Journal of Computational Physics 292 (2015) 114–140

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A second-order Cartesian method for the simulation of electropermeabilization cell models

M. Leguèbe [a,b,c], C. Poignard [a,b,c], L. Weynans [b,c,a,*]

[a] INRIA, F-33400 Talence, France
[b] Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France
[c] CNRS, IMB, UMR 5251, F-33400 Talence, France

ARTICLE INFO

Article history:
Received 1 October 2013
Received in revised form 30 January 2015
Accepted 15 March 2015
Available online 20 March 2015

Keywords:
Finite differences on Cartesian grids
Interface transmission conditions
Truncation error
Cell modeling

ABSTRACT

In this paper, we present a new finite differences method to simulate electropermeabilization models, like the model of Neu and Krassowska or the recent model of Kavian et al. These models are based on the evolution of the electric potential in a cell embedded in a conducting medium. The main feature lies in the transmission of the voltage potential across the cell membrane: the jump of the potential is proportional to the normal flux thanks to the well-known Kirchoff law. An adapted scheme is thus necessary to accurately simulate the voltage potential in the whole cell, notably at the membrane separating the cell from the outer medium. We present a second-order finite differences scheme in the spirit of the method introduced by Cisternino and Weynans for elliptic problems with immersed interfaces. This is a Cartesian grid method based on the accurate discretization of the fluxes at the interface, through the use of additional interface unknowns. The main novelty of our present work lies in the fact that the jump of the potential is proportional to the flux, and therefore is not explicitly known. The original use of interface unknowns makes it possible to discretize the transmission conditions with enough accuracy to obtain a second-order spatial convergence. We prove the second-order spatial convergence in the stationary linear one-dimensional case, and the first-order temporal convergence for the dynamical non-linear model in one dimension. We then perform numerical experiments in two dimensions that corroborate these results.

1. Introduction

In this paper, we aim at providing an efficient numerical method to compute the voltage potential in a biological cell submitted to high electric pulses. We propose a second-order Cartesian method to simulate the electric cell model of Kavian et al. but our method is applicable for other problems involving specific transmission conditions at the interface.

1.1. Motivations

Electropermeabilization (also called electroporation) is a significant increase in the electrical conductivity and permeability of the cell membrane that occurs when pulses of large amplitude (a few hundred volts per centimeter) are applied to

* Corresponding author. Tel.: +33 (0)5 40 00 21 23; fax: +33 (0)5 40 00 35 36.
E-mail address: Lisl.Weynans@math.u-bordeaux1.fr (L. Weynans).

http://dx.doi.org/10.1016/j.jcp.2015.03.028
0021-9991/

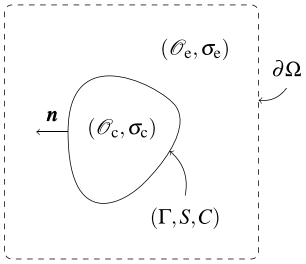


Fig. 1. Geometry of the problem. The cell $\mathscr{O}_c$ is imbedded in the bath $\mathscr{O}_e$. The whole domain Ω is defined by $\Omega = \mathscr{O}_e \cup \overline{\mathscr{O}_c}$.

the cells: due to the electric field, the cell membrane is permeabilized, and then non-permeant molecules can easily enter the cell cytoplasm by diffusion through the electropermeabilized membrane areas. If the pulses are too long, too numerous, or if their amplitude is too high, the cell membrane is irreversibly destroyed and the cells are killed. However, should the pulse duration be sufficiently short (a few milliseconds or a few microseconds, depending on the pulse amplitude) then the cell membrane reseals within several tens of minutes: this is termed reversible electroporation, which preserves the cell viability and is used in electrochemotherapy to vectorize the drugs until the cell inside. Many clinical studies (phase II and phase III) have proven that electrochemotherapy of cutaneous or subcutaneous metastases or tumors, either with bleomycin or with cisplatin displays an objective response rate of more than 80%. Reduction of tumor size has been achieved with electrochemotherapy faster and more efficiently than in standard chemotherapy for both cutaneous and subcutaneous tumors. Therefore the efficacy of such a treatment has been proven for superficial metastases. However, despite that cell electropermeabilization by micropulses is well-known from the experimental point of view, there is a lack of numerical results that are coherent with the experiments, preventing a systematic use of electropermeabilization in configurations far from the experiments, especially for deep tumors. This lack is mainly due to the complexity of the electropermeabilization models at the cell scale, which involve partial differential equations with non-standard transmission conditions through the cell membrane.

Actually, the probably most achieved model has been derived in the 2000s by Neu, Krassowska et al. from physical considerations ([19] and [5]). However, the model involved many parameters, that prevents fitting to the experiments. Moreover the theoretical analysis of this model, which can help in the derivation of accurate numerical schemes, seems to be hardly achievable due to the unboundedness of several unknowns. Roughly speaking, the current models provide a qualitative explanation of the electropermeabilization, but the problem of the quantitative description remains open. Very recently, Kavian et al. have provided a new phenomenological model involving few parameters, and they show the solvability of their model [10].

The aim of this paper is to present an accurate numerical method to compute the voltage potential in a biological cell when electropermeabilizing electric field is applied. The numerical method is a finite differences method on Cartesian grids inspired from the paper of Cisternino and Weynans [3] but adapted to the features of the electropermeabilization models.

1.2. Electropermeabilization model

We first recall the electropermeabilization model of Kavian et al. Due to its very thin thickness and its very low conductivity the cell membrane can be modeled as a surface electric material Γ with a capacity C and a surface conductivity S. The electroporation model of Kavian et al. describes the surface conductivity through a sliding-door type model [10] (see Fig. 1).

Denote by $\mathscr{O}_c$ the cell cytoplasm and by $\mathscr{O}_e$ the extracellular medium. Let σ be the conductivity chart of the medium, that is

$$\sigma = \begin{cases} \sigma_e, & \text{in the exterior domain } \mathscr{O}_e, \\ \sigma_c, & \text{inside the cell } \mathscr{O}_c. \end{cases}$$

The voltage potential satisfies the following P.D.E. model (1):

$$U(0,\cdot) = 0 \quad \text{in } \mathscr{O}_e \cup \mathscr{O}_c, \tag{1a}$$

and for $t > 0$

$$\Delta U = 0, \quad \text{in } (0,T) \times (\mathscr{O}_e \cup \mathscr{O}_c), \tag{1b}$$

$$U(t,\cdot) = g(t,\cdot) \quad \text{on } (0,+\infty) \times \partial\Omega, \tag{1c}$$

with the jump conditions

$$[\sigma \partial_{\mathbf{n}} U] = 0, \quad \text{on } (0,T) \times \Gamma, \tag{1d}$$

$$C\partial_t [U](t,\cdot) + S(t,[U])[U] = \sigma_c \partial_{\mathbf{n}} U(t,\cdot)|_{\Gamma^-}, \quad \text{on } (0,T) \times \Gamma. \tag{1e}$$

The surface membrane conductivity S is defined as an interpolation between the two values S_{ir} and S_{L}, which are respectively the surface conductivity of an irreversibly electropermeabilized region and the lipid surface conductivity:

$$S(t,\lambda) := S_{\text{L}} + (S_{\text{ir}} - S_{\text{L}})X(t,\lambda), \tag{1f}$$

where the function X satisfies the differential equation for $t > 0$ (we set $\lambda(t) := \lambda(t,s) := [U(t,s)]$ for *a.e.* $s \in \Gamma$):

$$\begin{cases} \dfrac{\partial X(t,\lambda)}{\partial t} = \max\left(\dfrac{\beta(\lambda(t)) - X(t,\cdot)}{\tau_{\text{ep}}} ; \dfrac{\beta(\lambda(t)) - X(t,\cdot)}{\tau_{\text{res}}} \right). \\ X(0,\lambda) = 0, \end{cases} \tag{2}$$

where β is an even regularized Heaviside function. For instance β can be chosen as

$$\text{for all } \lambda \in \mathbb{R}, \qquad \beta(\lambda) := (1 + \tanh(k_{\text{ep}}(|\lambda| - V_{\text{rev}}))/2.$$

Despite jump condition (1d) is nothing but the flux continuity, the transmission condition on the jump of the potential (1d) reflects the influence of the thin resistive membrane. Actually, the jump of the potential is linked to the electric flux through a Kadam–Katchalsky type condition and therefore the solution U is not continuous across the interface Γ. Note that in condition (1e), it is necessary to implicit the flux $\partial_{\mathbf{n}} u|_{\Gamma^-}$ in the time discretization for stability reasons, otherwise even for the linear model (which consists in $S(t,\lambda) \equiv S_{\text{L}}$) a drastic Courant–Friedrichs–Lax condition appears, as previously observed by Guyomarc'h et al. [8]. More precisely, in their paper they used a forward Euler time scheme (Eq. (26) p. 1005 of [8]), for which a very small time step has to be chosen. In [8], a discontinuous Galerkin method is proposed for the space discretization.

1.3. State of the art and originality of the numerical method

We present here a second-order finite differences method on a Cartesian grid, for which the flux of (1e) is implicit. The discretization is inspired by the second-order accurate method developed in [3] for elliptic problems with interfaces. The originality of the latter method relies on the use of additional unknowns located at the intersections between the grid and the interface, which makes possible to discretize straightforwardly the flux transmission conditions with enough accuracy to obtain a second-order spatial convergence.

This method is in the same spirit as the well-known Immersed Interface Method (IIM) of LeVeque and Li [12]. The IIM relies on a discretization of the elliptic operator near the interface with formulas accounting for the jumps across the interface. In order to find these formulas, a linear system with six unknowns has to be solved for each of the concerned grid points. The method is second-order accurate in L^∞-norm. Numerous developments of the IIM have been performed, among them: the fast IIM algorithm of Li [13] for elliptic problems with piecewise constant coefficients, the Explicit Jump Immersed Interface Method (EJIIM) by Wiegmann and Bube [24], the Decomposed Immersed Interface Method (DIIM) by Bethelsen [2], and the MIIM (maximum principle preserving) by Li and Ito [14].

The first Cartesian grid method for elliptic problems was designed by Mayo in 1984 [16], and developed further in [17] and [18]. In these previous papers, an integral equation was derived to solve elliptic interface problems with piecewise constant coefficients with a second-order accuracy in maximum norm. Another class of Cartesian method, recently introduced by Zhou et al., is the Matched Interface and Boundary (MIB) method [25]. This method can provide finite difference schemes of arbitrary high order. The solution on each side of the interface is extended on fictitious points on the other side. Coco and Russo [4] and Latige et al. [11] recently also proposed efficient schemes for this problem. Their approach is based, for the first ones, on the relaxation of inner equations and transmission conditions by a fictitious time method, and for the second ones on a piecewise polynomial representation of the solution on a dual grid. All the methods cited above are second-order accurate. Other classes of Cartesian methods also exist, less accurate in the case of interface problems, but probably simpler to implement and producing symmetric matrices: Fedkiw et al. [15] and Gibou et al. [7] developed methods inspired by Fedkiw's Ghost-Fluid Method [6] for multiphase flows. These methods are second-order accurate for Dirichlet boundary conditions on arbitrary domains, but only first order accurate for interface problems. Note that all the above methods deal with explicitly given jumps of the Dirichlet and Neumann traces.

Contrary to these methods, the method introduced in [3] is based on the use of additional interface unknowns rather than a non-standard discretization of the Laplacian directly accounting for the discontinuities near the interface. The interface unknowns are particularly useful in the case of the electropermeabilization model, since the value of the transmembrane potential is accurately needed. Moreover, the study of the continuous electropermeabilization model includes a re-formulation of the problem defined in the whole space as a problem on the interface using the non-local Steklov–Poincaré operators. It is thus quite natural to re-formulate the numerical method on interface points with discrete Steklov–Poincaré operators, and the use of interface unknowns simplifies this re-formulation. Directly placing points on the interface is also useful to this model, as some data are only defined on the membrane, such as the surface membrane conductivity S. This property of the method makes it possible for example to compute integrals on the interface with much efficacy, since no interpolation between grid values is needed.

Concerning the discretization requirements needed to get a second-order spatial convergence, it has been noted since the introduction of Cartesian grid methods that an $O(h)$ truncation error at the points near the interface is enough to get

an $O(h^2)$ convergence in maximum norm if the discretization is second-order on the regular grid points. However, in the literature, only few works have been devoted to this study. For one-dimensional methods, Huang and Li performed in [9] a convergence analysis of the IIM, using non-negative comparison functions, and in [24] Wiegmann and Bube presented a proof of convergence for one-dimensional problems with piecewise constant coefficients, using a detailed analysis and identification of the coefficients of the matrices involved. For two-dimensional methods, Beale and Layton proved in [1] the second-order convergence for piecewise constant diffusion coefficients, using the fact that a grid function located near the interface can be written as the divergence of a function smaller in norm, and Li and Ito proved in [14] the second-order convergence of their MIIM, using the maximum principle, but with a technical condition related to the location of the interface with respect to the grid point that is not always satisfied. Here again, we emphasize that these methods deal with explicitly given jumps of the Dirichlet and Neumann traces.

In this paper we present a new approach to obtain the second-order convergence in maximum norm. Note that the electropermeabilization problem is sensibly different from the elliptic problem studied in the references above, due to the jump of the Dirichlet trace, which is given implicitly. We thus propose a second-order method adapted from [3]. The proof of the order of accuracy is based on the monotonicity of the discretization matrix. Note that this monotonicity is not trivial since the discretization matrix is not diagonally dominant. Once it is proven that this matrix is monotone, then it is possible to obtain accurate estimations of the coefficients of its inverse, which, combined to the truncation error coefficients, lead to the second-order convergence.

1.4. Outline of the paper

The paper is organized as follows. Section 2 describes the finite differences method used to solve numerically the electropermeabilization model. In Section 3, we first prove that, in one dimension, this method is convergent with second order accuracy for the static linear model in space. We then prove, in Section 4, that in one dimension the method converges with first-order accuracy in time in the dynamical non-linear case. For the sake of clarity, we reduce the theoretical assessment to one dimension. In Section 5, we present two-dimensional numerical validations that corroborate the results of the previous sections. In particular we show that for the linear problem, the second-order accuracy is reached, while the method for the dynamical non-linear case converges with first-order accuracy in time. We conclude the paper by perspectives on the applications of our numerical scheme.

2. Statement of the method

2.1. Description of the interface and classification of grid points

We choose to perform the discretization on a Cartesian grid covering $\mathscr{O}_e \cup \mathscr{O}_c$. The points on this grid are defined by $M_{ij} = (x_i, y_j) = (ih, jh)$, with h the grid spacing. We denote by u_{ij} the approximation of U at the point (x_i, y_j). If necessary, the additional subscript u_{ij}^n will be used to indicate the time $t^n = n\,dt$, where dt is the time step. Similarly, if it is necessary to distinguish in which subdomain the grid point is, we will use the notations u_{ij}^c and u_{ij}^e.

In order to describe accurately the geometric configuration in the vicinity of the interface we use the level set method introduced by Osher and Sethian [21]. We refer the interested reader to [22,23] and [20] for recent reviews of this method. The zero isoline of the level set function, defined by the signed function φ, given by:

$$\varphi(x) = \begin{cases} dist_\Sigma(x), & \text{outside of the interface,} \\ -dist_\Sigma(x), & \text{inside of the interface,} \end{cases} \tag{3}$$

represents implicitly the interface Σ immersed in the computational domain. A useful property of the level set function is

$$\mathbf{n}(x) = \frac{\nabla\varphi(x)}{|\nabla\varphi(x)|}, \tag{4}$$

where $\mathbf{n}(x)$ is the outward normal vector of the isoline of φ passing on x. This allows us to compute the values of the normal to the interface. In the present study the level set will always be stationary, since the electropermeabilization model does not take into account a possible volume variation of the cell.

We say that a grid point is neighboring the interface if the sign of φ changes between this point and at least one of its neighbors, see Fig. 2. On the contrary, grid points far from the interface are said regular grid points. If the intersection of the interface and the segment $[M_{ij}M_{i+1j}]$ exists, then we define the interface point $I_{i,j,E} = (\tilde{x}_{i,j,E}, y_j)$ as this intersection. In the same way, if the intersection of the interface and $[M_{i-1j}M_{ij}]$ exists, then we define the interface point $I_{i,j,W} = (\tilde{x}_{i,j,W}, y_j)$ as this intersection.

Similarly, the interface points $I_{i,j,N} = (x_i, \tilde{y}_{i,j,N})$ and $I_{i,j,S} = (x_i, \tilde{y}_{i,j,S})$ are respectively defined as the intersections of the interface and the segments $[M_{ij}M_{ij+1}]$ and $[M_{ij-1}M_{ij}]$. Let us remark that, with this notation, the same interface point can be described in two different ways, see Fig. 2:

$$I_{i,j,S} = I_{i,j-1,N} \text{ or } I_{i,j,E} = I_{i+1,j,W}.$$

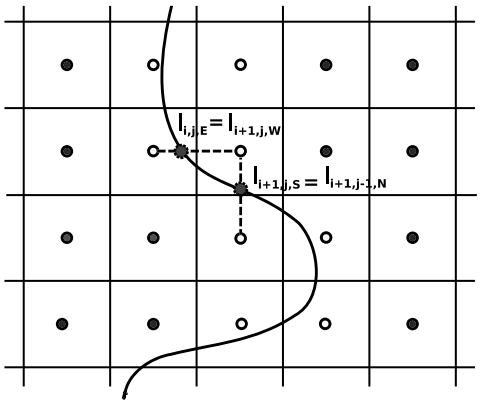


Fig. 2. Example of geometrical configuration, with regular grid points in black, points neighboring the interface in white, and interface points with the two possible notations.

At each interface point we create two additional unknowns, called interface unknowns, and denoted by $\tilde{u}_{i,j,\gamma}$ with $\gamma =$ E, W, N or S. The interface unknowns carry the values of the numerical solution on each side of the interface.

2.2. Global discretization of the system

We choose a first-order semi-implicit temporal discretization of the equations on the electric potential U, and an explicit first-order Euler discretization for X. The computational domain is a square of characteristic size 1, with N points in both x- and y-directions.

We denote by $(\Delta u)_{ij}$ the finite differences approximation of the Laplacian at point $M_{i,j}$. The approximated normal derivative at the interface point $I_{i,j,\gamma}$ in the exterior (*resp.* interior domain), with $\gamma =$ E, W, N or S, is denoted by $(\partial_{\mathbf{n}} u_{\mathrm{e}})_{i,j,\gamma}$ (*resp.* $(\partial_{\mathbf{n}} u^{\mathrm{c}})_{i,j,\gamma}$). We write the discretized system on the Cartesian grid:

- The Laplace operator on the grid points:

$$\forall\,(i,j) \in \{1,..,N\}^2\,,\ \forall\, n \geqslant 0, \quad (\Delta u)_{ij}^{n+1} = 0. \tag{5}$$

- The jump conditions (1d) :

$$\forall\,(i,j) \in \{1,..,N\}^2\,,\ \forall\, n \geqslant 0, \ \text{ if } I_{i,j,\gamma} \text{ exists, then } [\sigma\,(\partial_{\mathbf{n}} u)_{i,j,\gamma}^{n+1}] = 0. \tag{6}$$

- The transmembrane potential evolution (1e):

$$\forall\,(i,j) \in \{1,..,N\}^2\,,\ \forall\, n \geqslant 0, \ \text{ if } I_{i,j,\gamma} \text{ exists, then}$$
$$C\frac{[u]_{i,j,\gamma}^{n+1} - [u]_{i,j,\gamma}^{n}}{dt} + S_{i,j,\gamma}^{n}[u]_{i,j,\gamma}^{n} = \sigma_{\mathrm{c}}\,(\partial_{\mathbf{n}} u_{\mathrm{c}})_{i,j,\gamma}^{n+1}. \tag{7}$$

  where $S_{i,j,\gamma}^{n}$, with $\gamma =$ E, W, N or S, is given by

$$S_{i,j,\gamma}^{n} = S_{\mathrm{L}} + S_{\mathrm{ir}}\, X_{i,j,\gamma}^{n}.$$

- The coefficient X:

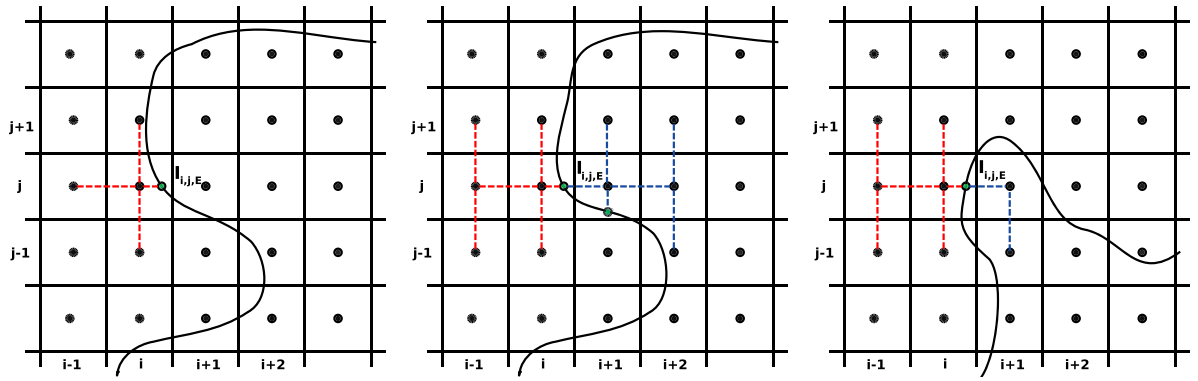
$$\forall\,(i,j) \in \{1,..,N\}^2\,,\ \forall\, n \geqslant 0, \ \text{ if } I_{i,j,\gamma} \text{ exists, then}$$
$$X_{i,j,\gamma}^{n+1} = X_{i,j,\gamma}^{n} + dt \max\left(\frac{\beta([u]_{i,j,\gamma}^{n}) - X_{i,j,\gamma}^{n}}{\tau_{\mathrm{ep}}}\,;\,\frac{\beta([u]_{i,j,\gamma}^{n}) - X_{i,j,\gamma}^{n}}{\tau_{\mathrm{res}}}\right). \tag{8}$$

Initial conditions are $u^0 = 0$ and $X^0 = 0$. We impose Dirichlet boundary conditions on the outer boundary of the computational domain. What we call in the following "linear problem" corresponds to the case where S is kept constant in time, in which case no electropermeabilization happens.

It is also possible to discretize the transmembrane potential evolution as

$$C\frac{[u]_{i,j,\gamma}^{n+1} - [u]_{i,j,\gamma}^{n}}{dt} + S_{i,j,\gamma}^{n}[u]_{i,j,\gamma}^{n+1} = \sigma_{\mathrm{c}}\,(\partial_{\mathbf{n}} u_{\mathrm{c}})_{i,j,\gamma}^{n+1}.$$

This option can be less convenient because it requires to solve at each time step a linear system with a different matrix, while the discretization that we have chosen uses the same matrix at each iteration. But on the other side, because it is



(a) Laplace operator (b) Second order flux discretization (c) First order flux discretization

Fig. 3. Examples of discretization: (a): Laplacian, (b): second-order flux, (c): first-order flux.

more implicit, it may require weaker stability conditions on the time step. The analysis of this alternative discretization can be carried on in a similar way to the discretization that we have chosen.

In the following subsection we explain how we discretize the terms $(\Delta u)^n_{ij}$, $(\partial_{\mathbf{n}} u_e)^n_{i,j,\gamma}$ and $(\partial_{\mathbf{n}} u_c)^n_{i,j,\gamma}$ with $\gamma = \mathrm{E}, \mathrm{W}, \mathrm{N}$ or S. Once these discretizations are achieved we obtain a linear system that is solved with a standard linear solver such as restarted GMRES.

2.3. Discrete elliptic operator

On the regular grid points, we use the standard centered finite differences scheme to approximate the Laplacian:

$$(\Delta u)_{ij} = \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{h^2} + \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{h^2}. \tag{9}$$

This formula is second-order accurate:

$$(\Delta u)_{ij} = \Delta U(x_i, y_j) + O(h^2).$$

For a grid point M_{ij} neighboring the interface, we compute $(\Delta u)_{ij}$ with the values on M_{ij} and the closest points (grid or intersection points) to M_{ij} in each direction. For instance for the situation illustrated on Fig. 3(a) we get:

$$(\Delta u)_{ij} = \frac{\dfrac{\tilde{u}_{i,j,\mathrm{E}} - u_{ij}}{\tilde{x}_{i,j,\mathrm{E}} - x_i} - \dfrac{u_{ij} - u_{i-1j}}{h}}{\dfrac{(\tilde{x}_{i,j,\mathrm{E}} - x_i) + h}{2}} + \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{h^2}. \tag{10}$$

This discretization is first-order accurate:

$$(\Delta u)_{ij} = \Delta U(x_i, y_j) + O(h).$$

2.4. Discretization of fluxes at the interface

Figs. 3(b) and 3(c) provide examples of interface configurations. It is straightforward to compute a second-order approximation of the x-derivative with three *a priori* non-equidistant points. For example, we approximate the x-derivative on the left side of the interface with the points M_{i-1j}, M_{ij} and $I_{i,j,\mathrm{E}}$ by:

$$(\partial_x u)_{i,j,\mathrm{E}} = \frac{(x_i - \tilde{x}_{i,j,\mathrm{E}})}{h\,(x_{i-1} - \tilde{x}_{i,j,\mathrm{E}})}(u_{i-1j} - \tilde{u}_{i,j,\mathrm{E}}) - \frac{(x_{i-1} - \tilde{x}_{i,j,\mathrm{E}})}{h\,(x_i - \tilde{x}_{i,j,\mathrm{E}})}(u_{ij} - \tilde{u}_{i,j,\mathrm{E}}). \tag{11}$$

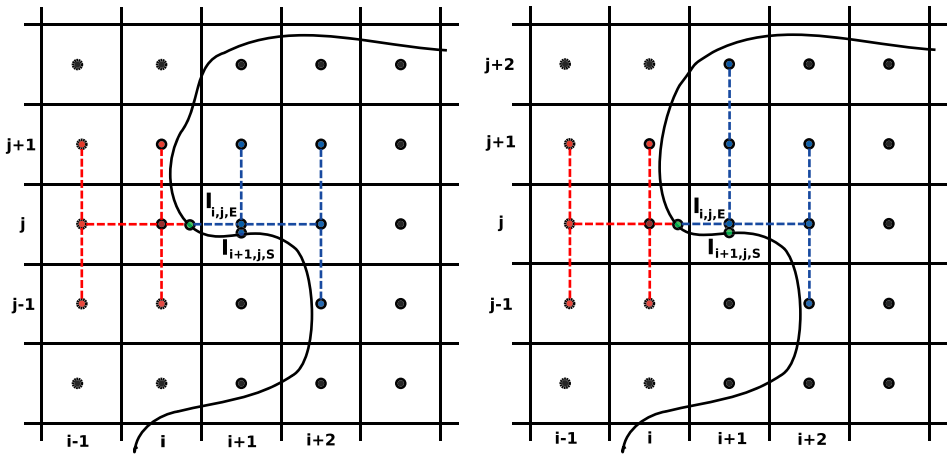
We thus have

$$(\partial_x u)_{i,j,\mathrm{E}} = \frac{\partial U}{\partial x}(\tilde{x}_{i,j,\mathrm{E}}, y_j) + O(h^2).$$

The right x-derivative $\frac{\partial U}{\partial x}(\tilde{x}_{i+1,j,\mathrm{W}}, y_j)$ is approximated by $(\partial_x u)_{i+1,j,\mathrm{W}}$ which is computed similarly.

Along the y-direction there is no grid unknown that can be used so we use a linear combination of $(\partial_y u)_{ij}$ and $(\partial_y u)_{i-1j}$, defined respectively as second-order approximations of the y-derivative on M_{ij} and M_{i-1j}. We obtain

$$(\partial_y u)_{i,j,\mathrm{E}} = \frac{\tilde{x}_{i,j,\mathrm{E}} - x_{i-1}}{h}\,(\partial_y u)_{ij} - \frac{\tilde{x}_{i,j,\mathrm{E}} - x_i}{h}\,(\partial_y u)_{i-1j}. \tag{12}$$



(a) Centered stencil (b) Decentered stencil

Fig. 4. Centered and decentered stencils. In (a) the discretization of the flux on the point $I_{i,j,\mathrm{E}}$ involves the intersection point $I_{i+1,j,\mathrm{S}}$ and is ill-conditioned. (b) shows the decentered stencil: the discretization of the flux on the point $I_{i,j,\mathrm{E}}$ involves grid point $(i+1, j+2)$ instead of the intersection point $I_{i+1,j,\mathrm{S}}$.

The resulting formula is second-order accurate:

$$(\partial_y u)_{i,j,\mathrm{E}} = \frac{\partial U}{\partial y}(\tilde{x}_{i,j,\mathrm{E}}, y_j) + O(h^2).$$

The formulas for $(\partial_y u)_{ij}$ and $(\partial_y u)_{i-1j}$ depend on the local configuration on the interface, but are based on the same principle as for (11). The formulas (11) and (12) are consistent if the point $M_{i-1,j}$ belongs to the same domain as M_{ij}. Consequently they require that there are at least two adjacent points in each direction belonging to the same domain. It means that a resolution of at least two points in each direction is necessary for this discretization. It is the case most of the time if the interface is sufficiently resolved.

Nevertheless, in some cases, as described on Fig. 3(c), only one grid point is available in one direction. In this case, we use a first-order discretization involving only three points of this side of the interface: see Fig. 3(c). Actually, it is always possible to perform at least a first order discretization for the fluxes: if two interfaces are arbitrary close, there are two possibilities. Either they lie between the same two grid points, which means that the interface is under-resolved, and the algorithm does not know that there is a small space between the two interfaces, but numerically there is no problem, or the two interfaces are separated by at least one grid point. In this case, using this grid point and the interface points associated, it is always possible to perform a first order discretization in the same spirit as on Fig. 3(c).

The normal derivatives at the interface are discretized by the scalar product of x- and y-derivatives with the normal to the interface:

$$(\partial_{\mathbf{n}} u)_{i,j,\gamma} = (\partial_x u)_{i,j,\gamma}\, n_x + (\partial_y u)_{i,j,\gamma}\, n_y, \text{ with } \gamma = \mathrm{E}, \mathrm{W}, \mathrm{N}, \mathrm{S}, \tag{13}$$

with (n_x, n_y) an approximation of the vector normal to the interface at point $I_{i,j,\gamma}$.

Numerical instabilities may appear when the discretization of the flux at an interface point involves another intersection point closely located to a grid point. This situation is illustrated in Fig. 4(a). In these cases we use a decentered discretization: instead of using the second intersection point we use the closest grid point located in the opposite direction. For instance, on Fig. 4(b), the term $(\partial_y u_c)_{i,j,\mathrm{E}}$ is computed with the grid points (x_{i+1}, y_j), (x_{i+1}, y_{j+1}) and (x_{i+1}, y_{j+2}) instead of (x_{i+1}, y_j), (x_{i+1}, y_{j+1}) and $I_{i+1,j,\mathrm{S}}$.

This alternative discretization may be not possible in some cases, because the closest grid point located in the opposite direction lies on the other side of the interface: for instance, if one extremity of a subdomain is only two grid points thick. Two options are then available: either use the usual stencil, which is possible if there are two points in each direction, or simply use a first-order scheme. We detail in Algorithm 1 how to choose the stencil for the flux discretization.

2.5. Nature of resulting linear system

Obviously the matrix resulting from the discretization detailed above is non-symmetric, due to the use of interface values. We have plotted on Fig. 5 two examples of matrix structures, one where the interface unknowns are numbered separately from grid unknowns, and one where the interface unknowns are numbered between grid unknowns that neighbor them.

On Fig. 6 are plotted the values of the condition number of the discretization matrix in two dimensions, as a function of the number of grid points and as a function of the conductivity parameter σ_e. The global tendency is that the condition number decreases when the number of discretization points increases, or when the conductivity increases. The higher

Algorithm 1 Algorithm for choosing the left flux stencil at interface point $I_{i,j,\mathrm{E}}$.

Without loss of generality, we assume that the interface point, denoted in the following $I_{i,j,\mathrm{E}}$, is on the x-axis, and we only describe the stencil construction for the left flux.

1. Check if the grid points M_{ij} and M_{i-1j} belong to the same subdomain. If yes, then go to 2, otherwise go to 3;
2. Construction of a second-order stencil:
   - Computation of a second-order x-derivative $(\partial_x u)_{i,j,\mathrm{E}}$ on $I_{i,j,\mathrm{E}}$ with points $I_{i,j,\mathrm{E}}$, M_{ij} and M_{i-1j}
   - Computation of a second-order y-derivative $(\partial_y u)_{ij}$ on grid point M_{ij}:
     - if M_{ij+1} belongs to the same subdomain as M_{ij}, then choose M_{ij+1}, otherwise if M_{ij-2} belongs to the same subdomain as M_{ij} then choose M_{ij-2}, otherwise choose the interface point between M_{ij} and M_{ij+1}.
     - if M_{ij-1} belongs to the same subdomain as M_{ij}, then choose M_{ij-1}, otherwise if M_{ij+2} belongs to the same subdomain as M_{ij} then choose M_{ij+2}, otherwise choose the interface point between M_{ij} and M_{ij-1}.
     - With the two chosen points and M_{ij}, compute a second-order y-derivative $(\partial_y u)_{ij}$ on grid point M_{ij}.
   - Computation of a second-order y-derivative $(\partial_y u)_{i-1j}$ on grid point M_{i-1j}, similarly as for $(\partial_y u)_{ij}$
   - Computation of a second order y-derivative $(\partial_y u)_{i,j,\mathrm{E}}$ on $I_{i,j,\mathrm{E}}$ with formula (12)
3. Construction of a first-order stencil:
   - Computation of a first-order x-derivative on $I_{i,j,\mathrm{E}}$ with points M_{ij} and $I_{i,j,\mathrm{E}}$,
   - Computation of a first-order y-derivative on point M_{ij} with point M_{ij} and M_{ij-1} or M_{ij+1} depending on the orientation of the normal.
4. Computation of the normal derivative on $I_{i,j,\mathrm{E}}$ with formula (13)

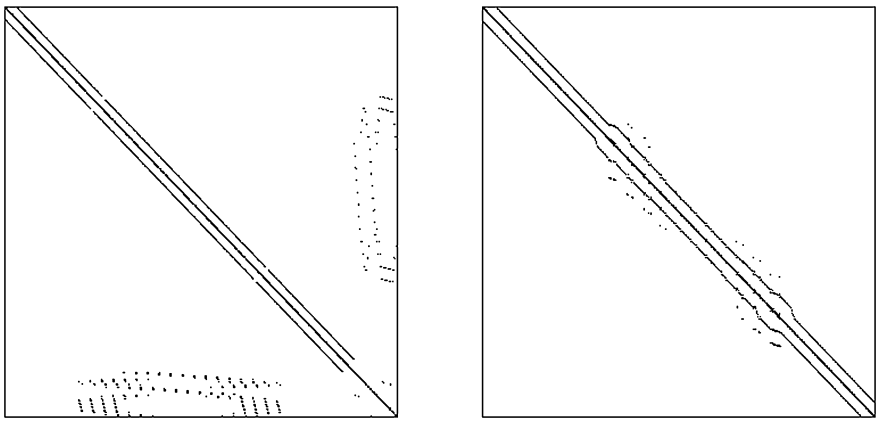


Fig. 5. Examples of matrix structures: only non-zero terms are plotted, in black, (a): interface unknowns numbered separately from grid unknowns, (b): interface unknowns are numbered between grid unknowns.

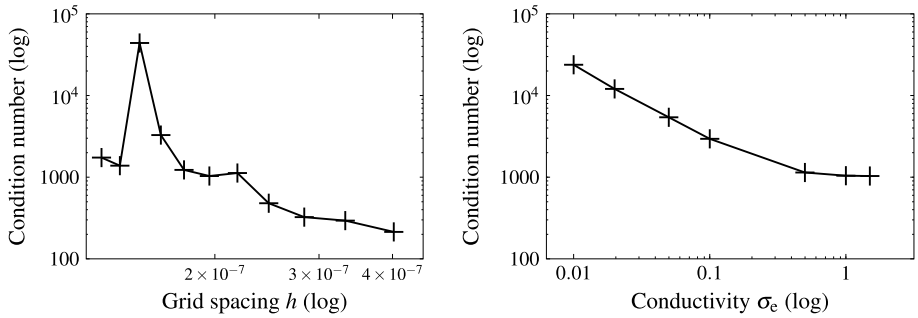


Fig. 6. Numerical study of the condition number in 2D, (a): depending of the number of points each direction, for $\sigma_e = 1.5$ S/m, (b): depending on the conductivity parameter σ_e, for 100×100 grid points.

values of the condition number are probably due to small distances between grid points and interface points, which can occur independently of the number of discretization points.

3. Second order convergence in linear static unidimensional case

In this section, we study the convergence of the method for the linear static model. In Subsections 3.1 and 3.2 we respectively give the notations that will be used hereafter and detail the linear system arising from the discretization of the method. Then in Subsection 3.3 we prove the monotonicity of the matrix of the linear system. Here the discretization matrix has not the usual features that help in characterizing monotone matrices, such as the diagonally dominant property.

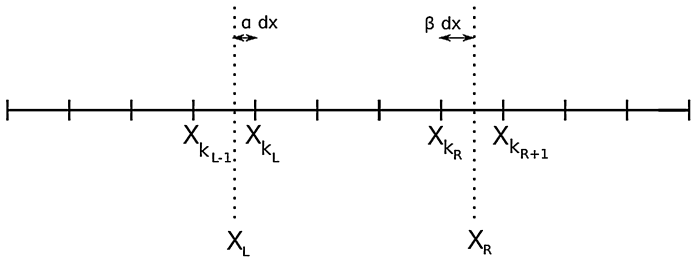


Fig. 7. Geometrical configuration in one dimension.

We directly prove that all the coefficients of the inverse matrix are positive. At the end, in Subsection 3.4 we use the monotonicity to estimate accurately the value of the coefficients of the inverse matrix, depending on their location in the matrix, and we prove the second-order convergence of the method in maximum norm.

3.1. Notations

Let us give some additional notations about the spatial configuration that we consider, cf. Fig. 7: the interior domain and the exterior domain are

$$\mathscr{O}_c = (x_L, x_R) \text{ and } \mathscr{O}_e = (0, x_L) \cup (x_R, 1).$$

The outer boundary and the interface are composed of two points:

$$\partial\Omega = \{0\} \cup \{1\} \text{ and } \Gamma = \{x_L\} \cup \{x_R\}.$$

The interface point x_L is located between the grid points x_{k_L-1} and x_{k_L} and let

$$x_L = x_{k_L} - \alpha h,$$

with $0 \le \alpha \le 1$. Similarly, the interface point x_R is located between the grid points x_{k_R} and x_{k_R+1} and we set

$$x_R = x_{k_R} + \beta h,$$

with $0 \le \beta \le 1$.

There are 2 interface unknowns on each interface point, one interior and one exterior: u_L^e, u_L^c, u_R^c and u_R^e. The computational domain is discretized on a uniform grid, where the grid points x_i are defined by

$$x_i = i\,h, \quad \text{with} \quad h = 1/(N+1).$$

3.2. Linear system

The fluxes at the interface are computed with the formulas:

$$(\partial_{\mathbf{n}} u^c)_L = -\frac{1+2\alpha}{\alpha(\alpha+1)h}(u_{k_L} - u_L^c) + \frac{\alpha}{(1+\alpha)h}(u_{k_L+1} - u_{k_L}), \tag{14}$$

$$(\partial_{\mathbf{n}} u^c)_R = \frac{1+2\beta}{\beta(\beta+1)h}(u_R^c - u_{k_R}) - \frac{\beta}{(\beta+1)h}(u_{k_R} - u_{k_R-1}), \tag{15}$$

and

$$(\partial_{\mathbf{n}} u^e)_L = -\frac{3-2\alpha}{(1-\alpha)(2-\alpha)h}(u_L^e - u_{k_L-1}) + \frac{1-\alpha}{(2-\alpha)h}(u_{k_L-1} - u_{k_L-2}), \tag{16}$$

$$(\partial_{\mathbf{n}} u^e)_R = \frac{3-2\beta}{(1-\beta)(2-\beta)h}(u_{k_R+1} - u_R^e) - \frac{1-\beta}{(2-\beta)h}(u_{k_R+2} - u_{k_R+1}). \tag{17}$$

All these flux formulas are on the same principle: if you have three points x_1, x_2 and x_3, then the coefficients allocated to these points in order to approximate with second-order accuracy the derivative in x_1 are respectively c_1, c_2 and c_3 such that:

$$c_1 + c_2 + c_3 = 0, \tag{18a}$$

$$x_1 c_1 + x_2 c_2 + x_3 c_3 = 1 \text{ or } -1, \tag{18b}$$

$$x_1^2 c_1 + x_2^2 c_2 + x_3^2 c_3 = 0. \tag{18c}$$

The sign in the above relationships depends on the orientation of the outer normal to the interface.

The jump conditions can be expressed as

$$\begin{aligned}
\sigma_c(\partial_{\mathbf{n}}u^c)_L - \sigma_e(\partial_{\mathbf{n}}u^e)_L &= 0,\\
\sigma_e(\partial_{\mathbf{n}}u^e)_R - \sigma_c(\partial_{\mathbf{n}}u^c)_R &= 0,\\
S(u_L^e - u_L^c) - \sigma_c(\partial_{\mathbf{n}}u^c)_L &= 0,\\
\sigma_c(\partial_{\mathbf{n}}u^c)_R - S(u_R^e - u_R^c) &= 0,
\end{aligned}$$

but we rather rewrite them as

$$S(u_L^e - u_L^c) - \sigma_e(\partial_{\mathbf{n}}u^e)_L = 0, \tag{19a}$$

$$S(u_R^e - u_R^c) - \sigma_e(\partial_{\mathbf{n}}u^e)_R = 0, \tag{19b}$$

$$\sigma_c(\partial_{\mathbf{n}}u^c)_L - S(u_L^e - u_L^c) = 0, \tag{19c}$$

$$\sigma_c(\partial_{\mathbf{n}}u^c)_R - S(u_R^e - u_R^c) = 0, \tag{19d}$$

so that the matrix of the linear system is monotone.

The discretization of the Laplacian operator on all grid points reads

$$\begin{aligned}
\frac{-u_0 + 2u_1 - u_2}{h^2} &= 0,\\
\frac{-u_1 + 2u_2 - u_3}{h^2} &= 0,\\
&\vdots \quad \vdots\\
\frac{-u_{k_L-3} + 2u_{k_L-2} - u_{k_L-1}}{h^2} &= 0,\\
-2\frac{u_L^e - u_{k_L-1}}{(1-\alpha)(2-\alpha)h^2} + 2\frac{u_{k_L-1} - u_{k_L-2}}{(2-\alpha)h^2} &= 0,\\
-2\frac{u_{k_L+1} - u_{k_L}}{(1+\alpha)h^2} + 2\frac{u_{k_L} - u_L^c}{\alpha(1+\alpha)h^2} &= 0,\\
\frac{-u_{k_L} + 2u_{k_L+1} - u_{k_L+2}}{h^2} &= 0,\\
&\vdots \quad \vdots\\
\frac{-u_{k_R-2} + 2u_{k_R-1} - u_{k_R}}{h^2} &= 0,\\
-2\frac{u_R^c - u_{k_R}}{\beta(1+\beta)h^2} + 2\frac{u_{k_R} - u_{k_R-1}}{(1+\beta)h^2} &= 0,\\
-2\frac{u_{k_R+2} - u_{k_R+1}}{(2-\beta)h^2} + 2\frac{u_{k_R+1} - u_R^e}{(1-\beta)(2-\beta)h^2} &= 0,\\
\frac{-u_{k_R+1} + 2u_{k_R+2} - u_{k_R+3}}{h^2} &= 0,\\
&\vdots \quad \vdots\\
\frac{-u_{N-2} + 2u_{N-1} - u_N}{h^2} &= 0,\\
\frac{-u_{N-1} + 2u_N - u_{N+1}}{h^2} &= 0.
\end{aligned}$$

3.3. Preliminary results

In the following, we denote by A the discretization matrix of the linear system described in the previous subsection. We first prove that A is monotone by showing that all the coefficients of its inverse are positive. Let u be an array of size $N+4$ corresponding to N grid points and four interface unknowns such that $Au \geqslant 0$, and let us show that $u \geqslant 0$.

3.3.1. Sign of the normal interface derivative

We begin by proving that if the minimum of u is located on an interior interface point, then the normal derivative at this interface point is negative. Similarly, if the minimum of u is located on an exterior interface point then the normal derivative at this interface point is positive. We do it for example for the normal derivative on x_{L} in the exterior domain. The other cases would be treated exactly in the same way. The exterior normal derivative at x_{L} is

$$(\partial_{\mathbf{n}} u^{\mathrm{e}})_{\mathrm{L}} = -\frac{3-2\alpha}{(1-\alpha)(2-\alpha)h}(u_{\mathrm{L}}^{\mathrm{e}} - u_{k_{\mathrm{L}}-1}) + \frac{1-\alpha}{(2-\alpha)h}(u_{k_{\mathrm{L}}-1} - u_{k_{\mathrm{L}}-2}).$$

By hypothesis, $Au \geqslant 0$ hence

$$\frac{u_{k_{\mathrm{L}}-1} - u_{k_{\mathrm{L}}-2}}{h} \geqslant \frac{u_{\mathrm{L}}^{\mathrm{e}} - u_{k_{\mathrm{L}}-1}}{(1-\alpha)h},$$

therefore

$$\begin{aligned}(\partial_{\mathbf{n}} u^{\mathrm{e}})_{\mathrm{L}} &\geqslant -\frac{3-2\alpha}{(1-\alpha)(2-\alpha)h}(u_{\mathrm{L}}^{\mathrm{e}} - u_{k_{\mathrm{L}}-2}) + \frac{1-\alpha}{(2-\alpha)(1-\alpha)h}(u_{\mathrm{L}}^{\mathrm{e}} - u_{k_{\mathrm{L}}-1}),\\ &\geqslant \frac{-3+2\alpha+1-\alpha}{(1-\alpha)(2-\alpha)h}(u_{\mathrm{L}}^{\mathrm{e}} - u_{k_{\mathrm{L}}-1}),\\ &\geqslant -\frac{1}{(1-\alpha)h}(u_{\mathrm{L}}^{\mathrm{e}} - u_{k_{\mathrm{L}}-1}),\\ &\geqslant 0.\end{aligned}$$

3.3.2. Monotonicity of the matrix

The minimum of u can either be reached on a grid point in the interior or exterior subdomains, or on an interface point, interior or exterior.

- *if the minimum is reached on a grid point in the interior domain:*
  In this case we denote by $i_{\min}$ the index of the smallest component of u. We first assume that $i_{\min} \neq 1, N$. Using the Laplacian inequality on this point:

  $$\frac{-u_{i_{\min}+1} + 2u_{i_{\min}} - u_{i_{\min}-1}}{h^2} \geqslant 0,$$

  we deduce that $u_{i_{\min}-1}$ and $u_{i_{\min}+1}$ are equal to $u_{i_{\min}}$, and, using then all other Laplacian inequalities in the interior subdomain, we deduce that all values in the interior subdomain are equal to the minimum value $u_{i_{\min}}$. Therefore, the approximate normal derivative is zero. Thus, using discrete relationships (19a)–(19d), the jump of the numerical solution and the normal exterior derivative are all negative:

  $$\begin{aligned}0 = \sigma_{\mathrm{c}}(\partial_{\mathbf{n}} u^{\mathrm{c}})_{\mathrm{L}} &\geqslant S(u_{\mathrm{L}}^{\mathrm{e}} - u_{\mathrm{L}}^{\mathrm{c}}) \geqslant \sigma_{\mathrm{e}}(\partial_{\mathbf{n}} u^{\mathrm{e}})_{\mathrm{L}},\\ 0 = \sigma_{\mathrm{c}}(\partial_{\mathbf{n}} u^{\mathrm{c}})_{\mathrm{R}} &\geqslant S(u_{\mathrm{R}}^{\mathrm{e}} - u_{\mathrm{R}}^{\mathrm{c}}) \geqslant \sigma_{\mathrm{e}}(\partial_{\mathbf{n}} u^{\mathrm{e}})_{\mathrm{R}}.\end{aligned}$$

  We consider for instance the situation on the right interface point. From $S(u_{\mathrm{R}}^{\mathrm{e}} - u_{\mathrm{R}}^{\mathrm{c}}) \leqslant 0$ we deduce that $u_{\mathrm{R}}^{\mathrm{e}} = u_{i_{\min}}$. We can also write

  $$(\partial_{\mathbf{n}} u^{\mathrm{e}})_{\mathrm{R}} = \frac{3-2\beta}{(1-\beta)(2-\beta)h}(u_{k_{\mathrm{R}}+1} - u_{\mathrm{R}}^{\mathrm{e}}) - \frac{1-\beta}{(2-\beta)h}(u_{k_{\mathrm{R}}+2} - u_{k_{\mathrm{R}}+1}) \leqslant 0, \tag{20}$$

  thus

  $$\frac{3-2\beta}{(1-\beta)(2-\beta)h}(u_{k_{\mathrm{R}}+1} - u_{\mathrm{R}}^{\mathrm{e}}) \leqslant \frac{1-\beta}{(2-\beta)h}(u_{k_{\mathrm{R}}+2} - u_{k_{\mathrm{R}}+1}).$$

  Moreover, due to the Laplacian inequality,

  $$u_{k_{\mathrm{R}}+2} - u_{k_{\mathrm{R}}+1} \leqslant \frac{u_{k_{\mathrm{R}}+1} - u_{\mathrm{R}}^{\mathrm{e}}}{1-\beta},$$

  thus

  $$\frac{3-2\beta}{(1-\beta)}(u_{k_{\mathrm{R}}+1} - u_{\mathrm{R}}^{\mathrm{e}}) \leqslant u_{k_{\mathrm{R}}+1} - u_{\mathrm{R}}^{\mathrm{e}}.$$

  Therefore $u_{k_{\mathrm{R}}+1} - u_{\mathrm{R}}^{\mathrm{e}} \leqslant 0$, from which we deduce that $u_{k_{\mathrm{R}}+1} = u_{\mathrm{R}}^{\mathrm{e}} = u_{i_{\min}}$ and thus $u_{k_{\mathrm{R}}+2} = u_{\mathrm{R}}^{\mathrm{e}} = u_{i_{\min}}$ also. Using the Laplacian inequality in the exterior subdomain we deduce that all point values in the right exterior subdomain are

equal to $u_{i_{\min}}$. The Laplacian inequality on the first grid point yields:

$$\frac{2u_1 - u_2}{h^2} \geqslant 0,$$

therefore all the values of u are positive.

- *if the minimum is reached on a grid point in the exterior domain:*
  Let us assume for instance that it is located on the left side. The right side would be treated in the same way. Following the same reasoning as above, we deduce directly that all values in the left exterior subdomain are equal and positive. Therefore, as the minimum is positive, then all values of u are positive.
- *if the minimum is reached on an exterior interface point.*
  Let us assume for instance that it is reached on $u_{\mathrm{R}}^{\mathrm{e}}$. We have $u_{\mathrm{R}}^{\mathrm{e}} - u_{\mathrm{R}}^{\mathrm{c}} \leqslant 0$ and due to (19a)–(19d), we can write:

  $$0 \geqslant S(u_{\mathrm{R}}^{\mathrm{e}} - u_{\mathrm{R}}^{\mathrm{c}}) \geqslant \sigma_{\mathrm{e}}(\partial_{\mathbf{n}} u^{\mathrm{e}})_{\mathrm{R}}.$$

  Now we use the same reasoning as in the case where the minimum is on a grid point in the interior subdomain, from inequality (20) and conclude that all values in the right exterior subdomain are positive and equal to the minimum value. Consequently all values of u are positive.
- *if the minimum is reached on an interior interface point.*
  Suppose for instance that $u_{\mathrm{L}}^{\mathrm{c}}$ is the minimum. Due to the preliminary result in Subsection 3.3.1, the normal derivative $(\partial_{\mathbf{n}} u^{\mathrm{c}})_{\mathrm{L}}$ is then negative. Due to (19a)–(19d), we can write:

  $$0 \geqslant \sigma_{\mathrm{c}}(\partial_{\mathbf{n}} u^{\mathrm{c}})_{\mathrm{L}} \geqslant S(u_{\mathrm{L}}^{\mathrm{e}} - u_{\mathrm{L}}^{\mathrm{c}}) \geqslant \sigma_{\mathrm{e}}(\partial_{\mathbf{n}} u^{\mathrm{e}})_{\mathrm{L}}.$$

  Therefore $u_{\mathrm{L}}^{\mathrm{e}} = u_{\mathrm{L}}^{\mathrm{c}}$. We conclude with the same reasoning than in the previous case that all values of u are positive.

We conclude that the matrix A is monotone and thus invertible.

3.3.3. Estimation of $\|A^{-1}\|_\infty$

Since all the coefficients of A^{-1} are positive we can write $\left\|A^{-1}\right\|_\infty = \max\limits_{1\leqslant i\leqslant N} \sum\limits_{j=1}^{N} a_{i,j}^{-1}$. We want to build an array v such that

$$Av = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

in order to have: $\sum\limits_{j=1}^{N} a_{i,j}^{-1} = v_i$. We look for v as a quadratic function with respect to x.

For values on the left exterior subdomain, including the interface point values, we set

$$v_i = -x_i^2/2 + Cx_i + D,$$

for values on the interior subdomain, including the interface point values v_i is such that

$$v_i = -x_i^2/2 + C'x_i + D',$$

and for values on the right exterior subdomain, including the interface point values, we write

$$v_i = -x_i^2/2 + C''x_i + D''.$$

We check that the discrete Laplacian equalities are satisfied for $(v_i)_{i=2,N-1}$

$$-\frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} = 1 \text{ for all } i \in \{2, k_{\mathrm{L}} - 2\} \cup \{k_{\mathrm{L}} + 1, k_{\mathrm{R}} - 1\} \cup \{k_{\mathrm{R}} + 2, N - 1\},$$

$$-\frac{\dfrac{v_{\mathrm{L}}^{\mathrm{e}} - v_{k_{\mathrm{L}}-1}}{(1-\alpha)h} - \dfrac{v_{k_{\mathrm{L}}-1} - v_{k_{\mathrm{L}}-2}}{h}}{\dfrac{(2-\alpha)h}{2}} = 1, \quad -\frac{\dfrac{v_{k_{\mathrm{L}}+1} - v_{k_{\mathrm{L}}}}{h} - \dfrac{v_{k_{\mathrm{L}}} - v_{\mathrm{L}}^{\mathrm{c}}}{\alpha h}}{\dfrac{1+\alpha}{2}} = 1,$$

$$-\frac{\dfrac{v_{\mathrm{R}}^{\mathrm{c}} - v_{k_{\mathrm{R}}}}{\beta h} - \dfrac{v_{k_{\mathrm{R}}} - v_{k_{\mathrm{R}}-1}}{h}}{\dfrac{1+\beta}{2}} = 1, \quad -\frac{\dfrac{v_{k_{\mathrm{R}}+2} - v_{k_{\mathrm{R}}+1}}{h} - \dfrac{v_{k_{\mathrm{R}}+1} - v_{\mathrm{R}}^{\mathrm{e}}}{(1-\beta)h}}{\dfrac{(2-\beta)h}{2}} = 1.$$

In order to satisfy the same relationship for the lines v_1 and v_N, corresponding to the two ends of the domain, we impose

$$D = 0,$$
$$D'' = -C'' + 1/2.$$

For the lines corresponding to the jump conditions, inserting (14)–(17) in the relationships (19) we get

$$\begin{aligned}
S(v_{\mathrm{L}}^{\mathrm{e}} - v_{\mathrm{L}}^{\mathrm{c}}) + \sigma_{\mathrm{e}} \frac{3-2\alpha}{(1-\alpha)(2-\alpha)h} v_{\mathrm{L}}^{\mathrm{e}} - \sigma_{\mathrm{e}} \frac{2-\alpha}{(1-\alpha)h} v_{k_{\mathrm{L}}-1} + \sigma_{\mathrm{e}} \frac{1-\alpha}{(2-\alpha)h} v_{k_{\mathrm{L}}-2} &= 1,\\
S(v_{\mathrm{R}}^{\mathrm{e}} - v_{\mathrm{R}}^{\mathrm{c}}) - \sigma_{\mathrm{e}} \frac{2-\beta}{(1-\beta)h} v_{k_{\mathrm{R}}+1} + \sigma_{\mathrm{e}} \frac{3-2\beta}{(1-\beta)(2-\beta)h} v_{\mathrm{R}}^{\mathrm{e}} + \sigma_{\mathrm{e}} \frac{1-\beta}{(2-\beta)h} v_{k_{\mathrm{R}}+2} &= 1,\\
-\sigma_{\mathrm{c}} \frac{1+\alpha}{\alpha h} v_{k_{\mathrm{L}}} + \sigma_{\mathrm{c}} \frac{1+2\alpha}{\alpha(\alpha+1)h} v_{\mathrm{L}}^{\mathrm{c}} + \sigma_{\mathrm{c}} \frac{\alpha}{(1+\alpha)h} v_{k_{\mathrm{L}}+1} - S(v_{\mathrm{L}}^{\mathrm{e}} - v_{\mathrm{L}}^{\mathrm{c}}) &= 1,\\
\sigma_{\mathrm{c}} \frac{1+2\beta}{\beta(\beta+1)h} v_{\mathrm{R}}^{\mathrm{c}} - \sigma_{\mathrm{c}} \frac{1+\beta}{\beta h} v_{k_{\mathrm{R}}} + \sigma_{\mathrm{c}} \frac{\beta}{(\beta+1)h} v_{k_{\mathrm{R}}-1} - S(v_{\mathrm{R}}^{\mathrm{e}} - v_{\mathrm{R}}^{\mathrm{c}}) &= 1.
\end{aligned}$$

Due to the relationships (18), the above expressions simplify into:

$$\begin{aligned}
S\left((C - C')x_{\mathrm{L}} + D - D'\right) + \sigma_{\mathrm{e}} C &= 1,\\
S\left((C'' - C')x_{\mathrm{R}} + D'' - D'\right) - \sigma_{\mathrm{e}} C'' &= 1,\\
\sigma_{\mathrm{c}} C' - S\left((C - C')x_{\mathrm{L}} + D - D'\right) &= 1,\\
-\sigma_{\mathrm{c}} C' - S\left((C'' - C')x_{\mathrm{R}} + D'' - D'\right) &= 1.
\end{aligned}$$

With such coefficients C, D, C', D', C'' and D'' that do not depend on h we have

$$\left\| A^{-1} \right\|_{\infty} = \max_{1 \leqslant i \leqslant N} v_i \leqslant \max\left(\frac{C^2}{2} + D, \frac{C'^2}{2} + D', \frac{C''^2}{2} + D'' \right).$$

3.4. Proof of second-order convergence

Proposition 3.1. *The numerical solution to the linear stationary problem converges with a second-order accuracy in L^{∞}-norm to the exact solution to the problem.*

Proof. We denote by z the i-th row of A^{-1}, *i.e.* $z_j = A^{-1}_{i,j}$. We use the same indexing system for z as for the unknowns of the linear system. We define the Kronecker symbol for interface values as:

$$\delta_i^{\mathrm{L,c}} = 1 \text{ if } i \text{ is the index of the left interior interface value, 0 otherwise,}$$
$$\delta_i^{\mathrm{R,c}} = 1 \text{ if } i \text{ is the index of the right interior interface value, 0 otherwise,}$$
$$\delta_i^{\mathrm{L,e}} = 1 \text{ if } i \text{ is the index of the left exterior interface value, 0 otherwise,}$$
$$\delta_i^{\mathrm{R,e}} = 1 \text{ if } i \text{ is the index of the right exterior interface value, 0 otherwise.}$$

We multiply the row vector z with the columns of A.

The columns 1 to $k_{\mathrm{L}} - 1$ yield:

$$\begin{aligned}
2z_1 - z_2 &= h^2 \delta_i^1,\\
-z_1 + 2z_2 - z_3 &= h^2 \delta_i^2,\\
&\vdots \quad \vdots\\
-z_{k_{\mathrm{L}}-4} + 2z_{k_{\mathrm{L}}-3} - z_{k_{\mathrm{L}}-2} &= h^2 \delta_i^{k_{\mathrm{L}}-3},\\
-z_{k_{\mathrm{L}}-3} + 2z_{k_{\mathrm{L}}-2} - \frac{2}{(2-\alpha)} z_{k_{\mathrm{L}}-1} + \sigma_{\mathrm{e}} h \frac{1-\alpha}{2-\alpha} z_{\mathrm{L}}^{\mathrm{e}} &= h^2 \delta_i^{k_{\mathrm{L}}-2},\\
-z_{k_{\mathrm{L}}-2} + \frac{2}{(1-\alpha)(2-\alpha)} z_{k_{\mathrm{L}}-1} + \frac{2}{(2-\alpha)} z_{k_{\mathrm{L}}-1} - \sigma_{\mathrm{e}} h \frac{2-\alpha}{1-\alpha} z_{\mathrm{L}}^{\mathrm{e}} &= h^2 \delta_i^{k_{\mathrm{L}}-1}.
\end{aligned}$$

The columns $k_{\mathrm{L}} + 1$ to $k_{\mathrm{R}} - 1$ yield:

$$
\begin{aligned}
-z_{k_L+1} + \frac{2}{(1+\alpha)} z_{k_L} + \frac{2}{\alpha(1+\alpha)} z_{k_L} - \sigma_c h \frac{1+\alpha}{\alpha} z_L^c &= h^2 \delta_i^{k_L}, \\
2z_{k_L+1} - z_{k_L+2} - \frac{2}{(1+\alpha)} z_{k_L} + \sigma_c h \frac{\alpha}{1+\alpha} z_L^c &= h^2 \delta_i^{k_L+1}, \\
-z_{k_L+1} + 2z_{k_L+2} - z_{k_L+3} &= h 2 \delta_i^{k_L+2}, \\
\vdots \qquad \vdots & \\
-z_{k_R-3} + 2z_{k_R-2} - z_{k_R-1} &= h^2 \delta_i^{k_R-2}, \\
-z_{k_R-2} + 2z_{k_R-1} - \frac{2}{(1+\beta)} z_{k_R} + \sigma_c h \frac{\beta}{\beta+1} z_R^c &= h^2 \delta_i^{k_R-1}, \\
-z_{k_R-1} + \frac{2}{\beta(1+\beta)} z_{k_R} + \frac{2}{(1+\beta)} z_{k_R} - \sigma_c h \frac{1+\beta}{\beta} z_R^c &= h^2 \delta_i^{k_R},
\end{aligned}
$$

and the columns $k_R + 1$ to N:

$$
\begin{aligned}
\frac{2}{2-\beta} z_{k_R+1} + \frac{2}{(1-\beta)(2-\beta)} z_{k_R+1} - \sigma_e h \frac{2-\beta}{1-\beta} z_R^e - z_{k_R+2} &= h^2 \delta_i^{k_R+1}, \\
2z_{k_R+2} - z_{k_R+3} - \frac{2}{2-\beta} z_{k_R+1} + \sigma_e h \frac{1-\beta}{2-\beta} z_R^e &= h^2 \delta_i^{k_R+2}, \\
-z_{k_R+2} + 2z_{k_R+3} - z_{k_R+4} &= h^2 \delta_i^{k_R+3}, \\
\vdots \qquad \vdots & \\
-z_{N-2} + 2z_{N-1} - z_N &= h^2 \delta_i^{N-1}, \\
-z_{N-1} + 2z_N &= h^2 \delta_i^{N}.
\end{aligned}
$$

The columns corresponding to interface values yield:

$$
\begin{aligned}
\left(S + \sigma_e \frac{3-2\alpha}{(1-\alpha)(2-\alpha)h} \right) z_L^e - \frac{2}{(1-\alpha)(2-\alpha)h^2} z_{k_L-1} - S z_L^c &= \delta_i^{L,e}, \\
-S z_L^e + \left(S + \sigma_c \frac{1+2\alpha}{\alpha(\alpha+1)h} \right) z_L^c - \frac{2}{\alpha(1+\alpha)h^2} z_{k_L} &= \delta_i^{L,c}, \qquad (21) \\
-\frac{2}{\beta(1+\beta)h^2} z_{k_R} + \left(S + \sigma_c \frac{1+2\beta}{\beta(\beta+1)h} \right) z_R^c - S z_R^e &= \delta_i^{R,c}, \qquad (22) \\
-S z_R^c + \left(S + \sigma_e \frac{3-2\beta}{(1-\beta)(2-\beta)h} \right) z_R^e - \frac{2}{(1-\beta)(2-\beta)h^2} z_{k_R+1} &= \delta_i^{R,e}.
\end{aligned}
$$

We now want to prove that all the coefficients of z corresponding to the interior subdomain, that is, z_j with $k_L \leqslant j \leqslant k_R$, are $O(h)$. The lines (21) and (22) can be rewritten:

$$
\begin{aligned}
\left(S + \sigma_c \frac{1+2\alpha}{\alpha(\alpha+1)h} \right) z_L^c &= \frac{2}{\alpha(1+\alpha)h^2} z_{k_L} + S z_L^e + \delta_i^{L,c}, \\
\left(S + \sigma_c \frac{1+2\beta}{\beta(\beta+1)h} \right) z_R^c &= \frac{2}{\beta(1+\beta)h^2} z_{k_R} + S z_R^e + \delta_i^{R,c}.
\end{aligned}
$$

Because we have proved that $\|A^{-1}\|_\infty$ is bounded independently of h we can write that z_L^e, z_L^c, z_R^c and z_R^e are all $O(1)$. We also know, since the matrix A is monotone, that all the coefficients z_j are positive. Consequently, z_{k_L} and z_{k_R} are $O(h)$. Now, if we sum the lines corresponding to columns $k_L + 1$ to $k_R - 1$ we get:

$$
z_{k_L+1} - \frac{2}{1+\alpha} z_{k_L} + \sigma_c h \frac{\alpha}{1+\alpha} z_L^c + z_{k_R-1} - \frac{2}{1+\beta} z_{k_R} + \sigma_c h \frac{\beta}{\beta+1} z_R^c = \sum_{i=k_L+1}^{k_R-1} h^2 \delta_i^j.
$$

Then

$$
z_{k_L+1} + \sigma_c h \frac{\alpha}{1+\alpha} z_L^c + z_{k_R-1} + \sigma_c h \frac{\beta}{\beta+1} z_R^c = \frac{2}{1+\alpha} z_{k_L} + \frac{2}{1+\beta} z_{k_R} + \sum_{i=k_L+1}^{k_R-1} h^2 \delta_i^j,
$$

from which we infer that z_{k_L+1} and z_{k_R-1} are $O(h)$. Similarly, doing successive sums, we prove that all the coefficients z_j with $k_L \leqslant j \leqslant k_R$ are $O(h)$. We would prove similarly the same result for the other coefficients z_j with $1 \leqslant j \leqslant k_L - 1$ or $k_R + 1 \leqslant j \leqslant N$ but for the sake of brevity we do not write the proof here.

We eventually conclude that all the coefficients of the row i of A^{-1} at the exception of z_L^e, z_L^c, z_R^c and z_R^e are of order $O(h)$, without any coefficient of the type $\frac{1}{\alpha}$, $\frac{1}{1-\alpha}$, $\frac{1}{\beta}$ or $\frac{1}{1-\beta}$ that may be singular if α (resp. β) tend to 0 or 1 and lead to instabilities in the convergence. Consequently the array containing the punctual difference between the numerical solution and the exact solution on each discretization point, which is equal to the product of the array of the truncation errors on each point by A^{-1}, is of order $O(h^2)$, which ensures the second-order accuracy in L^∞-norm of the method.

4. Convergence result for the one-dimensional dynamical model

In Subsection 4.1 we rewrite the linear problem to solve as an interface problem. Then, in Subsection 4.2, we prove that in one dimension the matrix appearing in the formulation of the interface problem is symmetric and that its spectral radius is bounded by one. Theses properties are crucial to prove the stability of the numerical method, and this subsection is the technical core of the proof. Then in Subsection 4.3 we prove that the scheme is converging with first-order accuracy, with techniques similar to the proof of existence and uniqueness in Kavian et al. [10]. The geometrical configuration and notations are the same as in Section 3, see Fig. 7.

4.1. Formulation of the discrete problem as an interface problem

In order to analyze the convergence of the numerical method to the exact solution to the electropermeabilization model, we need to rewrite the linear system as an interface problem. To this purpose we rewrite the whole linear system by separating the unknowns in the interior domain from the unknowns in the exterior domain. We denote by u_g^e and u_g^c the respective unknowns on grid points in the exterior and interior domains, and u_p^e and u_p^c the unknowns on interface points in the exterior and interior domains. This leads to

$$\begin{pmatrix} \Delta_g^e & 0 & \Delta_p^e & 0 \\ 0 & \Delta_g^c & 0 & \Delta_p^c \\ \Phi_g^e & \Phi_g^c & \Phi_p^e & \Phi_p^c \\ 0 & \Phi_g^c & \frac{C}{dt}\mathrm{Id} & \Phi_p^c - \frac{C}{dt}\mathrm{Id} \end{pmatrix} \begin{pmatrix} u_g^e \\ u_g^c \\ u_p^e \\ u_p^c \end{pmatrix}^{n+1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{C}{dt}\mathrm{Id} - S^n & -\frac{C}{dt}\mathrm{Id} + S^n \end{pmatrix} \begin{pmatrix} u_g^e \\ u_g^c \\ u_p^e \\ u_p^c \end{pmatrix}^n + \begin{pmatrix} \tilde{g} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{23}$$

The two first block lines represent the discretization of the Laplacian operator on the exterior and interior grid points. Δ_g^e and Δ_p^e are the block matrices corresponding to the discretization of the Laplacian in the exterior subdomain, and Δ_p^c and Δ_g^c the block matrices corresponding to the discretization of the Laplacian in the interior subdomain. Remark that Δ_g^e and Δ_g^c are invertible since they correspond to the discretization of the Laplacian on grid points in the exterior and interior domain. The source term $\tilde{g}$ contains the terms accounting for boundary conditions.

The third block line represents the discretization of the flux equality across the interface. More precisely,

$$\Phi_g^e u_g^e + \Phi_p^e u_p^e = \sigma_e \partial_{\mathbf{n}} u_e,$$
$$\Phi_g^c u_g^c + \Phi_p^c u_p^c = -\sigma_c \partial_{\mathbf{n}} u_c,$$

where $\partial_{\mathbf{n}} u_e$ and $\partial_{\mathbf{n}} u_c$ are vectors containing the discretization of the normal derivative to the interface respectively in the exterior and interior domains. The last block line represents the evolution of the jump of the solution across the interface. S^n is the diagonal matrix containing the values of $(S)_{i,j,\gamma}^n$ at each interface point.

Using the invertibility of Δ_g^e and Δ_p^c we write

$$u_g^e = -(\Delta_g^e)^{-1}(\Delta_p^e u_p^e - \tilde{g}),$$
$$u_g^c = -(\Delta_g^c)^{-1}\Delta_p^c u_p^c.$$

Injecting these expressions in the discrete approximations of fluxes yields

$$\sigma_e \partial_{\mathbf{n}} u_e = -\Phi_g^e(\Delta_g^e)^{-1}(\Delta_p^e u_p^e - \tilde{g}) + \Phi_p^e u_p^e,$$
$$\sigma_c \partial_{\mathbf{n}} u_c = \Phi_g^c(\Delta_g^c)^{-1}\Delta_p^c u_p^c - \Phi_p^c u_p^c,$$

and we define the matrices Λ_c, Λ_e and Λ_0 as

$$\sigma_c \Lambda_c = \Phi_g^c(\Delta_g^c)^{-1}\Delta_p^c - \Phi_p^c,$$
$$\sigma_e \Lambda_e = -\Phi_g^e(\Delta_g^e)^{-1}\Delta_p^e + \Phi_p^e,$$
$$\sigma_e \Lambda_0 = \Phi_g^e(\Delta_g^e)^{-1}.$$

Note that Λ_c and Λ_e correspond to discrete versions of the Dirichlet-to-Neumann operators given in Kavian et al. We have

$$\sigma_e \Lambda_0 \tilde{g} + \sigma_e \Lambda_e u_p^e + \sigma_c \Lambda_c u_p^c = 0. \tag{24}$$

We do not write the proof here for the sake of brevity, but one can prove that Λ_e is invertible. Let us notice also that the matrix $\sigma_e\Lambda_e + \sigma_c\Lambda_c$ is invertible. Indeed, it is the linear application that associates to a vector w_p representing the values at the interface points (both exterior and interior, assuming that there is no discontinuity across the interface for these values), the jump of the exterior and interior normal derivatives to the interface of the solution to the linear problems

$$\Delta_g^c w_g^c = -\Delta_p^c w_p,$$
$$\Delta_g^e w_g^e = -\Delta_p^e w_p.$$

Consequently, the matrix $\left(\mathrm{Id} + \frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c\right) = \frac{1}{\sigma_e}\Lambda_e^{-1}\left(\sigma_e\Lambda_e + \sigma_c\Lambda_c\right)$ is also invertible. Therefore, using (24) and denoting $[u] = u_p^e - u_p^c$ we write

$$u_p^e = \frac{1}{\sigma_e}\Lambda_e^{-1}(-\sigma_e\Lambda_0\tilde{g} + \sigma_c\Lambda_c u_p^c),$$
$$u_p^c = u_p^e - [u] = -\frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c u_p^c + \Lambda_e^{-1}\Lambda_0\tilde{g} - [u].$$

Therefore

$$u_p^c = \left(\mathrm{Id} + \frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c\right)^{-1}\left(-[u] + \frac{1}{\sigma_e}\Lambda_e^{-1}\Lambda_0\tilde{g}\right). \tag{25}$$

The matrix $-\left(\mathrm{Id} + \frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c\right)^{-1}$ is the linear application that associates to an array $[v]$, representing a jump across the interface, the array v_p^c representing the interior values at the interface, satisfying the linear problem:

$$\begin{pmatrix} \Delta_g^e & 0 & \Delta_p^e & 0 \\ 0 & \Delta_g^c & 0 & \Delta_p^c \\ \Phi_g^e & \Phi_g^c & \Phi_p^e & \Phi_p^c \end{pmatrix}\begin{pmatrix} u_g^e \\ u_g^c \\ u_p^e \\ u_p^c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

with $v_p^e - v_p^c = [v]$.

The temporal evolution of $[u]$ can be expressed as

$$\frac{C}{dt}[u]^{n+1} - \sigma_c\Lambda_c u_p^c = \left(\frac{C}{dt}\mathrm{Id} - S^n\right)[u]^n. \tag{26}$$

The evolution of $[u]$ can be rewritten using formula (25) as

$$\left(\mathrm{Id} + \frac{dt}{C}\sigma_c\Lambda_c\left(\mathrm{Id} + \frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c\right)^{-1}\right)[u]^{n+1} = \left(\mathrm{Id} - \frac{dt}{C}S^n\right)[u]^n + dtF, \tag{27}$$

with $F = -\frac{\sigma_c}{C}\Lambda_c\left(\mathrm{Id} + \frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c\right)^{-1}\Lambda_e^{-1}\Lambda_0\tilde{g}$.

In the following, for the sake of brevity, we will note

$$M = \left(\mathrm{Id} + \frac{dt}{C}\sigma_c\Lambda_c\left(\mathrm{Id} + \frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c\right)^{-1}\right)^{-1}.$$

The relationship (27) can thus be re-written

$$[u]^{n+1} = M\left(\mathrm{Id} - \frac{dt}{C}S^n\right)[u]^n + dt\,MF. \tag{28}$$

Proposition 4.1. *Problem (23) and problem (28) are equivalent.*

Note that $\rho(\mathrm{Id} - \frac{dt}{C}S^n) < 1$ if $|1 - \frac{dt}{C}S| < 1$ for all values of $(S)_{i,j,\gamma}$ with $\gamma = \mathrm{E}, \mathrm{W}, \mathrm{S}, \mathrm{N}$, which is true notably if $dt < 2\frac{C}{S_L + S_{ir}}$.

The expression

$$[u]^{n+1} = M\left(\mathrm{Id} - \frac{dt}{C}S^n\right)[U]^n + dtF,$$

is linearly equivalent to the initial linear system (26), which is consistent with the equations to solve. Therefore, by injecting the exact solution $\tilde{u}$ in the latter formula, one can write

$$[\tilde{u}](t^{n+1,x}) = M\left(\mathrm{Id} - \frac{dt}{C}S(X(t^n,x))\right)[\tilde{u}](t^n,x) + dtF + O(dt), \quad \forall x \in \Gamma.$$

It is proven in [10] that the exact solution is bounded in $C([0,T], L^2(\Gamma))$, therefore we deduce that $\|F\|_2 = O(1)$.

4.2. Preliminary results on the matrix properties

Lemma 4.2. *In one-dimension, the matrix* $\Lambda_c\left(\mathrm{Id} + \frac{\sigma_c}{\sigma_e}\Lambda_e^{-1}\Lambda_c\right)^{-1}$ *is symmetric and all its eigenvalues are positive.*

Proof. We begin by proving that Λ_c is symmetric and has only positive eigenvalues, i.e.

$$\forall v_p, w_p, \quad (\Lambda_c v_p, w_p) = (\Lambda_c w_p, v_p) \text{ and } (\Lambda_c v_p, v_p) \geq 0.$$

We denote by v_L, and w_L two values on x_L, v_R and w_R two values on x_R. Let v_g and w_g be the solutions to the linear problems

$$\Delta_g^c v_g = -\Delta_p^c \begin{pmatrix} v_L \\ v_R \end{pmatrix} \text{ and } \Delta_g^c w_g = -\Delta_p^c \begin{pmatrix} w_L \\ w_R \end{pmatrix}.$$

The arrays $-\Lambda_c v_p$ and $-\Lambda_c w_p$ represent the discrete normal derivatives at the interface. We thus can write:

$$\Phi_g^c v_g + \Phi_p^c v_p = -\Lambda_c v_p \text{ and } \Phi_g^c w_g + \Phi_p^c w_p = -\Lambda_c w_p.$$

In one dimension we can write:

$$v_g = \begin{pmatrix} v_{k_L} \\ v_{k_L+1} \\ \vdots \\ v_{k_R} \end{pmatrix}, \text{ and } w_g = \begin{pmatrix} w_{k_L} \\ w_{k_L+1} \\ \vdots \\ w_{k_R} \end{pmatrix}.$$

We sum over every grid point i inside the interior domain the product $(\Delta v)_i w_i$ where $(\Delta v)_i$ is the discrete Laplacian of v defined by $(\Delta_g^c v_g + \Delta_p^c v_p)_i$. For the indices k_L and k_R, the Laplacian term is also multiplied by $(\alpha+1)/2$ or $(\beta+1)/2$ in order to obtain the same denominator as the interior terms, which will simplify the computations. Then we make a summation by parts.

$$\begin{aligned}
&\frac{(v_{k_L+1} - v_{k_L})}{h^2} w_{k_L} - \frac{(v_{k_L} - v_L)}{\alpha h^2} w_{k_L} + \sum_{i=k_L+1}^{k_R-1} \frac{(v_{i+1} - v_i) - (v_i - v_{i-1})}{h^2} w_i \\
&\quad + \frac{(v_R - v_{k_R})}{\beta h^2} w_{k_R} - \frac{(v_{k_R} - v_{k_R-1})}{h^2} w_{k_R} = 0, \\
&-\frac{(v_{k_L} - v_L)}{\alpha h^2} w_{k_L} - \sum_{i=k_L}^{k_R-1} \frac{(v_{i+1} - v_i)}{h^2}(w_{i+1} - w_i) + \frac{(v_R - v_{k_R})}{\beta h^2} w_{k_R} = 0.
\end{aligned}$$

We use the fact that the discrete Laplacian is zero at points x_{k_L} and x_{k_R}:

$$\begin{aligned}
&\frac{v_R - v_{k_R}}{\beta h} - \frac{v_{k_R} - v_{k_R-1}}{h} = 0, \\
&\frac{v_{k_L+1} - v_{k_L}}{h} - \frac{v_{k_L} - v_L}{\alpha h} = 0.
\end{aligned}$$

Furthermore, the discrete second order normal derivative to the interface at points x_L and x_R are

$$\begin{aligned}
(\partial_{\mathbf{n}} v)_L &= -(\partial_x v)_L = \frac{1+2\alpha}{\alpha(\alpha+1)h}(v_{k_L} - v_L) - \frac{\alpha}{(1+\alpha)h}(v_{k_L+1} - v_{k_L}), \\
(\partial_{\mathbf{n}} v)_R &= (\partial_x v)_R = \frac{1+2\beta}{\beta(\beta+1)h}(v_R - v_{k_R}) - \frac{\beta}{(\beta+1)h}(v_{k_R} - v_{k_R-1}).
\end{aligned}$$

With the help of these latter equations we prove that

$$(\partial_{\mathbf{n}} v)_L = -\frac{(v_{k_L} - v_L)}{\alpha h}, \tag{29}$$

$$(\partial_{\mathbf{n}} v)_R = \frac{(v_R - v_{k_R})}{\beta h}. \tag{30}$$

Thus,

$$\frac{(\partial_{\mathbf{n}} v)_{\mathrm{L}}}{h} w_{k_{\mathrm{L}}} + \frac{(\partial_{\mathbf{n}} v)_{\mathrm{R}}}{h} w_{k_{\mathrm{R}}} = \sum_{i=k_{\mathrm{L}}}^{k_{\mathrm{R}}-1} \frac{(v_{i+1} - v_i)}{h^2}(w_{i+1} - w_i). \tag{31}$$

We can consequently write

$$\frac{(\partial_{\mathbf{n}} v)_{\mathrm{L}}}{h} (w_{\mathrm{L}} - \alpha h(\partial_{\mathbf{n}} w)_{\mathrm{L}}) + \frac{(\partial_{\mathbf{n}} v)_{\mathrm{R}}}{h} (w_{\mathrm{R}} - \beta h(\partial_{\mathbf{n}} w)_{\mathrm{R}}) = \sum_{i=k_{\mathrm{L}}}^{k_{\mathrm{R}}-1} \frac{(v_{i+1} - v_i)}{h^2}(w_{i+1} - w_i),$$

which leads to

$$\frac{(\partial_{\mathbf{n}} v)_{\mathrm{L}}}{h} w_{\mathrm{L}} + \frac{(\partial_{\mathbf{n}} v)_{\mathrm{R}}}{h} w_{\mathrm{R}} = \sum_{i=k_{\mathrm{L}}}^{k_{\mathrm{R}}-1} \frac{(v_{i+1} - v_i)}{h^2}(w_{i+1} - w_i) + \alpha(\partial_{\mathbf{n}} v)_{\mathrm{L}}(\partial_{\mathbf{n}} w)_{\mathrm{L}} + \beta(\partial_{\mathbf{n}} v)_{\mathrm{R}}(\partial_{\mathbf{n}} w)_{\mathrm{R}}. \tag{32}$$

We conclude that

$$\frac{(\partial_{\mathbf{n}} v)_{\mathrm{L}}}{h} w_{\mathrm{L}} + \frac{(\partial_{\mathbf{n}} v)_{\mathrm{R}}}{h} w_{\mathrm{R}} = \frac{(\partial_{\mathbf{n}} w)_{\mathrm{L}}}{h} v_{\mathrm{L}} + \frac{(\partial_{\mathbf{n}} w)_{\mathrm{R}}}{h} v_{\mathrm{R}},$$

which is equivalent to

$$(\Lambda_{\mathrm{c}} v_{\mathrm{p}}, w_{\mathrm{p}}) = (\Lambda_{\mathrm{c}} w_{\mathrm{p}}, v_{\mathrm{p}}),$$

and therefore Λ_{c} is symmetric. Furthermore, taking $w_i = v_i$ for all i in (32) we obtain $(\Lambda_{\mathrm{c}} v_{\mathrm{p}}, v_{\mathrm{p}}) \geq 0$ which proves that Λ_{c} has only positive eigenvalues. We would prove similarly the same result for Λ_{e}.

Now we consider again the matrix $\Lambda_{\mathrm{c}} \left(\mathrm{Id} + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}\right)^{-1}$. Let v be an array of the same size as $[u]$. We denote by $w = \left(\mathrm{Id} + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}\right)^{-1} v$.

$$\begin{aligned}
\Lambda_{\mathrm{c}} \left(\mathrm{Id} + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}\right)^{-1} v \cdot v &= \Lambda_{\mathrm{c}} \left(\mathrm{Id} + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}\right)^{-1} v \cdot \left(\mathrm{Id} + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}\right) \left(\mathrm{Id} + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}\right)^{-1} v, \\
&= \Lambda_{\mathrm{c}} w \cdot \left(\mathrm{Id} + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}\right) w, \\
&= \Lambda_{\mathrm{c}} w \cdot w + \frac{\sigma_{\mathrm{c}}}{\sigma_{\mathrm{e}}} \Lambda_{\mathrm{e}} (\Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}} w) \cdot (\Lambda_{\mathrm{e}}^{-1} \Lambda_{\mathrm{c}}) \geq 0.
\end{aligned}$$

4.3. Convergence result in one dimension

We address the convergence of the numerical solution to the exact solution to the non-linear problem. We denote by $[u]^n$ the vector containing the values of the jump of the exact solution at the interface points at time t^n, and $[u]_h^n$ the vector containing the values of the jump of the numerical solution at the same points and same time. X^n and X_h^n are defined similarly as the exact and approximated vector values of X at these interface points. We denote the error in discrete L^2 norm at time step n by $E^n = \left\| [u]^n - [u]_h^n \right\|_2$. Because our method is formulated in the finite differences formalism, we need to assume that the solutions to the electropermeabilization model considered are smooth enough to have the truncation errors tending to zero when the grid spacing h tends to zero.

Lemma 4.3. *If $dt < \tau_{\mathrm{ep}}$, then for all $n \geq 0$ and on each interface point, we have $0 \leq X_h^n \leq 1$.*

Proof. The value of X_h^{n+1} is computed from X_h^n and $[u]_h^n$ with a first order explicit Euler scheme. The formula reads

$$X_h^{n+1} = X_h^n + dt \max\left(\frac{\beta([u]_h^n) - X_h^n}{\tau_{\mathrm{ep}}}, \frac{\beta([u]_h^n - X_h^n}{\tau_{\mathrm{res}}}\right).$$

We use the notation

$$X_h^{n+1} = X_h^n + dt \frac{\beta([u]_h^n) - X_h^n}{\tau^n},$$

with $\tau^n = \tau_{\mathrm{ep}}$ or τ_{res} depending on the sign of $\beta([u]_h^n) - X_h^n$.

We define $(X_h^n)^- = \max(0, -X_h^n)$ and multiply the latter equation by $(X_h^{n+1})^- + (X_h^n)^-$. We obtain:

$$\left((X_h^{n+1})^- + (X_h^n)^-\right)\frac{X_h^{n+1} - X_h^n}{dt} = \left((X_h^{n+1})^- + (X_h^n)^-\right)\frac{\beta([u]_h^n) - X_h^n}{\tau^n},$$

$$\frac{X_h^{n+1}(X_h^{n+1})^- - X_h^n(X_h^n)^-}{dt} = \left((X_h^{n+1})^- + (X_h^n)^-\right)\frac{\beta([u]_h^n) - X_h^n}{\tau^n} + \frac{X_h^n(X_h^{n+1})^-}{dt} - \frac{X_h^{n+1}(X_h^n)^-}{dt},$$

$$-\frac{((X_h^{n+1})^-)^2 - ((X_h^n)^-)^2}{dt} = \left((X_h^{n+1})^- + (X_h^n)^-\right)\frac{\beta([u]_h^n)}{\tau^n} - \frac{X_h^n(X_h^n)^-}{\tau^n} + X_h^n(X_h^{n+1})^-\left(\frac{1}{dt} - \frac{1}{\tau^n}\right) - \frac{X_h^{n+1}X_h^n}{dt}.$$

The terms on the first line of the right hand-side are positive, notably because $\beta \geqslant 0$. We assume that $dt \leqslant \tau^n$. If $X_h^n > 0$, or if $X_h^n < 0$ and $X_h^{n+1} < 0$ then the term on the second line of the right hand-side is also positive and $(X_h^{n+1})^- < (X_h^n)^-$. On the contrary, if $X_h^n > 0$ and $X_h^{n+1} > 0$ then one has also $(X_h^{n+1})^- < (X_h^n)^-$. Consequently the sequence $((X_h^n)^-)_{n\geqslant 0}$ is decreasing. Because $(X_h^n)^- \geqslant 0$ for all $n \geqslant 0$ and $(X_h^0)^- = 0$ one concludes that $X_h^n \geqslant 0$ for all $n \geqslant 0$.

With the same reasoning, by considering $(X_h^n - 1)^+ = \max(0, X_h^n - 1)$ and using the fact that $\beta - 1 \leqslant 0$, we can prove that $X_h^n \leqslant 1$ for all $n \geqslant 0$.

Lemma 4.4. *For all $T > 0$ there exists a constant $K(T)$ depending on T, β and τ_{ep} but not on other parameters, such that on every interface point the following inequality is satisfied for all $0 \leqslant n \leqslant \frac{T}{dt}$*

$$\sup_{i\leqslant n}\left|X^i - X_h^i\right| \leqslant K(T)\sup_{i\leqslant n}\left|[u]^i - [u]_h^i\right| + O(dt). \tag{33}$$

Proof. The function β' is bounded on $]-\infty, +\infty[$, and we denote by $\beta'_{max} = \sup|\beta'(x)|$.

$$\begin{aligned}\left|X^{n+1} - X_h^{n+1}\right| \leqslant{}& \left|X^n - X_h^n\right| + \left|\varepsilon_2^n\right| \\ &+ dt\left|\max\left(\frac{\beta([u]^n) - X^n}{\tau_{\text{ep}}}, \frac{\beta([u]^n) - X^n}{\tau_{\text{res}}}\right) - \max\left(\frac{\beta([u]_h^n) - X^n}{\tau_{\text{ep}}}, \frac{\beta([u]_h^n) - X^n}{\tau_{\text{res}}}\right)\right| \\ &+ dt\left|\max\left(\frac{\beta([u]_h^n) - X^n}{\tau_{\text{ep}}}, \frac{\beta([u]_h^n) - X^n}{\tau_{\text{res}}}\right) - \max\left(\frac{\beta([u]_h^n) - X_h^n}{\tau_{\text{ep}}}, \frac{\beta([u]_h^n) - X_h^n}{\tau_{\text{res}}}\right)\right|.\end{aligned}$$

Let us observe that

$$\begin{aligned}\left|\max\left(\frac{\beta([u]^n) - X^n}{\tau_{\text{ep}}}, \frac{\beta([u]^n) - X^n}{\tau_{\text{res}}}\right) - \max\left(\frac{\beta([u]_h^n) - X^n}{\tau_{\text{ep}}}, \frac{\beta([u]_h^n) - X^n}{\tau_{\text{res}}}\right)\right| &\leqslant \frac{\left|\beta([u]^n) - \beta([u]_h^n)\right|}{\tau_{\text{ep}}} \\ &\leqslant \left\|\beta'\right\|_\infty \frac{\left|[u]^n - [u]_h^n\right|}{\tau_{\text{ep}}},\end{aligned}$$

and similarly

$$\left|\max\left(\frac{\beta([u]_h^n) - X^n}{\tau_{\text{ep}}}, \frac{\beta([u]_h^n) - X^n}{\tau_{\text{res}}}\right) - \max\left(\frac{\beta([u]_h^n) - X_h^n}{\tau_{\text{ep}}}, \frac{\beta([u]_h^n) - X_h^n}{\tau_{\text{res}}}\right)\right| \leqslant \frac{\left|X^n - X_h^n\right|}{\tau_{\text{ep}}}.$$

Consequently

$$\left|X^{n+1} - X_h^{n+1}\right| \leqslant \left(1 + \frac{dt}{\tau_{\text{ep}}}\right)\left|X^n - X_h^n\right| + dt\frac{\left\|\beta'\right\|_\infty}{\tau_{\text{ep}}}\left|[u]^n - [u]_h^n\right| + \left|\varepsilon_2^n\right|.$$

Now we use a discrete Gronwall lemma to obtain

$$\begin{aligned}\left|X^n - X_h^n\right| &\leqslant \exp\left(\frac{n\,dt}{\tau_{\text{ep}}}\right)\left|X^0 - X_h^0\right| + \sum_{i=0}^{n-1}\exp\left(\frac{n\,dt - i\,dt}{\tau_{\text{ep}}}\right)\left(\frac{dt\,\left\|\beta'\right\|_\infty}{\tau_{\text{ep}}}\left|[u]^i - [u]_h^i\right| + \left|\varepsilon_2^i\right|\right), \\ &\leqslant \sum_{i=0}^{n-1}\exp\left(\frac{T}{\tau_{\text{ep}}}\right)\left(\frac{dt\,\left\|\beta'\right\|_\infty}{\tau_{\text{ep}}}\left|[u]^i - [u]_h^i\right| + \left|\varepsilon_2^i\right|\right), \\ \sup_{i\leqslant n}\left|X^i - X_h^i\right| &\leqslant \left\|\beta'\right\|_\infty\frac{T}{\tau_{\text{ep}}}\exp\left(\frac{T}{\tau_{\text{ep}}}\right)\sum_{i=0}^{n-1}dt\left|[u]^i - [u]_h^i\right| + \sum_{i=0}^{n-1}\exp\left(\frac{T}{\tau_{\text{ep}}}\right)\left|\varepsilon_2^i\right|,\end{aligned}$$

from which we deduce the result of the lemma.

Lemma 4.5. *The sequences* $\left\|[u]_h^n\right\|_2$ *and* $\left\|[u]_h^n\right\|_\infty$ *are bounded independently of* n *and* h *in one-dimension if* $\left|1-\frac{dt}{C}S\right|<1$ *for all values of* $(S)_{i,j,\gamma}$ *with* $\gamma=\mathrm{E,W,S,N}$. *This condition is satisfied for instance if we take* $dt<2\frac{C}{S_\mathrm{L}+S_\mathrm{ir}}$.

Proof. We first have to prove that $\left\|[u]_h^n\right\|_2$ is bounded independently of n and h.

Because of Lemma 4.2, the matrix $\Lambda_\mathrm{c}\left(\mathrm{Id}+\frac{\sigma_\mathrm{c}}{\sigma_\mathrm{e}}\Lambda_\mathrm{e}^{-1}\Lambda_\mathrm{c}\right)^{-1}$ is symmetric and has only positive eigenvalues. As we have proved in Lemma 4.3 that $0\leqslant X\leqslant 1$ then $S>0$. Therefore $M=\left(\mathrm{Id}+\frac{dt}{C}\sigma_\mathrm{c}\Lambda_\mathrm{c}\left(\mathrm{Id}+\frac{\sigma_\mathrm{c}}{\sigma_\mathrm{e}}\Lambda_\mathrm{e}^{-1}\Lambda_\mathrm{c}\right)^{-1}\right)^{-1}$ is also symmetric and:

$$\|M\|_2=\rho(M)\leqslant 1.$$

Using (28) we write

$$\begin{aligned}\left\|[u]_h^{n+1}\right\|_2 &\leqslant \rho(M)\left\|[u]_h^n\right\|_2+dt\,\rho(M)\|F\|_2,\\ &\leqslant \left\|[u]_h^0\right\|_2+dt\sum_{i=0}^{n}\|F\|_2\leqslant\left\|[u]_h^0\right\|_2+T\|F\|_2.\end{aligned}$$

Therefore $\left\|[u]_h^n\right\|_2$ is bounded independently of n and h. In one-dimension the interface values are only two, therefore the boundedness in L^2 norm implies the boundedness in L^∞ norm.

Proposition 4.6. *If the ratio* dt/h *is kept constant, if* $dt<\tau_\mathrm{ep}$ *and* $\left|1-\frac{dt}{C}S\right|<1$ *for all values of* $(S)_{i,j,\gamma}$ *with* $\gamma=\mathrm{E,W,S,N}$, *then in one-dimension the numerical method converges with first-order accuracy to the exact solution.*

Proof. Let us recall the formula used to compute $[u]_h^{n+1}$ and X_h^{n+1} from $[u]_h^n$ and X_h^n:

$$[u]_h^{n+1}=M\left[\left(\mathrm{Id}-\frac{dt}{C}S(X_h^n)\right)[u]_h^n+dtF\right], \tag{34}$$

$$X_h^{n+1}=X_h^n+dt\,\max\left(\frac{\beta([u]_h^n)-X_h^n}{\tau_\mathrm{ep}},\frac{\beta([u]_h^n)-X_h^n}{\tau_\mathrm{res}}\right). \tag{35}$$

We know that the exact solution satisfies

$$[u]^{n+1}=M\left[\left(\mathrm{Id}-\frac{dt}{C}S(X^n)\right)[u]^n+dtF\right]+\varepsilon_1^n, \tag{36}$$

with ε_1^n related to the truncation error of the equation. If we assume that the ratio dt/h is kept constant, we know that $\varepsilon_1^n=O(dt^2)$ because the discretization in first-order accurate in time.

$$\begin{aligned}[u]^{n+1}-[u]_h^{n+1} &= M\left[\left(\mathrm{Id}-\frac{dt}{C}S(X^n)\right)[u]^n+dtF-\left(\mathrm{Id}-\frac{dt}{C}S(X_h^n)\right)[u]_h^n+dtF\right]+\varepsilon_1^n,\\ &= M([u]^n-[u]_h^n)-\frac{dt}{C}M\left(S(X^n)[u]^n-S(X_h^n)[u]_h^n\right)+\varepsilon_1^n,\\ \left\|[u]^{n+1}-[u]_h^{n+1}\right\|_2 &\leqslant \left\|[u]^n-[u]_h^n\right\|_2+\frac{dt}{C}(S_\mathrm{L}+S_\mathrm{ir})\left\|[u]^n-[u]_h^n\right\|_2\\ &\quad+\frac{dt}{C}\left\|[u]_h^n\right\|_\infty\left\|S(X^n)-S(X_h^n)\right\|_2+\left\|\varepsilon_1^n\right\|_2,\\ \left\|[u]^{n+1}-[u]_h^{n+1}\right\|_2 &\leqslant \left(1+\frac{dt}{C}(S_\mathrm{L}+S_\mathrm{ir})\right)\left\|[u]^n-[u]_h^n\right\|_2\\ &\quad+\frac{dt}{C}K(T)S_\mathrm{ir}\left\|[u]_h^n\right\|_\infty\sup_{i\leqslant n}\left\|[u]^n-[u]_h^n\right\|_2+\left\|\varepsilon_1^n\right\|_2+O(dt^2).\end{aligned}$$

$$\left\|[u]^{n+1}-[u]_h^{n+1}\right\|_2\leqslant\left(1+\frac{dt}{C}K(T)S_\mathrm{ir}\sup_{i\leqslant n}\left\|[u]_h^i\right\|_\infty+\frac{dt}{C}(S_\mathrm{L}+S_\mathrm{ir})\right)\sup_{i\leqslant n}\left\|[u]^i-[u]_h^i\right\|_2+\left\|\varepsilon_1^n\right\|_2+O(dt^2),$$

$$\sup_{i\leqslant n+1}\left\|[u]^i-[u]_h^i\right\|_2\leqslant\underbrace{\left(1+\frac{dt}{C}K(T)S_\mathrm{ir}\sup_{i\leqslant n}\left\|[u]_h^i\right\|_\infty+\frac{dt}{C}(S_\mathrm{L}+S_\mathrm{ir})\right)}_{1+dt\tilde{K}(T)}\sup_{i\leqslant n}\left\|[u]^i-[u]_h^i\right\|_2+\left\|\varepsilon_1^n\right\|_2+O(dt^2).$$

Table 1
Parameters set to fit to the results given by [19,5]. (EP stands for electropermeabilization, and EPd stands for electropermeabilized.)

Variable	Symbol	Value	Unit
Biological parameters:			
Extracellular conductivity	σ_e	5	S/m
Intracellular conductivity	σ_c	0.455	S/m
Capacitance	C	9.5×10^{-3}	F/m^2
Membrane surface conductivity	S_L	1.9	S/m^2
Cell radius	r	50	µm
Membrane thickness	δ	5	nm
Specific parameters of the model:			
EP threshold	V_{rev}	1,5	V
EP switch speed	k_{ep}	40	V^{-1}
EP characteristic time	τ_{ep}	1×10^{-6}	s
Resealing characteristic time	τ_{res}	1×10^{-3}	s
EPd membrane surface conductivity	S_{ir}	2.5×10^{8}	S/m^2

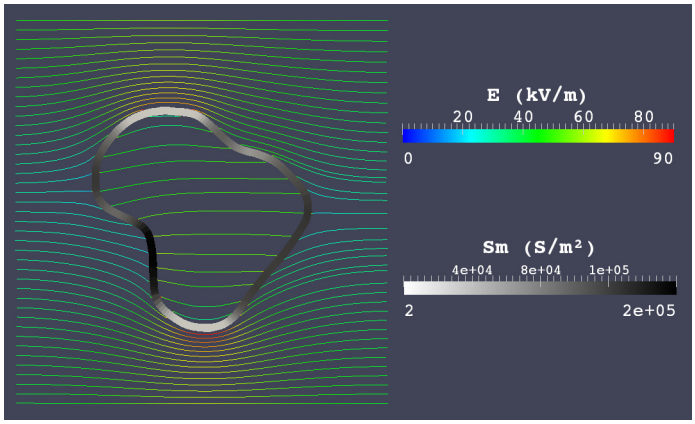


Fig. 8. Numerical electric field and permeabilization of a cell with an irregular shape. The electric field lines are visualized with colored lines while the membrane conductivity is depicted by a white to black scale and the thickness of the membrane. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

With a discrete Gronwall lemma, we deduce that

$$\sup_{i\leqslant n}\left\|[u]^i-[u]_h^i\right\|_2 \leqslant e^{T\tilde{K}(T)}\underbrace{\left\|[u]^0-[u]_h^0\right\|_2}_{=0}+\underbrace{\sum_{i=0}^{n}e^{(ndt-idt)\tilde{K}(T)}\left(\left\|\varepsilon_1^i\right\|_2+O(dt^2)\right)}_{=O(dt)}.$$

We conclude that the method converges with a first-order accuracy.

5. Numerical validations

In this section, we study numerically the convergence of the method in two dimensions, for both static and dynamic cases, with and without electropermeabilization (corresponding to non-linear and linear cases respectively). For an analysis of the "physical" behavior of the model we refer to the numerical studies in [10].

5.1. Parameters used in the simulations

Table 1 presents the parameters used for the following simulations. Some of them are biological parameters, chosen in accordance with reference studies in the literature, the other ones are specific to the model described in Subsection 1.2. Fig. 8 provides an example of results that are usually obtained in our simulations, for an irregularly shaped cell in this case. The electric field is visualized with lines while the membrane conductivity is depicted by a white to black scale and the thickness of the membrane. Note that this thickness is only a handful visualization artifact to locate the permeabilized regions of the membrane.

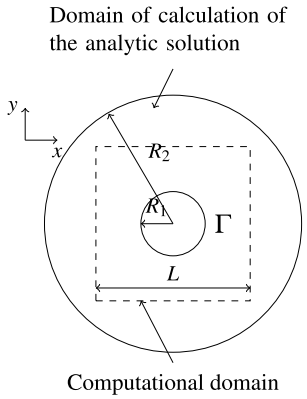


Fig. 9. Numerical estimation of the order of accuracy of the method. The analytic solution to the linear problem is calculated in concentric circular domains. The restriction of this solution to the boundary of the computational domain (dashed line) provides the Dirichlet data for the numerical solution.

5.2. Convergence study for the linear static problem

One can express an analytic solution of the linear problem in the case of a circular cell or radius R_1, centered on the origin, inside a concentric domain Ω of radius R_2, $R_2 > R_1$, with a constant conductivity $S = S_L$. Let $g = ER_2\cos\theta$ be the source with E tuning the amplitude of the electric field.

The analytic solution writes

$$\forall r > 0,\ \forall \theta \in [0, 2\pi],\ U(r,\theta) = \begin{cases} U_e(r)\cos\theta & \text{in } \mathscr{O}_e, \\ U_c(r)\cos\theta & \text{in } \mathscr{O}_c, \end{cases} \tag{37}$$

where

$$U_e(r) = \alpha_e r + \beta_e r^{-1},$$
$$U_c(r) = \alpha_c r,$$

with α_e, β_e, and α_c given by

$$\alpha_c = \left(\left(\frac{\sigma_c}{S_L R_1} + 1 + \frac{\sigma_c}{\sigma_e} \right) R_2 + \left(\frac{\sigma_c}{S_L R_1} + 1 - \frac{\sigma_c}{\sigma_e} \right) \frac{R_1^2}{R_2} \right)^{-1} g,$$
$$\alpha_e = \frac{1}{2} \left(\frac{\sigma_c}{S_L R_1} + 1 + \frac{\sigma_c}{\sigma_e} \right) \alpha_c,$$
$$\beta_e = \frac{1}{2} \left(\frac{\sigma_c}{S_L R_1} + 1 - \frac{\sigma_c}{\sigma_e} \right) \alpha_c R_1^2.$$

This solution is used as a Dirichlet condition on the boundary of our simulation box, which is a square contained in the disk of radius R_2, and containing the disk of radius R_1 (cf. Fig. 9).

The error between the result u of our simulation and the exact solution U is measured on both grid and interface points. In order to compute a discretized L^2 norm on the interface, the latter is considered linear between intersection points. If M_{m-1}, M_m and M_{m+1} designate three consecutive interface points, the length element Δl_m associated to the point M_m is chosen as:

$$\Delta l_m = \frac{1}{2} \left(\|M_{m+1} - M_m\| + \|M_m - M_{m-1}\| \right).$$

The error on the interface is defined by:

$$E_p(u - U) = \left(\sum_{m=0}^{N_l} \Delta l_m [u_m - U(x_m)]^2 \right)^{1/2}, \tag{38}$$

where N_l is the number of unknowns at interface.

For grid points, a special treatment is needed for points neighboring the interface for a second order L^2 norm calculation. Indeed, as shown in Fig. 10, considering the solution constant on a virtual cell of size $h_x \times h_y$ around each grid point is not valid, as these cells may intersect with both inner and outer domains. Let $\mathscr{A}_{ij}^c$ (resp. $\mathscr{A}_{ij}^e$) be the area of the part of the cell which is in $\mathscr{O}_c$ (resp. $\mathscr{O}_e$), which can be null if the cell is entirely in one domain. If the point M_{ij} is located in $\mathscr{O}_c$ (resp.

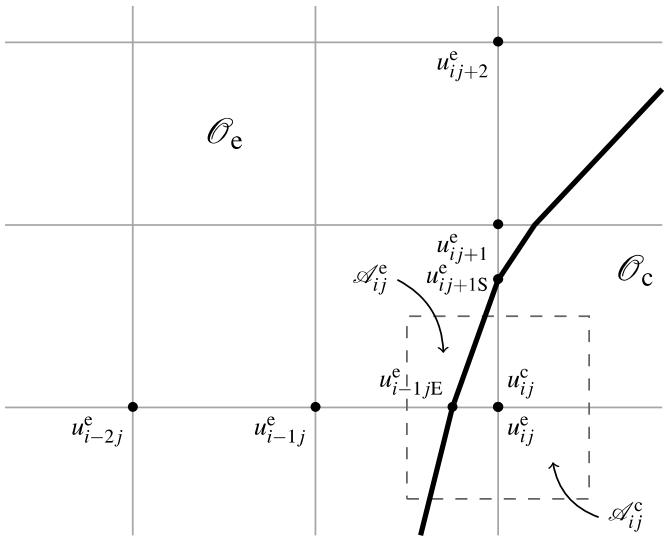


Fig. 10. L^2 norm calculation for grid points. u^{e}_{ij} is an extrapolation of the solution from the values of u^{e} in $\mathscr{O}_e$ to the point M_{ij} in $\mathscr{O}_c$, involving interface points as well.

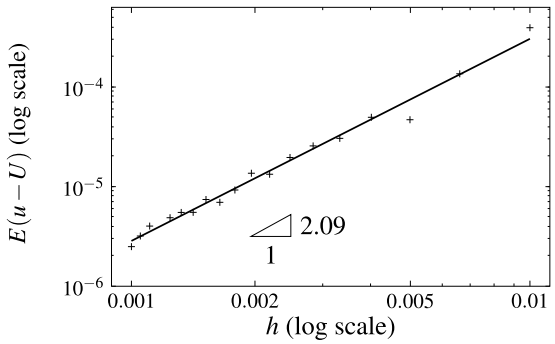


Fig. 11. Error vs relative grid spacing $h = 1/N_x = 1/N_y$ for the linear static case, with N_x and N_y being the number of points in the x and y directions.

$\mathscr{O}_e$), the solution in the exterior domain $\mathscr{O}_e$ (resp. $\mathscr{O}_c$), as well as the solution U, are extrapolated on M_{ij}, with respective values u^{e}_{ij} and U^{e}_{ij} (resp. u^{c}_{ij} and U^{c}_{ij}).

The error on grid points is then defined by:

$$E_g(u-U) = \left(\sum_{i,j} (u^{e}_{ij} - U^{e}_{ij})^2 \mathscr{A}^{e}_{ij} + (u^{c}_{ij} - U^{c}_{ij})^2 \mathscr{A}^{c}_{ij} \right)^{1/2}. \tag{39}$$

The total error reads:

$$E(u-U) = E_g(u-U) + E_p(u-U). \tag{40}$$

For the static problem, a second-order of convergence is achieved, as presented in Fig. 11.

5.3. Convergence study for the linear dynamic problem

5.3.1. Case of a circular cell

The solution of the dynamical problem has the same form as the solution (37) to the static equation, with time-dependent coefficients:

$$U_e(r,t) = \alpha_e(t) r + \beta_e(t) r^{-1},$$

$$U_c(r,t) = \alpha_c(t) r.$$

Then the jump on the interface $[U]$ satisfies the following ordinary differential equation (ODE):

$$C \frac{d}{dt}[U](t) + S_L[U](t) = -\sigma_c \alpha_c(t). \tag{41}$$

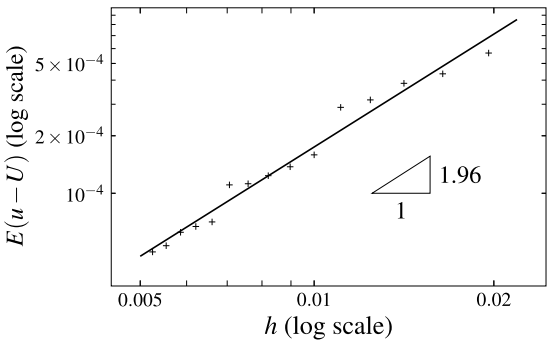


Fig. 12. Error vs relative grid spacing h for the linear dynamic case, with a circular cell. The error was computed after application of a 100 µs pulse. The ratio dx/dt was kept constant to 40.

Using the continuity of the flux (1d) we infer that:

$$\alpha_c(t) = K_1 g(t) + K_2[U](t),$$
$$\alpha_e(t) = \left(\frac{R_2}{R_1^2+R_2^2} + 2R_2\frac{\sigma_c}{\sigma_e}K\right) g(t) + \frac{\sigma_c}{\sigma_e}\frac{K}{R_1}[U](t),$$
$$\beta_e(t) = \left(\frac{R_2}{R_1^2+R_2^2} - 2R_2\frac{\sigma_c}{\sigma_e}K\right) R_1^2 g(t) - \frac{\sigma_c}{\sigma_e}\frac{K}{R_1}[U](t),$$

with

$$K = \frac{-\sigma_e}{R_1^2(\sigma_e-\sigma_c)+R_2^2(\sigma_e+\sigma_c)},\ K_1 = 2R_2K,\ K_2 = \frac{R_1^2+R_2^2}{R_1}K.$$

Therefore the jump is given by:

$$[U](t) = \left([U](0) + \frac{\sigma_c K_1}{C}\int_0^t g(s)\exp\left(\frac{S_L-\sigma_c K_2}{C}s\right)\mathrm{d}s\right)\exp\left(-\frac{S_L-\sigma_c K_2}{C}t\right),$$

from which we infer all the coefficients.

The error is computed at the final time of the simulation $T_f = 100$ µs, similarly as for the static equation. Fig. 12 shows that a second-order of convergence is obtained for this case as well.

5.3.2. Case of more complex geometries

Here we study the empirical convergence in the case of more complex geometries:

- an oblong interface, but with sharp corners, defined as the intersection of the two circles

$$C_1 = \{(x,y),\ \sqrt{x^2+(y+50\,\mu\mathrm{m})^2} = 75\,\mu\mathrm{m}\}\quad C_2 = \{(x,y),\ \sqrt{x^2+(y-50\,\mu\mathrm{m})^2} = 75\,\mu\mathrm{m}\}$$

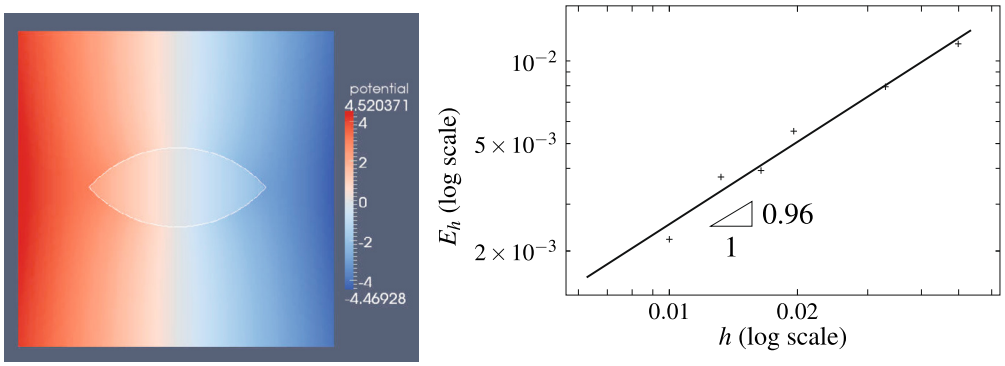
  in the computational domain $[-100\ \mu\mathrm{m}, 100\ \mu\mathrm{m}]^2$.
- a flower-like cell interface defined by:

$$\varphi(r,\theta) = r - r_0 - 0.2\sin(\omega\theta),$$

  with $r = \sqrt{(x-x_c)^2+(y-y_c)^2}$, $\theta = arctan((y-y_c)/(x-x_c))$, $x_c = y_c = 0.2\sqrt{20}\times 50$ µm, $\omega = 5$, $r_0 = 0.8\times 50$ µm.

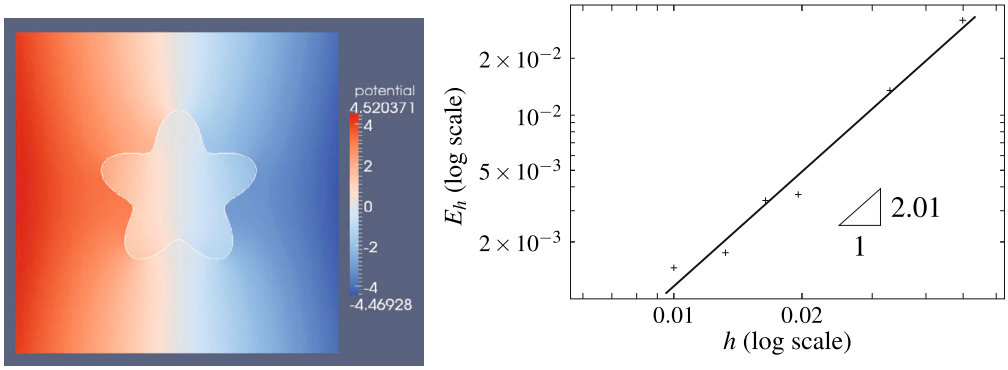
In both cases, the empirical convergence rate is computed by comparing a refined numerical solution, obtained with 301×301 grid points, to numerical solutions obtained with a number of grid points ranging from 21×21 to 101×101. The ratio $\frac{dt}{dx}$ is kept constant, equal to τ_{ep}. The final time of the simulation is $T_f = \frac{\tau_{ep}}{2}$. E_h is only computed on grid points, since interface points may not match on both grids. The numerical solution with 201×201 grid points and the empirical convergence rate are presented on Fig. 13 for the ellipsoidal like interface, and on Fig. 14 for the flower-like interface.

We observe a second order convergence for the flower-like interface, and a first-order convergence for the interface with sharp corners. The first-order convergence in the latter case is coherent with the fact that a first-order discretization in time is used. However, since it is not observed for the other geometries, it can also be correlated with the first-order discretization used at the corners. This is an effect not observed in the test-cases with sharp corners in [3], where a second-order convergence is always observed, but the loss of accuracy of the first-order discretization may be amplified here by the numerous time iterations. In the case of cell interfaces, such sharp corners are not likely to be observed.



(a) (b)

Fig. 13. (a): Numerical solution for 201×201 grid points, (b): Error E_h versus grid spacing h.



(a) (b)

Fig. 14. (a): Numerical solution for 201×201 grid points, (b): Error E_h versus grid spacing h.

5.4. Convergence study with electropermeabilization

When electropermeabilization is taken into account, *i.e.* when the conductivity S depends on the TMP difference $[U]$, we are not able to compute an analytic solution. In order to compute the order of convergence p of our method, a solution u_h computed with a grid spacing h is compared to $u_{h/2}$, computed with a refined grid spacing $h/2$. The error E_h is then of the order of h^p, with:

$$E_h = \frac{\|u_h - u_{h/2}\|_{L^2}}{\|u_{h/2}\|_{L^2}} \leqslant \frac{\|u_h - \bar{u}\|_{L^2}}{\|u_{h/2}\|_{L^2}} + \frac{\|u_{h/2} - \bar{u}\|_{L^2}}{\|u_{h/2}\|_{L^2}} = \mathcal{O}(h^p) + \mathcal{O}\left(\frac{h^p}{2^p}\right).$$

E_h is only computed on grid points, since interface points may not match on both grids.

On Fig. 15 are presented the convergence results in the static and dynamic case. In the static case, the numerical solution is obtained with a pseudo-time iterative scheme described in [10]. In the dynamic case, we perform two studies, one with $dt = \tau_{ep} \times dx$, and another one with $dt = 10 \times \tau_{ep} \times dx^2$. We take $k_{ep} = 10$ rather than 40, because the electropermeabilization happens early with the choice $k_{ep} = 10$, helping us to perform the convergence study in time. E_h is measured at the final time $T_f = \frac{\tau_{ep}}{2}$ in the first case, and at the time $T_f = \frac{\tau_{ep}}{10}$ in the second case, to avoid too long computations due to the small values of dt. In the second case, E_h is computed in maximum norm to show that the computed almost second-order convergence is satisfied even near the interface, which would be not so evident if it was measured in L^2 norm. In the first case, we observe an almost first-order convergence, which is coherent with the fact that the temporal integration scheme is only order one. In the second case, the computed rate of convergence is slightly inferior to second-order, which agrees with the fact that we use a dt proportional to dx^2.

6. Conclusion

In this paper we have presented a new finite differences method to simulate electropermeabilization models based on elliptic PDE's describing the evolution of the electric potential in conducting media. It is a Cartesian grid method based on the accurate discretization of the fluxes at the interface between the exterior domain and the cell, representing the

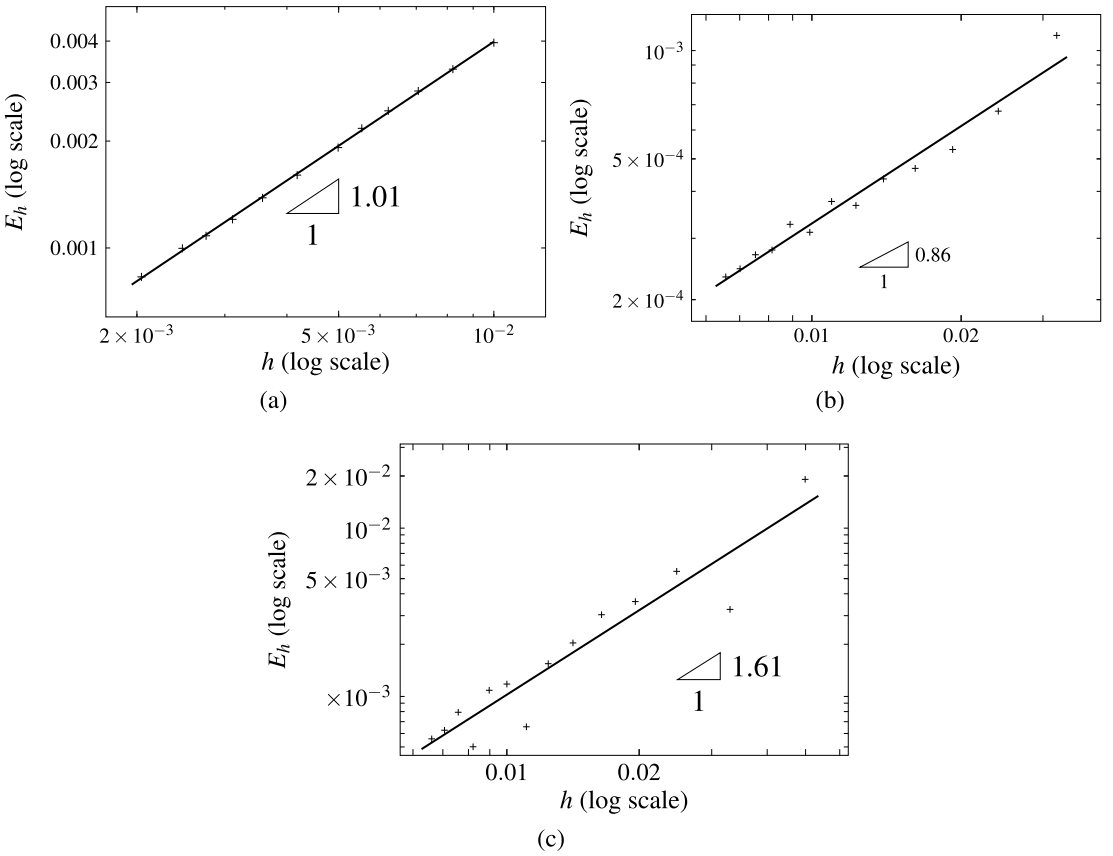


Fig. 15. Error E_h versus grid spacing h, in both static (a) and dynamic case (b): $dt = \tau_{ep} \times dx$ and (c) $dt = 10 \times \tau_{ep} \times dx^2$.

cell membrane. This accurate discretization is performed with additional interface unknowns, located at the intersection between the interface and the grid axes. The use of interface unknowns is particularly relevant in the context of electropermeabilization, because the jump in electric potential across the cell membrane is explicitly given in the models and is measured by experiments.

We have studied the stability and convergence of the method in one dimension, and proved that in one dimension, it converges with a first-order accuracy for the dynamic non-linear problem, and with a second-order accuracy for the linear stationary problem. The first order-accuracy in the non-linear case is due to the fact that a first-order scheme is employed for the time evolution. Finally we have presented numerical results in two dimensions corroborating the theoretical study.

In the future, we intend to use this method to perform simulations of the electropermeabilization of high numbers of cells in the same computational domain. To this purpose we will develop a parallelized version of the method, with the help of the PETSc library. Furthermore, we plan to modify the method to take into account a possible variation of the volume of the electropermeabilized cells, which is highlighted by experiments. This will be handled by computing the temporal evolution of the level-set function, following the equation of evolution for the interface that will be added to the model.

Acknowledgements

This study has been carried out in the frame of the Investments for the future Programme IdEx Bordeaux CPU (ANR-10-IDEX-03-02). The authors have been partly granted by the French national agency throughout the research projects INTCELL (2010-BLAN-916-04) and MEMOVE (2011-BS01-006-01). Numerical experiments presented in this paper were carried out using the PLAFRIM experimental testbed, being developed under the Inria PlaFRIM development action with support from LABRI and IMB and other entities: Conseil Régional d'Aquitaine, FeDER, Université de Bordeaux and CNRS (see https://plafrim.bordeaux.inria.fr/). This work was partly performed in the scope of the European Associated Laboratory EBAM.

References

[1] J.T. Beale, A.T. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, Commun. Appl. Math. Comput. Sci. 1 (2006) 91–119.

[2] P. Bethelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (2004) 364–386.
[3] M. Cisternino, L. Weynans, A parallel second order cartesian method for elliptic interface problems, Commun. Comput. Phys. 12 (2012) 1562–1587.
[4] A. Coco, G. Russo, Finite-difference ghost-point multigrid methods on Cartesian grids for elliptic problems in arbitrary domaines, J. Comput. Phys. 241 (2013) 464–501.
[5] K. DeBruin, W. Krassowska, Modelling electroporation in a single cell. I. Effects of field strength and rest potential, Biophys. J. 77 (Sept. 1999) 1213–1224.
[6] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[7] F. Gibou, R.P. Fedkiw, L.T. Cheng, M. Kang, A second order accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.
[8] Grégory Guyomarc'h, Chang-Ock Lee, Kiwan Jeon, A discontinuous Galerkin method for elliptic interface problems with application to electroporation, Commun. Numer. Methods Eng. 25 (10) (2009) 991–1008.
[9] H. Huang, Z. Li, Convergence analysis of the immersed interface method, IMA J. Numer. Anal. 19 (1999) 583–608.
[10] Otared Kavian, Michael Leguèbe, Clair Poignard, Lisl Weynans, "Classical" electropermeabilization modeling at the cell scale, J. Math. Biol. (2012) 1–31.
[11] M. Latige, T. Colin, G. Gallice, A second order cartesian finite volume method for elliptic interface and embedded Dirichlet problems, Comput. Fluids 83 (2013) 70–76.
[12] R.J. Leveque, L.Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044.
[13] Z.L. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1998) 230–254.
[14] Z.L. Li, K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM J. Sci. Comput. 23 (2001) 339–361.
[15] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.
[16] A. Mayo, The fast solution of Poisson's and the biharmonic equations on general regions, SIAM J. Numer. Anal. 21 (1984) 285–299.
[17] A. Mayo, The rapid evaluation of volume integrals of potential theory on general regions, J. Comput. Phys. 100 (1992) 236–245.
[18] A. Mayo, A. Greenbaum, Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Sci. Stat. Comput. 13 (1992) 101–118.
[19] J. Neu, W. Krassowska, Asymptotic model of electroporation, Phys. Rev. E 53 (3) (Mar 1999) 3471–3482.
[20] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.
[21] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (12) (1988).
[22] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, UK, 1999.
[23] J.A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, J. Comput. Phys. 169 (2001) 503–555.
[24] A. Wiegmann, K. Bube, The explicit jump immersed interface method: finite difference method for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (3) (2000) 827–862.
[25] Y.C. Zhou, S. Zhao, M. Feig, G.W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (2006) 1–30.