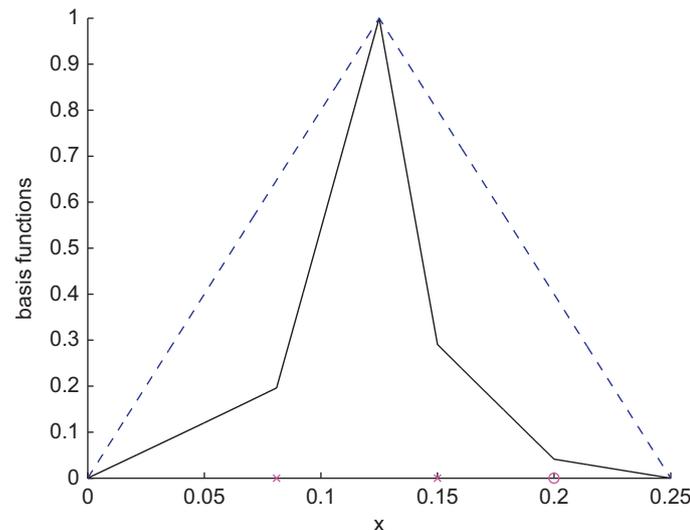
The next step is to assume that  $V$  can be approximated by a *piecewise linear function*, i.e.,

$$V(x, t) \approx V_h(x, t) = \sum_{k=1}^N V_k(t) \psi_k(x), \quad (7)$$

where the unknowns  $V_1, \dots, V_N$  depend on  $t$  only, and  $\psi_k$  is a *basis function* that is linear within each element, continuous, and such that

$$\psi_k(x_j) = \delta_{kj} = \begin{cases} 1 & \text{if } k=j, \\ 0 & \text{if } k \neq j. \end{cases} \quad (8)$$

See a typical basis function depicted in blue in Fig. 1. Note that since  $\psi_k$  vanishes at all nodes except  $x_k$ , it follows that  $V_h(x_j, t) = V_j(t)$ . For simplicity, assume further that  $V_0(x) = \sum_{k=1}^N V_k^0 \psi_k(x)$  for some constants  $V_1^0, \dots, V_N^0$ .



To compute the unknowns, we use (5) replacing  $V$  by  $V_h$  and  $w$  by  $\psi_j$ , obtaining the  $N$ -dimensional system of ordinary differential equations (ODEs) given as follows:

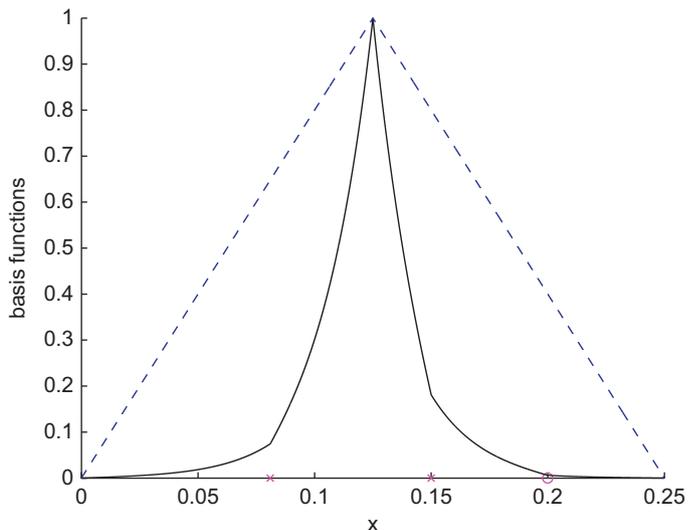
$$\begin{aligned} \sum_{k=1}^N \tau_m V_k \int_0^1 \psi_k(x) \psi_j(x) \, dx \\ + V_k \int_0^1 \left[ \varepsilon \frac{\partial \psi_k(x)}{\partial x} \frac{\partial \psi_j(x)}{\partial x} + \psi_k(x) \psi_j(x) + G \psi_k(x) \psi_j(x) \right] dx \\ = \int_0^1 f \psi_j(x) \, dx, \end{aligned} \quad (9)$$

$$V_j(0) = V_j^0$$

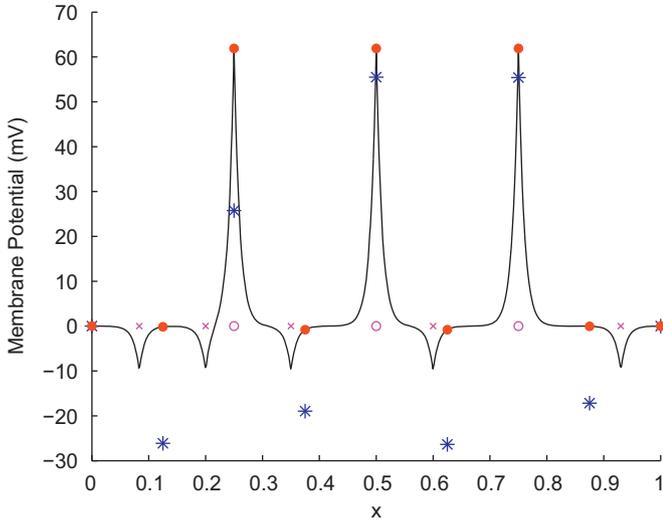
for  $j=1, \dots, N$ . Thus, to find the approximation (7), it suffices to solve (9). The method just described is commonly called *method of lines*, and it approximates the partial differential evolution equation by a system of ODEs.

**Remark.** The method of lines described above is used quite often in finite elements, but there are other possibilities. Less popular alternatives for the method of lines include time discontinuous Galerkin method, and Rothe's method, also known as *the horizontal method of lines* [33]. In the first case, the finite element scheme is developed in a space-time domain, using discontinuous Galerkin techniques to handle the time derivative. Such scheme hopes to inherit some of the nice properties of the finite element methods. However, due to its discontinuous nature, the method turns out cumbersome to develop and implement, and so far seems more restricted to the legions of finite element fans. In Rothe's method, the time discretization is performed *before* the domain discretization. It seems that such idea never received much attention from the scientific community, but there are recent attempts to use the scheme combined with modern spatial discretization [20,24]. Both schemes lack the simplicity of the method of lines, which also profits from the huge literature and knowledge on numerical schemes for ODEs developed over the last few decades.

**Remark.** Although we restrict ourselves to the killed-end boundary conditions, we ought to mention that the implementation of sealed-end boundary conditions in finite elements is straightforward, even for general multidimensional domains.



**Fig. 1.** Typical functions in the region  $(0, 2h)$ , for  $h=0.125$ . There are two inhibitory synapses (at  $x=0.08$  and  $x=0.15$ , marked with  $\times$ , and an excitatory synapse (at  $x=0.2$ , marked with  $\odot$ ). The figures were obtained using  $\varepsilon$  large (left) and small (right). The classical basis functions (dashed line) depend only on the size of the elements, while the multiscala basis functions (continuous line) depend on the size of the elements and also on the physiological parameters.



**Fig. 2.** Steady state solutions for small  $\varepsilon (=10^{-4})$ . The loci of the synapses are marked with  $\times$  (inhibitory) and  $\circ$  (excitatory). Note that the classical finite element solution (asterisks) fails to yield a reasonable approximation to the exact solution (solid line), while the multiscale solution (dots) is nodally exact.

**Remark.** Another advantage of finite element formulations, of particular importance in the present work, is the evaluations of delta functions. Indeed, since finite elements are based on formulas that involve integrals like (5), it becomes natural to compute the action of the Dirac deltas, as defined in (2).

The trouble with the scheme described above is that it fails to work well under all parameter regimes. For instance if  $\varepsilon$  is much smaller than the mesh size, then spurious oscillations show up in a numerical solution that turns out to be unreliable (see for instance the example considered in Section 4.1, with numerical results depicted in Fig. 2). The same bad behavior holds for finite difference schemes as well [6]. This is not a problem of *how* one obtains the discretization (finite element or difference), but of discretizations that fail to capture the small scale behavior of the solution.

### 3. The multiscale finite element method (MsFEM)

We propose here an alternative scheme, still based on a finite element approach, but with modified basis functions. As already hinted, the idea behind the MsFEM is that the basis functions should carry all the microscale information, and that shall be done by solving local problems. We next define the method, and then get into the details of how to compute the microscale solutions and upscale the related information.

#### 3.1. The definition of the method

Still partitioning the domain with the mesh (6), we approximate  $V$  by the multiscale function  $V_h^{ms}$  defined by

$$V_h^{ms}(x,t) = \sum_{k=1}^N V_k^{ms}(t)\lambda_k(x). \quad (10)$$

As before, the unknowns  $V_1^{ms}(t), \dots, V_N^{ms}(t)$  depend on  $t$  only. The new basis functions  $\lambda_1, \dots, \lambda_N$  are still continuous, but instead of being linear within each element, they satisfy the local,

elementwise problems

$$\lambda_k(x) = 0 \quad \text{if } x \notin (x_{k-1}, x_{k+1}),$$

$$-\varepsilon \frac{\partial^2 \lambda_k}{\partial x^2} + \lambda_k + G\lambda_k = 0 \quad \text{in } (x_{k-1}, x_k) \text{ and } (x_k, x_{k+1}),$$

$$\lambda_k(x_k) = 1 \quad (11)$$

for  $k \in \{1, \dots, N\}$ .

In a few particular cases it is possible to compute  $\lambda_k$  explicitly, but in general it is necessary to approximate it numerically. In Fig. 1 we depict two typical basis functions. In both examples, the support of  $\lambda_k$ , i.e. the region where  $\lambda_k$  is nonzero, is assembled by joining two consecutive elements,  $(0,0.125)$  and  $(0.125,0.25)$ , and it contains two inhibitory synapses at  $x=0.08$  and  $x=0.15$ , and an excitatory synapse at  $x=0.2$ . The left figure was obtained for large  $\varepsilon$ , and the right one, for small  $\varepsilon$ .

It is worth pointing out that these basis functions adapt and capture the local physiological heterogeneities effects of the dendrites. If there are synapses, the functions have jumps in their derivatives, as the exact solution of the original problem does. If  $\varepsilon$  is small, the functions have an exponential profile, just like the solution of the original problem. That is how the upscaling process occurs. Note that in classical elements, the functions would be piecewise linear, regardless of the parameters.

As long as the multiscale basis functions  $\lambda_k$  are computed, from (11), the unknowns  $V_1^{ms}, \dots, V_N^{ms}$  are defined by (compare with the classical (9)):

$$\begin{aligned} & \sum_{k=1}^N \tau_m \frac{dV_k^{ms}}{dt} \int_0^1 \lambda_k(x)\lambda_j(x) dx \\ & + V_k^{ms} \int_0^1 \left[ \varepsilon \frac{\partial \lambda_k(x)}{\partial x} \frac{\partial \lambda_j(x)}{\partial x} + \lambda_k(x)\lambda_j(x) + G\lambda_k(x)\lambda_j(x) \right] dx \\ & = \int_0^1 f\lambda_j(x) dx, \\ V_j^{ms}(0) & = V_0(x_j) \end{aligned} \quad (12)$$

for  $j=1, \dots, N$ . With such choice of basis functions, it is now possible to have an accurate method with the size of the system (12) independent of  $\varepsilon$  and the number of synapses. The task of incorporating the microstructure, where the synapses play a direct role and rise the costs, is concentrated in the computation of the basis functions. That is the subject of the next subsection.

As we already remarked, a striking property of multiscale methods is that, for steady state problems in one-dimensional domains, the numerical solution yields the exact solution at every node. Such property follows from the very definition of the method, i.e. from (11) and (12). The proof is quite simple, and we include the main arguments below. Note that what we want to prove is that the interpolant of  $V$  given by  $I_h V = \sum_{k=1}^N V(x_k)\lambda_k$  solves

$$\begin{aligned} & \int_0^1 \left[ \varepsilon \frac{\partial I_h V}{\partial x} \frac{\partial \lambda_j}{\partial x} + I_h V \lambda_j + G I_h V \lambda_j \right] dx \\ & = \int_0^1 f \lambda_j dx = \int_0^1 \left[ \varepsilon \frac{\partial V}{\partial x} \frac{\partial \lambda_j}{\partial x} + V \lambda_j + G V \lambda_j \right] dx \\ & \text{for } j = 1, \dots, N. \end{aligned} \quad (13)$$

From an integration by parts, (11), and the identity  $I_h V = V$  at every node, it follows that

$$\int_{x_{i-1}}^{x_i} \left[ \varepsilon \frac{\partial (I_h V - V)}{\partial x} \frac{\partial \lambda_j}{\partial x} + (I_h V - V)\lambda_j + G(I_h V - V)\lambda_j \right] dx = 0$$

at every element  $(x_{i-1}, x_i)$ . Thus, (13) holds.

Note in the proof above, that we used the fact that  $\lambda_j$  solves (11) exactly. In fact, for most multiscale methods, local problems have to be solved numerically, and it is a matter of discussion in

the multiscale community how well such problems need to be resolved.

### 3.2. Finding the basis functions

To find the functions  $\lambda_k$  it is necessary to solve (11). Note that the equations are the same, but posed in different elements. For simplicity, we deal here only with problem (11) posed on a fixed element  $(x_{k-1}, x_k)$ .

First, let  $x_1^\mu < x_2^\mu < x_3^\mu < \dots < x_{N^\mu}^\mu$  be the locations of the  $N^\mu$  synapses lying inside the element  $(x_{k-1}, x_k)$ . Let the submesh  $x_{k-1} = x_0^\mu < x_1^\mu < x_2^\mu < x_3^\mu < \dots < x_{N^\mu}^\mu < x_{N^\mu+1}^\mu = x_k$ , and define for  $n = 1, \dots, N^\mu$ ,

$$\phi_n^\mu(x) = \begin{cases} \frac{\sinh\left(\frac{x-x_{n-1}^\mu}{\sqrt{\varepsilon}}\right)}{\sinh\left(\frac{x_n^\mu-x_{n-1}^\mu}{\sqrt{\varepsilon}}\right)} & \text{if } x \in [x_{n-1}^\mu, x_n^\mu], \\ \frac{\sinh\left(\frac{x_{n+1}^\mu-x}{\sqrt{\varepsilon}}\right)}{\sinh\left(\frac{x_{n+1}^\mu-x_n^\mu}{\sqrt{\varepsilon}}\right)} & \text{if } x \in (x_n^\mu, x_{n+1}^\mu], \\ 0 & \text{otherwise,} \end{cases}$$

$$\phi_{N^\mu+1}^\mu(x) = \begin{cases} \frac{\sinh\left(\frac{x-x_{N^\mu}^\mu}{\sqrt{\varepsilon}}\right)}{\sinh\left(\frac{x_{N^\mu+1}^\mu-x_{N^\mu}^\mu}{\sqrt{\varepsilon}}\right)} & \text{if } x \in (x_{N^\mu}^\mu, x_{N^\mu+1}^\mu], \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Observe that  $\phi_n^\mu$  solves

$$-\varepsilon \frac{\partial^2 \phi_n^\mu}{\partial x^2} + \phi_n^\mu = 0 \quad \text{em } (x_{n-1}^\mu, x_n^\mu) \cup (x_n^\mu, x_{n+1}^\mu),$$

$$\phi_n^\mu(x_{n-1}^\mu) = \phi_n^\mu(x_{n+1}^\mu) = 0, \quad \phi_n^\mu(x_n^\mu) = 1,$$

and thus it is a Green's function (fundamental solution) for the reaction–diffusion problem above in  $(x_{k-1}, x_k)$ . Hence, we seek  $\lambda_k$  such that its restriction to  $(x_{k-1}, x_k)$  is given by

$$\lambda_k = \sum_{m=1}^{N^\mu} c_m \phi_m^\mu + \phi_{N^\mu+1}^\mu \quad (15)$$

solving the variational form of (11), i.e.,  $c_m$  must be determined for  $m = 1, \dots, N^\mu$  by

$$\int_{x_{k-1}}^{x_k} \varepsilon \frac{\partial \lambda_k}{\partial x} \frac{\partial \phi_j^\mu}{\partial x} + \lambda_k \phi_j^\mu + G \lambda_k \phi_j^\mu \, dx = 0 \quad \text{for } j = 1, \dots, N^\mu. \quad (16)$$

Defining the matrices  $A^\mu = S^\mu + M^\mu$  where

$$S_{ij}^\mu = \int_{x_{k-1}}^{x_k} \varepsilon \frac{\partial \phi_i^\mu}{\partial x} \frac{\partial \phi_j^\mu}{\partial x} + \phi_i^\mu \phi_j^\mu \, dx, \quad M_{ij}^\mu = \int_{x_{k-1}}^{x_k} G \phi_i^\mu \phi_j^\mu \, dx$$

we gather that  $\mathbf{c} = [c_1, \dots, c_{N^\mu}]$  solve  $A^\mu \mathbf{c} = \mathbf{f}$ , where  $\mathbf{f} = [f_1, \dots, f_{N^\mu}]$ , and

$$f_1 = \dots = f_{N^\mu-1} = 0, \quad f_{N^\mu} = \frac{-\sqrt{\varepsilon}}{\sinh\left(\frac{x_{N^\mu+1}^\mu-x_{N^\mu}^\mu}{\sqrt{\varepsilon}}\right)}.$$

From (14), we conclude that  $S^\mu$  is symmetric and tridiagonal, that  $M^\mu$  is diagonal, and

$$S_{j-1,j}^\mu = S_{j,j-1}^\mu = \frac{-\sqrt{\varepsilon}}{\sinh\left(\frac{x_j^\mu-x_{j-1}^\mu}{\sqrt{\varepsilon}}\right)} \quad \text{for } j = 2, \dots, N^\mu,$$

$$S_{j,j}^\mu = \sqrt{\varepsilon} \left[ \frac{\cosh\left(\frac{x_j^\mu-x_{j-1}^\mu}{\sqrt{\varepsilon}}\right)}{\sinh\left(\frac{x_j^\mu-x_{j-1}^\mu}{\sqrt{\varepsilon}}\right)} + \frac{\cosh\left(\frac{x_{j+1}^\mu-x_j^\mu}{\sqrt{\varepsilon}}\right)}{\sinh\left(\frac{x_{j+1}^\mu-x_j^\mu}{\sqrt{\varepsilon}}\right)} \right] \quad \text{for } j = 1, \dots, N^\mu,$$

$$M_{ij}^\mu = \frac{g_j}{\sigma^m} \delta_{ij} \quad \text{for } i, j = 1, \dots, N^\mu.$$

In the above definition of  $M^\mu$ , we denote  $g_j$  as being either  $g_j^i$  or  $g_j^e$ , depending on whether the synapse at  $x_j^\mu$  is inhibitory or excitatory.

We should point out that the solution given by (15) is actually exact. This (surprising?) result comes from a cumbersome computation that involves splitting (16) using the subgrid elements, and integrations by parts in each subelement.

A worth remark regarding the MsFEM concerns its computational efficiency, as compared with traditional methods. The gain comes from the general principle that it is cheaper to solve several small decoupled problems than one single big task, even when considering serial computation.

Although the construction of the basis functions is fully decoupled, it introduces an overhead cost. On the bright side, such construction is performed only once for a given set of physiological parameters, and it results in a final system of reduced size.

Consider a discretization of the elliptic, one dendrite case, with  $N$  elements and  $M$  subcell elements—in our case  $M$  is the number of synapses per element. So, the traditional method computer memory grows as  $O(MN)$ , whereas the MsFEM requires  $O(M+N)$ . In terms of operation count, traditional methods behave as  $O(MN)$ , and the MsFEM as  $O(N+MN)$ , if a solver with linear cost is employed (e.g., a multigrid solver). For solvers that are polynomial in time, such as direct solvers, then operation count for traditional methods scales as  $O(M^\gamma N^\gamma)$ , but the MsFEM scales as  $O(N^\gamma + NM^\gamma)$ , where  $\gamma$  is the solver operation count rate of growth. Typically,  $\gamma = 3$  for LU decomposition [21]. Note that in the multiple dendrite case the matrix associated with finite element or difference discretization has increased bandwidth, but this effect is mitigated in the MsFEM. The increased bandwidth deteriorates the performance of a number of linear algebra methods.

Even though the operation counts for both traditional (finite elements or differences) and MsFEM are asymptotically similar if multigrid methods are employed, the number of multigrid iterations depends on the problem and the discretization under consideration. The use of MsFEM can actually reduce the number of iterations necessary for convergence [32].

Our comments above hold under the assumption that serial computers are used. For parallel computations the efficiency gains are even more striking. We refer again to [32], where a careful numerical investigation of the performance of the methods was conducted.

## 4. Different regimes and their solutions

As pointed out previously, varying the different parameters in (1) leads to different neurological regimes. In many instances this causes spurious oscillatory behavior in numerical computations. To understand the different behaviors that show up, it is useful to perform an analysis using (3). We consider here only the non-transient problem.

As an example, consider the case when  $\varepsilon$  is much smaller than one ( $\varepsilon \ll 1$ ). This happens for instance when the dendrite is too thin or too long ( $d \ll L^2$ ), when the longitudinal conductance is too big ( $\sigma^l \gg \sigma^m$ ), or when a combination of the above instances occur. In such cases numerical difficulties appear, as we shall see. Another different situation is when there is a huge number of

synapses, or when the membrane conductance is either too large or too low. In what follows, we present separate formal studies of these *asymptotic limits* and show how the classical and multiscale finite elements perform. To exalt the effects of each separate situation, we isolate each one of them using parameters that are not necessarily biologically plausible.

#### 4.1. Long or thin dendrites, or small longitudinal conductance

One situation where numerical difficulties occur is when the parameter  $\varepsilon$  is too small. In terms of physiology of the dendrites there are many instances when this can happen, as described above. However, regardless of the origins, the numerical outcomes are the same.

To find out how the solution  $V$  depends on the parameter  $\varepsilon$ , we use the method of matching asymptotics, postulating that

$$V(x) \sim V_0(x) + \varepsilon V_1(x) + \varepsilon^2 V_2(x) + \dots, \tag{17}$$

where the functions  $V_j$  are to be determined. Formally replacing (17) in (3), we gather that

$$V_0 + GV_0 + \varepsilon \left( -\frac{\partial^2 V_0}{\partial x^2} + V_1 + GV_1 \right) + \varepsilon^2 \left( -\frac{\partial^2 V_1}{\partial x^2} + V_2 + GV_2 \right) + \dots = f \quad \text{in } (0, 1).$$

Collecting the  $\varepsilon = 0$  limit terms, it follows that  $V_0 + GV_0 = f$ , i.e.,

$$\sigma^m V_0 + \left( \sum_{l=1}^{N^i} g_l^i \delta_{x_l^i} + \sum_{l=1}^{N^e} g_l^e \delta_{x_l^e} \right) V_0 = \sum_{l=1}^{N^i} g_l^i \delta_{x_l^i} V^{\text{in}} + \sum_{l=1}^{N^e} g_l^e \delta_{x_l^e} V^{\text{ex}}.$$

After a multiplication by an arbitrary and smooth function  $\phi$ , and an integration, the above equation becomes

$$\int_0^1 \sigma^m V_0 \phi \, dx + \sum_{l=1}^{N^i} g_l^i V_0(x_l^i) \phi(x_l^i) + \sum_{l=1}^{N^e} g_l^e V_0(x_l^e) \phi(x_l^e) = V^{\text{in}} \sum_{l=1}^{N^i} g_l^i \phi(x_l^i) + V^{\text{ex}} \sum_{l=1}^{N^e} g_l^e \phi(x_l^e).$$

By considering special functions  $\phi$  (actually, a sequence of them), it is possible to prove that

$$V_0(x) = \begin{cases} 0 & \text{if } x \notin \{x_1^i, \dots, x_{N^i}^i, x_1^e, \dots, x_{N^e}^e\}, \\ V^{\text{in}} & \text{if } x \in \{x_1^i, \dots, x_{N^i}^i\}, \\ V^{\text{ex}} & \text{if } x \in \{x_1^e, \dots, x_{N^e}^e\}. \end{cases}$$

Thus, as  $\varepsilon \rightarrow 0$ , the exact solution  $V$  approaches the discontinuous function  $V_0$ . Since  $V$  itself is continuous, there is an onset of internal layers at the points of discontinuity. As we remark in the Introduction, layers are regions with large derivatives, in this case, regions around the synapses where the voltage “jumps”. These layers cause severe numerical trouble [43].

Such behavior of the exact solution does not come as a surprise. Indeed the neurological meaning of “ $\varepsilon$  small” is that there is relatively little diffusion of ions, as occurs when the dendrite is thin, or the longitudinal conductance is small. In such instance, the electric “jumps” that take place at the synapses concentrate in a narrow neighborhood of the synaptic loci.

As a numerical test, depicted in Fig. 2, we pick an example where five inhibitory (locations marked with  $\times$ ) and three excitatory (locations marked with  $\odot$ ) synapses are disposed along the dendrite,  $V^{\text{in}} = -10$ ,  $V^{\text{ex}} = 65$ ,  $g_l^i = 4 \times 10^{-2}$ ,  $g_l^e = 10^{-2}$ , and  $\varepsilon = 10^{-4}$ . In this example, and all that follow, an “exact solution” (displayed in solid black line) is numerically computed by “overkill”, using a numerical method with a sufficiently refined mesh. We solve the same problem using classical (computed nodal values marked by blue asterisks) and multiscale finite element

methods (computed nodal values marked by red dots), with nine nodal points in both cases.

We first comment on the exact solution. Observe that it is close to zero except in a small neighborhood of the synapses. Over the synapses the value of the exact solution is close to either  $V^{\text{in}} = -10$  or  $V^{\text{ex}} = 65$ . This confirms our theoretical prediction that, whenever  $\varepsilon$  is small,  $V$  should be close to  $V_0$ . Regarding the numerical aspects, the classical method yields a solution that is essentially wrong, where the multiscale solution matches the exact solution at every node, as predicted by the theory. Sure enough, if sufficient points are used in the classical scheme, we would eventually obtain a reasonable approximation. For instance, for the present example, 129 points are necessary to bring the relative errors within a range of 10% error at every node.

#### 4.2. Large number of synapses

Suppose that  $N^i = N^e$ , and let  $\alpha = 1/(2N^i)$ . Assume further that the synapses are disposed periodically, i.e., the Dirac deltas are located at the sites  $x_l^i = (2l-1)\alpha$  and  $x_l^e = 2l\alpha$ . In the present case, the interest is when the synapses are narrowly packed, i.e.  $\alpha \ll 1$ , and this situation is tricky to analyze.

The idea is to rewrite the solution of (3) as the minimizer of the energy

$$J(V) + \alpha^{-1} I_\alpha(V), \tag{18}$$

where

$$J(V) = \frac{1}{2} \int_0^1 \varepsilon \sigma^m \left( \frac{\partial V}{\partial x} \right)^2 + \sigma^m V^2 \, dx,$$

$$I_\alpha(V) = \frac{\alpha}{2} \sum_{l=1}^{N^i} g_l^i V^2(x_l^i) + \frac{\alpha}{2} \sum_{l=1}^{N^e} g_l^e V^2(x_l^e) - \alpha V^{\text{in}} \sum_{l=1}^{N^i} g_l^i V(x_l^i) - \alpha V^{\text{ex}} \sum_{l=1}^{N^e} g_l^e V(x_l^e).$$

We assume now some reasonable conditions on  $g_l^i$  and  $g_l^e$ , for instance that  $g_l^i = g^i(x_l^i)$  and  $g_l^e = g^e(x_l^e)$ , where the functions  $g^i$  and  $g^e$  are defined in  $(0,1)$  and have at most a finite number of discontinuities. Then, as  $\alpha \rightarrow 0$  the term  $I_\alpha$  concentrates most of the total energy. Thus  $\lim_{\alpha \rightarrow 0} V = W_0$  in a reasonable mathematical sense (i.e.,  $\lim_{\alpha \rightarrow 0} \int_0^1 (V - W_0)^2 \, dx = 0$ ), where  $W_0$  minimizes  $\lim_{\alpha \rightarrow 0} I_\alpha$ , and it turns out that

$$W_0 = \frac{V^{\text{in}} g_l^i + V^{\text{ex}} g_l^e}{g_l^i + g_l^e} \tag{19}$$

does the job.

As a numerical test, we consider the case of 350 inhibitory and excitatory synapses,  $g_l^i = 4 \times 10^{-2}$ ,  $g_l^e = 10^{-2}$ ,  $V^{\text{in}} = -10$ , and  $V^{\text{ex}} = 65$ . In this case,  $W_0 = 5$ , and that is exactly the number around which the solution oscillates. Numerically, we consider 10 nodal points for both methods. Note in Fig. 3 that the classical method (blue asterisks) oscillates close to the boundaries, but delivers a reasonable approximation in the interior of the domain. The multiscale method (red dots) is nodally exact, as it should be.

#### 4.3. Large or low membrane conductance

In this part, we analyze the effects of the membrane conductance on the solution, i.e., how  $V$  behaves as  $\sigma^m \rightarrow \infty$  or  $\sigma^m \rightarrow 0$ . We define the set  $S = \{\sigma^d d / (4L^2), g_1^i, \dots, g_{N^i}^i, g_1^e, \dots, g_{N^e}^e\}$ , and assume  $\sigma^m \gg \max S$  or  $\sigma^m \ll \min S$ . It turns out that both situations are somewhat simple.

The former case occur when the membrane is too “diffusive”, allowing the transmembrane free flow of ions. Thus, the  $V=0$

limit comes as no surprise, and can be formally obtained by making  $\sigma^m \rightarrow \infty$  in the steady state version of (3):

$$-\frac{\sigma^l d}{4L^2 \sigma^m} \frac{\partial^2 V}{\partial x^2} + V + \frac{\sigma^{in} + \sigma^{ex}}{\sigma^m} V = \frac{\sigma^{in} V^{in} + \sigma^{ex} V^{ex}}{\sigma^m} \quad \text{in } (0, 1),$$

$$V(0) = V(1) = 0.$$

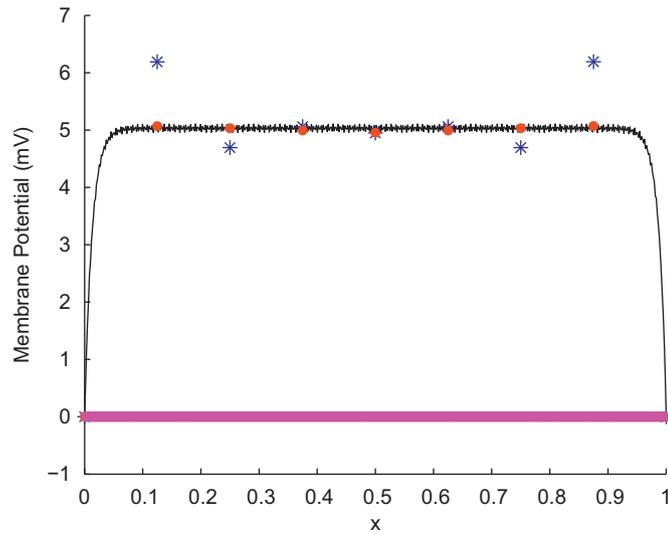
In the latter situation, as  $\sigma^m \rightarrow 0$ ,  $V$  satisfies

$$-\frac{\sigma^l d}{4L^2} \frac{\partial^2 V}{\partial x^2} + (\sigma^{in} + \sigma^{ex}) V = \sigma^{in} V^{in} + \sigma^{ex} V^{ex} \quad \text{in } (0, 1),$$

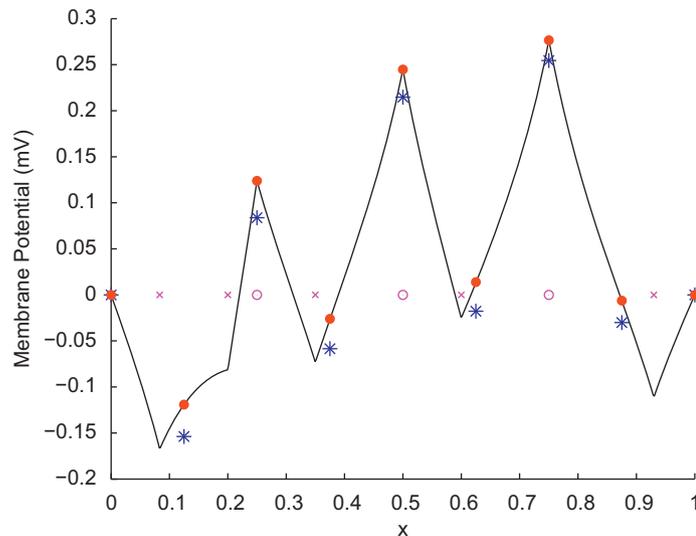
$$V(0) = V(1) = 0 \tag{20}$$

reflecting the fact that the cross membrane flow of ions occur only through the synapses.

We plot both cases in Fig. 4. The figure in the left was obtained with  $\max S = 10^{-2}$  and  $\sigma^m = 1$ , and for the one in the right,



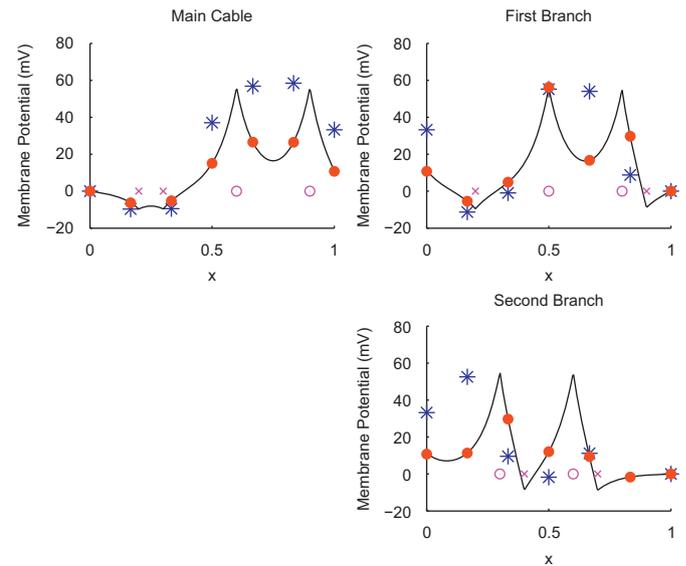
**Fig. 3.** Numerical test with a large number of synapses. As predicted by the theory, the exact solution (solid line) is close to  $W_0$  (formula (19)). The classical method (asterisks) oscillates close to the boundaries, but approximates well the exact solution in the interior of the domain. The multiscale method (dots) is nodally exact.



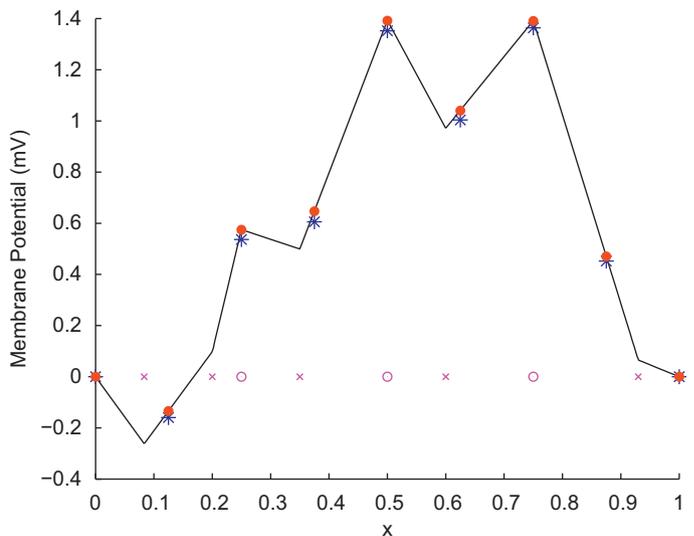
$\min S = 10^{-2}$ , and  $\sigma^m = 10^{-3}$ . Note that, for the sake of clarity, the plots have different scales. Indeed, as shown in the left figure, the magnitude of the solution is small, as predicted by the theory. It is interesting to observe the good performance (at last!) of the traditional method. This is simply because in the asymptotic regime described by (20), the exact solution is a combination of piecewise linear functions.

### 5. Further numerical tests

We test here two cases which are prototypes of more interesting practical situations: networks of dendrites, and time dependent problems. Although the tests presented here clearly do not intend to fully investigate all computational and



**Fig. 5.** Steady state solution in a Y-shaped domain. The classical approximation (asterisks) suffers again from spurious oscillations, whereas the MsFEM (dots), although no longer provably nodally exact, yields an excellent approximation to the exact solution (solid line).



**Fig. 4.** Numerical experiments for large (left) and small (right) membrane conductance. As  $\sigma^m$  grows, the exact solution  $V$  (solid line) approaches the zero function, and as  $\sigma^m$  decreases,  $V$  converges to a piecewise linear function. In both cases, the MsFEM (dots) is exact at the nodes, and in the latter case, the classical finite element approximation (asterisks) results reasonable.

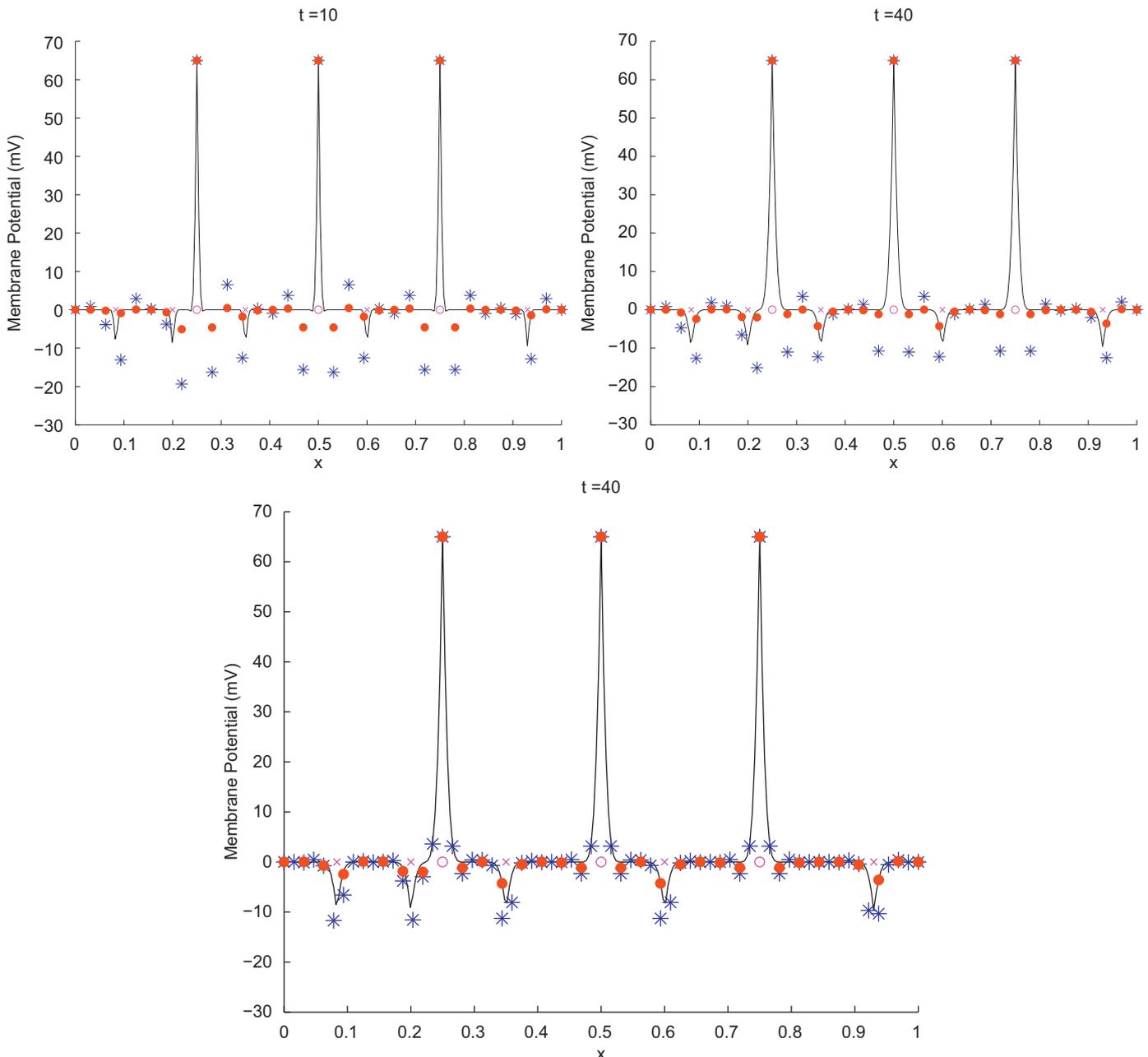
theoretical aspects of both situations, they indicate that the MsFEM can handle these more sophisticated examples.

We first consider the steady state version of (1) in a Y-shaped domain. We added a nodal point at the branching point, and impose there continuity of voltage and current [47]. Note that such node is the end point of three different elements, one belonging to each branch. Thus, the support of the basis function associated to such node intersects the three branches, and three equations of the form (11) have to be solved, each with parameters coming from the different branches.

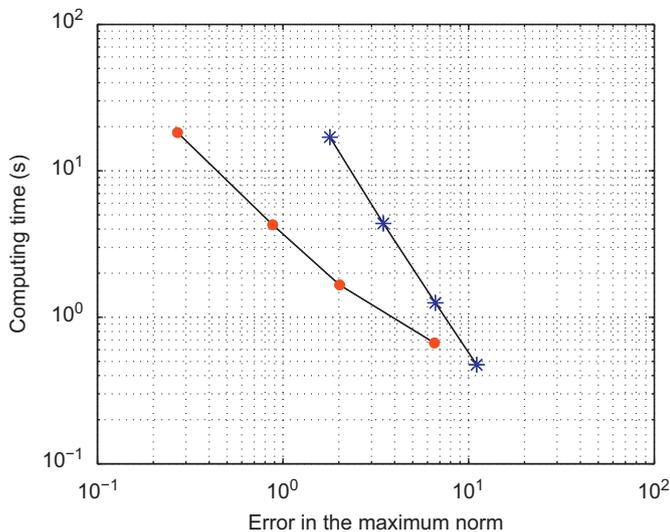
The MsFEM is no longer provably nodally exact, but it still yields an excellent approximation for the exact solution. In Fig. 5,  $V^{\text{in}} = -10$ ,  $V^{\text{ex}} = 65$ ,  $L = 0.2$ ,  $d = 0.01$ ,  $\sigma^{\ell} = 10^{-3}$  and  $\sigma^{\text{m}} = 10^{-2}$  for all the branches, and there two excitatory and two inhibitory synapses in each of the branches. We used a total of 19 nodal

points for each method. There is one nodal value for the classical method missing: the classical scheme yielded a value below  $-20$ , quite far from the exact solution (black solid line). It is notable that few points give excellent accuracy, and that allows for great efficiency when several branches are coupled.

We next consider a transient problem, with the parameters being the same as in Section 4.1 and  $\tau_m = 10^{-2}$  s, see [10]. After discretizing in space and obtaining a system of ordinary equations, we use backward Euler's method for time discretization, yielding an implicit scheme. In all tests we consider the time step equal to  $10^{-3}$ . In our first test, we use 33 mesh points for both the classical and multiscale method. At time  $t = 10$  s (top left of Fig. 6), both methods exhibit some oscillatory behavior, which is much more severe for the classical scheme. At  $t = 40$  s, the multiscale scheme is already quite accurate, while the classical method still oscillates



**Fig. 6.** Snap shots of solutions for the transient problem (exact solution in solid line). At time  $t = 10$  s (top left), both classical (asterisks) and MsFEM (dots) approximations suffer from spurious oscillations. At  $t = 40$  s (top right), the MsFEM is accurate, while the classical method still oscillates. Doubling the number of nodes for the classical scheme yields a reasonable solution (bottom graphic), although taking more than two times longer than the MsFEM.



**Fig. 7.** Log–log plot of computing times for the transient problem at time 10 s. The x-axis displays the error in the maximum norm, and the y-axis displays the computing time for the classical (asterisks) and multiscale (dots) methods. For a fixed desired tolerance, the multiscale method needs less points, and its computing time is significantly smaller than for the classical methods.

(top right of Fig. 6). Doubling the number of nodes for the classical scheme yields a reasonable solution (bottom of Fig. 6).

In terms of computing time in a personal computer, in the above tests with the final time equal to 40 s, when both methods use 33 points, it took 0.7 s for the classical method and 0.8 s for the multiscale method. Considering the classical scheme with 65 points, it took 1.8 s. Note that the MsFEM is more time consuming than the classical method, if the same number of nodal points is used. This is so due to the need to construct the basis functions. However, since most of the computing time is spent on the time evolution process, which has similar cost for both methods, the final difference is not significant. But this is not the whole story, since, to obtain an equivalent accuracy, the classical method requires more points than the MsFEM. Refining the mesh in the classical scheme improves accuracy, but significantly deteriorates its performance. To further illustrate that point we consider the computing times corresponding to a given approximation error, for each method.

Given the exact solution  $V$ , consider the classical finite element approximation  $V_h$ , a fixed time  $T$ , and compute the approximation error in the maximum norm. The maximum error norm measures the largest approximation error among the nodes, and is given by

$$\max_{k=1,\dots,N} |V(x_k, T) - V_h(x_k, T)|.$$

The same procedure is performed with the multiscale approximation  $V_h^{\text{ms}}$  replacing  $V_h$ .

Let  $T = 10$  s,  $\tau_m = 10^{-2}$  s, and  $2^j + 1$  nodes for  $j = 6, \dots, 9$ , and keep all other parameters as in the case of Section 4.1. In Fig. 7, the x-axis displays the error in the maximum norm, and the y-axis displays the computing time for the classical (blue asterisks) and multiscale (red dots) methods. Note that, for a desired tolerance, the multiscale method computing times are significantly smaller than for the classical methods since the MsFEM requires less points.

## 6. Conclusions

Several models in neuroscience are of multiscale type. Classical numerical methods do not deal with them in a natural way, but rather require brute force, also known as refined

discretization. This is necessary to capture essential physiological details. We present here a viable numerical alternative.

Our method cuts computational costs since it solves the microscale details as a *pre-processing* task, and that can be done in parallel. In particular very few mesh points are necessary to obtain accurate results, and this can be important if involved problems are considered, for instance for large neuron trees. The numerical upscaling procedure is done on the fly, and does not demand any special physiological characteristics. It fits well neuroscience problems having cost as an issue.

The problem we consider here depends in a nontrivial way on several parameters. In our case by case analysis, we show how the solutions rely on them. In practice, such extreme and isolated situations are unlikely. Biologically plausible examples exhibit actually a *combination* of these effects, in an attenuated fashion, but when using classical numerical approximations that combination might lead a disastrous net effect. Homogenization techniques can help in some regimes, but they are reliable only under very specific circumstances. On the other hand, our method is always robust with respect to different scales of the parameters, accurate, and easy to parallelize.

For transient problems, the multiscale method is still very accurate, but it exhibits some short term oscillatory behavior under certain conditions, as time starts to evolve. Its performance is still much better than that of classical methods, but some more involved space-time multiscale methods are currently under investigation, see [20] for instance. These new methods should deliver fast and reliable computations, qualities that are in high demand for tough computational neuroscience problems.

## Acknowledgments

The first author gratefully acknowledges the hospitality of the Institute for Mathematics and its Applications (IMA) during the New Directions Short Course in Mathematical Neuroscience (2008). He also thanks the partial financial support of CNPq, Grant numbers 560108/2010-9 and 308670/2007-8, and also FAPERJ, Grant number E-26/102.255/2009. The second author thanks the financial support from CNPq (PCI/LNCC and Grant number 560108/2010-9).

## References

- [1] P.F. Antonietti, F. Brezzi, L.D. Marini, Bubble stabilization of discontinuous Galerkin methods, *Comput. Methods Appl. Mech. Eng.* 198 (21–26) (2009) 1651–1659 MR 2517937.
- [2] S.M. Baer, S. Crook, Dur-E-Ahmad, Z. Jackiewicz, Numerical solution of calcium-mediated dendritic branch model, *J. Comput. Appl. Math.* 229 (2) (2009) 416–424. doi:10.1016/j.cam.2008.04.011.
- [3] J.M. Bower, D. Beeman, *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural Simulation System* <<http://www.genesis-sim.org/GENESIS/>>, 2003.
- [4] P.C. Bressloff, Traveling fronts and wave propagation failure in an inhomogeneous neural network, *Physica D* 155 (1–2) (2001) 83–100 MR 1837205 (2002d:92001).
- [5] P.C. Bressloff, B.A. Earnshaw, M.J. Ward, Diffusion of protein receptors on a cylindrical dendritic membrane with partially absorbing traps, *SIAM J. Appl. Math.* 68 (5) (2008) 1223–1246 MR 2407121 (2009g:92017).
- [6] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* 32 (1–3) (1982) 199–259 FENOMECH '81, Part I (Stuttgart, 1981), MR 679322 (83k:76005).
- [7] D. Cai, L. Tao, A.V. Rangan, D.W. McLaughlin, Kinetic theory for neuronal network dynamics, *Commun. Math. Sci.* 4 (1) (2006) 97–127 MR 2204080 (2007a:82053).
- [8] N. Carnevale, M.L. Hines, *The NEURON Book*, Cambridge University Press, 2006.
- [9] D. Cioranescu, P. Donato, *An introduction to homogenization*, Oxford Lecture Series in Mathematics and its Applications, vol. 17, The Clarendon Press, Oxford University Press, New York, 1999 MR 1765047 (2001j:35019).
- [10] P. Dayan, L.F. Abbott, *Computational and mathematical modeling of neural systems*, computational neuroscience, in: *Theoretical Neuroscience*, MIT Press, Cambridge, MA, 2001 MR 1985615 (2004g:92008).

- [11] E. De Schutter, Computational neuroscience: more math is needed to understand the human brain, in: *Mathematics Unlimited—2001 and Beyond*, Springer, Berlin, 2001, pp. 381–391. MR 1852166.
- [12] E. Weinan, B. Engquist, Multiscale modeling and computation, *Not. Am. Math. Soc.* 50 (9) (2003) 1062–1070 MR 2002752 (2004m:65163).
- [13] Y. Efendiev, T.Y. Hou, Theory and applications, *Multiscale Finite Element Methods, Surveys and Tutorials in the Applied Mathematical Sciences*, vol. 4, Springer, New York, 2009 MR 2477579.
- [14] Y. Efendiev, T.Y. Hou, Multiscale computations for flow and transport in heterogeneous media, *Quantum transport, Lecture Notes in Mathematics*, vol. 1946, Springer, Berlin, 2008, pp. 169–248 MR 2497877.
- [15] Y. Efendiev, A. Pankov, Numerical homogenization of monotone elliptic operators, *Multiscale Model. Simul.* 2 (1) (2003) 62–79 (electronic). MR 2044957 (2005a:65153).
- [16] Y.R. Efendiev, X.-H. Wu, Multiscale finite element for problems with highly oscillatory coefficients, *Numer. Math.* 90 (3) (2002) 459–486 MR 1884226 (2002m:65114).
- [17] B. Ermentrout, Neural networks as spatial-temporal pattern-forming systems, *Rep. Prog. Phys.* 61 (1998) 353–430.
- [18] L.P. Franca, A.L. Madureira, F. Valentin, Towards multiscale functions: enriching finite element spaces with local but not bubble-like functions, *Comput. Methods Appl. Mech. Eng.* 194 (27–29) (2005) 3006–3021 MR 2142535 (2006a:65159).
- [19] L.P. Franca, A.L. Madureira, L. Tobiska, F. Valentin, Convergence analysis of a multiscale finite element method for singularly perturbed problems, *Multiscale Model. Simul.* 4 (3) (2005) 839–866. doi:10.1137/040608490 (electronic). MR 2203943 (2006k:65316).
- [20] L.P. Franca, J.V.A. Ramalho, F. Valentin, Enriched finite element methods for unsteady reaction-diffusion problems, *Commun. Numer. Methods Eng.* 22 (6) (2006) 619–625. doi:10.1002/cnm.838 MR 2235032.
- [21] G.H. Golub, C.F. Van Loan, *Matrix computations*, Johns Hopkins Series in the Mathematical Sciences, vol. 3, Johns Hopkins University Press, Baltimore, MD, 1983 MR 733103 (85h:65063).
- [22] D. Hansel, G. Mato, C. Meunier, L. Neltner, On numerical simulations of integrate-and-fire neural networks, *Neural Comput.* 10 (2) (1998) 467–483.
- [23] D.D. Haroske, H. Triebel, *Distributions, Sobolev spaces, elliptic equations*, in: EMS Textbooks in Mathematics, European Mathematical Society EMS, Zürich, 2008 MR 2375667 (2009a:46003).
- [24] C.A.A. Henao, A.L.G.A. Coutinho, L.P. Franca, A stabilized method for transient transport equations, *Comput. Mech.* 46 (1) (2010) 199–204. doi:10.1007/s00466-010-0465-5 MR 2644409.
- [25] A.V.M. Herz, T. Gollisch, C.K. Machens, D. Jaeger, Modeling single-neuron dynamics and computations: a balance of detail and abstraction, *Science* 314 (5796) (2006) 80–85 MR 2253402 (2007d:92020).
- [26] J.S. Hesthaven, T. Warburton, *Nodal discontinuous Galerkin methods, Texts in Applied Mathematics, Algorithms, Analysis, and Applications*, vol. 54, Springer, New York, 2008 MR 2372235 (2008k:65002).
- [27] M. Hines, Efficient computation of branched nerve equations, *Int. J. Bio-Med. Comput.* 15 (1984) 69–76.
- [28] M.L. Hines, N.T. Carnevale, The NEURON simulation environment, *Neural Comput.* 9 (6) (1997) 1179–1209.
- [29] M.L. Hines, H. Eichner, F. Schürmann, Neuron splitting in compute-bound parallel network simulations enables runtime scaling with twice as many processors, *J. Comput. Neurosci.* 25 (1) (2008) 203–210. doi:10.1007/s10827-007-0073-3.
- [30] M.L. Hines, H. Markram, F. Schürmann, Fully implicit parallel simulation of single neurons, *J. Comput. Neurosci.* 25 (2008) 439–448. doi:10.1007/s10827-008-0087-5.
- [31] T.Y. Hou, Numerical approximations to multiscale solutions in partial differential equations, *Universitext*, in: *Frontiers in Numerical Analysis (Durham 2002)*, Springer, Berlin, 2003, pp. 241–301 MR 2006969 (2004m:65219).
- [32] T.Y. Hou, X.-H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, *J. Comput. Phys.* 134 (1) (1997) 169–189. doi:10.1006/jcph.1997.5682 MR 1455261 (98e:73132).
- [33] P. Knabner, L. Angermann, *Numerical methods for elliptic and parabolic partial differential equations, Texts in Applied Mathematics*, vol. 44, Springer-Verlag, New York, 2003 MR 1988268 (2004j:65002).
- [34] C.R. Laing, T.A. Frewen, I.G. Kevrekidis, Coarse-grained dynamics of an activity bump in a neural field model, *Nonlinearity* 20 (2007) 2127–2146. doi:10.1088/0951-7715/20/9/007.
- [35] A.L. Madureira, A multiscale finite element method for partial differential equations posed in domains with rough boundaries, *Math. Comput.* 78 (265) (2009) 25–34 MR 2448695.
- [36] D. McLaughlin, R. Shapley, M. Shelley, Large-scale modeling of the primary visual cortex: influence of cortical architecture upon neuronal response, *J. Physiol.—Paris* 97 (2003) 237–252.
- [37] C. Meunier, B.L. d'Incamps, Extending cable theory to heterogeneous dendrites, *Neural Comput.* 20 (7) (2008) 1732–1775 MR 2417105 (2009e:92021).
- [38] C. Meunier, I. Segev, Playing the devil's advocate: is the Hodgkin–Huxley model useful? *Trends Neurosci.* 25 (11) (2002) 558–563.
- [39] A. Omurtag, B.W. Knight, L. Sirocich, On the simulation of large populations of neurons, *J. Comput. Neurosci.* 8 (8) (2000) 51–63.
- [40] A.V. Rangan, D. Cai, Fast numerical methods for simulating large-scale integrate-and-fire neuronal networks, *J. Comput. Neurosci.* 22 (2007) 81–100. doi:10.1007/s10827-006-8526-7.
- [41] M.J. Rempe, D.L. Chopp, A predictor-corrector algorithm for reaction-diffusion equations associated with neural activity on branched structures, *SIAM J. Sci. Comput.* 28 (6) (2006) 2139–2161 (electronic). MR 2272255 (2008f:65148).
- [42] M.J. Rempe, N. Spruston, W.L. Kath, D.L. Chopp, Compartmental neural simulations with spatial adaptivity, *J. Comput. Neurosci.* 25 (2008) 465–480. doi:10.1007/s10827-008-0089-3.
- [43] H.-G. Roos, M. Stynes, L. Tobiska, *Numerical methods for singularly perturbed differential equations, Springer Series in Computational Mathematics*, vol. 24, Springer-Verlag, Berlin, 1996 Convection-diffusion and flow problems. MR 1477665 (99a:65134).
- [44] J. Rubin, M. Wechselberger, The selection of mixed-mode oscillations in a Hodgkin–Huxley model with multiple timescales, *Chaos* 18 (1) (2008) 015105 MR 2404661 (2009a:37194).
- [45] M.J. Shelley, L. Tao, Efficient and accurate time-stepping schemes for integrate-and-fire neuronal networks, *J. Comput. Neurosci.* 11 (2001) 111–119.
- [46] M.J. Shelley, D.W. McLaughlin, Coarse-grained reduction and analysis of a network model of cortical response: i drifting grating stimuli, *J. Comput. Neurosci.* 12 (2002) 97–122.
- [47] H.C. Tuckwell, *Linear cable theory and dendritic structure, Introduction to theoretical neurobiology*, vol. 1, Cambridge Studies in Mathematical Biology, vol. 8, Cambridge University Press, Cambridge, 1988 MR 947344 (90a:92003a).
- [48] H.C. Tuckwell, *Nonlinear and stochastic theories, Introduction to theoretical neurobiology*, vol. 2, Cambridge Studies in Mathematical Biology, vol. 8, Cambridge University Press, Cambridge, 1988 MR 947345 (90a:92003b).
- [49] Y. Zhuang, A parallel and efficient algorithm for multicompartment neuronal modelling, *Neurocomputing* 69 (10–12) (2006) 1035–1038. doi:10.1016/j.neucom.2005.12.040.



**Alexandre L. Madureira** is a researcher at the Laboratório Nacional de Computação Científica (LNCC) in Brazil. He received his Ph.D. in Mathematics from The Pennsylvania State University in 1999, and his investigation concerns analysis and modeling of partial differential equations that depend nontrivially on parameters.



**Daniele Q.M. Madureira** holds a Postdoc position in Computational Neuroscience at the Laboratório Nacional de Computação Científica (LNCC) in Brazil. She received her D.Sc. in Computer Science from COPPE-UFRRJ, and her research involves computational models in neuropsychiatry and musical cognition.



**Pedro Oliveira Pinheiro** is an Eng. student at Institut National des Sciences Appliquées (INSA), Lyon, France, currently doing a M2 master on the specialisation “Control, Systems, Image and Telecommunication”. During the summer of 2010, he worked at Laboratório Nacional de Computação Científica (LNCC) in Brazil with Dr. Madureira. His interests include applied mathematics, information theory and signal processing.