A numerical strategy to discretize and solve Poisson equation on dynamically adapted multiresolution grids for time-dependent streamer discharge simulation

Max Duarte, Zdenek Bonaventura, Marc Massot, Anne Bourdon

A numerical strategy to discretize and solve Poisson equation on dynamically adapted multiresolution grids for time-dependent streamer discharge simulation

Max Duarte[†‡] Zdeněk Bonaventura[§] Marc Massot[¶‖**] Anne Bourdon[¶‖]

February 22, 2014

Abstract

We investigate the numerical solution of Poisson equations on dynamically adapted structured grids generated by multiresolution analysis, arising in the numerical simulation of time-dependent problems modeled by evolutionary partial differential equations (PDEs) disclosing propagating fronts such as streamer discharge phenomena. A multiresolution technique not only involves important savings in computational costs, but we also prove that it guarantees numerical approximations to the Poisson equation within an accuracy tolerance. In contrast to most adaptive meshing approaches in the literature that solve such elliptic PDEs level-wise and thus at uniform resolution throughout the set of adapted grids, we introduce a numerical procedure mainly based on inter-level operations to represent the elliptic operators on the adapted grid. In this way the discrete Poisson equation can be solved at once over the entire computational domain as a completely separate process, strongly coupling inter-grid relations and independently of the mesh generation or any other grid-related data structure or geometric consideration. We investigate the validity of both the theoretical characterization as well as the numerical construction of the discrete operators in the context of streamer discharge simulations. Multiresolution error bounds are thus evaluated in a simpler configuration with analytical solution that nevertheless mimics the spatial structure found in such physical configurations. The complete model is then considered to simulate double-headed streamers with photoionization mechanisms, for which we study the performance and capabilities of various direct and iterative linear solvers. The numerical strategy is finally assessed by showing its ability to accurately describe the complex interaction dynamics of two positive streamer discharges, a problem especially relevant for applications.

Keywords

Poisson equation, finite volume multiresolution, error bounds, streamer discharges.

AMS subject classifications

35J05, 65M15, 65M50, 65G20, 65M08, 65Y15, 65Z05, 76X05.

[†]Univ. Nice Sophia Antipolis, CNRS, LJAD, UMR 7351, 06100 Nice, France.

[‡]CCSE, Lawrence Berkeley National Laboratory, 1 Cyclotron Rd. MS 50A-1148, 94720 Berkeley, CA, USA (MDGonzalez@lbl.gov).

[§]Department of Physical Electronics, Faculty of Science, Masaryk University, Kotlářská 2, 61137 Brno, Czech Republic (zbona@physics.muni.cz).

[¶]CNRS, UPR 288, Laboratoire EM2C, Grande voie des vignes, 92295 Châtenay-Malabry, France.

[‖]Ecole Centrale Paris, 92295 Châtenay-Malabry, France ({marc.massot,anne.bourdon}@ecp.fr).

[**]Fédération de Mathématiques de l'Ecole Centrale Paris, FR CNRS 3487.

1 Introduction

In numerous scientific applications we have to deal with the numerical solution of elliptic PDEs such as Poisson equation coupled with evolutionary PDEs in order to address the numerical simulation of time-dependent physical processes. One major example is given, for instance, by the so-called projection methods [13, 50], widely investigated, extended, and implemented in the literature to solve the incompressible Navier-Stokes equations (see, *e.g.*, [25] and references therein). Solving Poisson equations is also very common in plasma physics simulations. As an example, in the framework of a drift-diffusion model consisting of a set of continuity equations for charged species coupled with a Poisson equation for the electric potential, non-linear ionization waves also called streamers can be simulated. These ionization waves occur as a consequence of the high electric field induced by the fast variations of the net charge density ahead of an electron avalanche with large amplification [46]. Streamers are at the basis of the filamentary structure of discharges at atmospheric pressure [4, 31] and are of interest for many low-temperature plasma applications. In either situation Poisson-type equations must be solved often several times at every time-step throughout the numerical simulation, a task that depending on the size and complexity of the problem can easily become cumbersome in both CPU time and memory. In particular phenomena characterized by propagating fronts, as considered in this work, commonly require a sufficiently fine spatial representation and potentially large systems of equations need then to be solved. Therefore, even though many dedicated software packages are available in the literature to solve this kind of problem, supplementary work proves to be necessary to ease the corresponding computational efforts.

In this regard, once the size of the finest resolved spatial scale is chosen, grid adaptation for time-dependent problems disclosing localized fronts is specifically designed to yield high data compression and hence important savings in computational costs (see, *e.g.*, [7, 6]). The context of this work is thus established by the numerical solution of elliptic PDEs of Poisson-type on time-varying adapted grids, resulting from the use of adaptive gridding techniques for time-dependent problems modeled by stiff PDEs. Among the many adaptive meshing approaches developed in the literature, we consider in this work adaptive multiresolution schemes based on [26, 27], namely the multiresolution finite volume scheme introduced in [17] for conservation laws. Besides the inherent advantages of grid adaptation, multiresolution techniques rely on biorthogonal wavelet decomposition [16] and thus offer a rigorous mathematical framework for adaptive meshing schemes [14, 39]. In this way not only approximation errors coming from grid adaptation and thus data compression can be tracked, but general and robust techniques can be built since the wavelet decomposition is independent of any physical particularity of the problem and accounts only for the spatial regularity of the discrete variables at a given simulation time. Adaptive multiresolution schemes have been successfully implemented for the simulation of compressible fluids modeled by Euler or Navier-Stokes equations (see, *e.g.*, [40, 11, 19] and references therein), as well as for the numerical solution of time-dependent parabolic [47, 12] and stiff parabolic PDEs [23, 22].

Having a set of equations discretized on an adapted structured grid a critical aspect throughout the practical implementation is related to the way of working with non-uniform discretizations, especially for elliptic operators that act simultaneously on the whole domain. In the context of embedded structured grids with a given number of grid-levels a particular attention must be addressed to the inter-grid interfaces in order to consistently define the discrete operations there. Otherwise potential mismatches may lead to substantial differences in the numerical approximations as well as loss of conservation (see [1] for a detailed discussion). The most common way of solving an elliptic PDE on this type of adapted grid consists in solving the discrete system level-wise, that is, considering one grid-level at a time followed by inter-level operations to synchronize shared interfaces at different grid-levels as well as overlapped regions. In this way the main idea is to perform successive computations over partial regions of the whole domain at a uniform mesh resolution until the problem is entirely solved on the adapted grid. Some examples can be found, for instance, in [1, 51, 37, 36, 49]. Such a level-wise approach requires then a sufficiently accurate resolution

at coarser levels since these solutions will be successively used for finer grid computations, for instance, to define the boundary conditions of finer regions inside the computational domain. For intensive computations iterative linear solvers based on geometric multigrid schemes are usually implemented, taking advantage of the multi-mesh representation of the problem [1, 51, 36].

In this paper we consider the numerical solution of Poisson equations on adapted grids generated by multiresolution analysis. Based on the mathematical framework of wavelet decomposition we investigate the influence of data compression on the accuracy of the approximations obtained from the Poisson equations discretized on the adapted mesh, in the specific case of a multiresolution finite volume scheme for time-dependent PDEs[1]. Instead of solving the discrete equations level-wise throughout the set of embedded grids, we have conceived a numerical procedure to represent the elliptic operators discretized directly on the adapted grid consisting of cells with different spatial resolution. The algorithm relies on a local reconstruction of uniform-grid zones at inter-level interfaces by means of multiresolution operations between consecutive grid-levels that guarantee the conservation and accuracy properties of multiresolution schemes. This is a general approach that not only yields a compact representation of the problem thanks to grid adaptation, but also results in a separate algebraic system completely independent of any consideration related to the adaptive meshing scheme or its corresponding data structure, as well as of the numerical integration of the stiff time-dependent PDEs associated with the model. The resulting discrete systems can thus be solved at once over the whole computational domain with no need of grid overlapping by considering an appropriate linear solver.

In order to investigate the theoretical insights we have derived in the framework of multiresolution approximations and to validate the numerical construction of the discrete operators, we carry out some numerical computations in the context of streamer discharge simulations at atmospheric pressure. The detailed physics of these discharges reveals an important time-space multi-scale character [24]. Specifically a large variation of space scales needs to be taken into account since the Debye length at atmospheric pressure can be as small as a few micrometers, while the inter-electrode gaps where discharges propagate are usually of the order of a few centimeters. Grid adaptation is therefore highly desirable and was already considered for streamer simulations, for instance, in [37, 45, 52], where in particular the elliptic operator was directly discretized on the non-uniform grid in [52] in a different context for an asynchronous time integration scheme. In [21] we introduced a time-space adaptive numerical scheme to simulate propagating streamers on adaptive multiresolution grids with error control. Nevertheless, a simplified geometry was considered there in order to avoid the numerical solution of the Poisson equation. The present work describes then the required fundamentals and further developments needed to consider and solve the Poisson equations on the adapted grids according to the approach established in [21], which aims at assuring a tracking capability of the numerical errors and a full resolution of the equations on the adapted grid without the need of grid overlapping.

We consider first a simplified model with analytical solution that retains the spatial characteristics of the complete model in order to assess the multiresolution error bounds. A more realistic model of double-headed streamers with photoionization mechanisms is then studied, for which we conduct a study on the numerical performance of some software packages available in the literature to solve linear systems paying a particular attention to algebraic multigrid solvers. Finally, we show how the numerical strategy developed in this work ensures a fine time-space resolution of a particularly relevant configuration in the field of plasma physics given by the complex interaction of two positive streamers, the physics of which has been recently investigated in [8]. Even though the numerical strategy remains perfectly valid for three-dimensional configurations, only two-dimensional problems are simulated and described here which are complex enough to thoroughly evaluate the numerical technique. In this way this paper aims at describing different aspects related to the

[1]In this paper we do not investigate the convergence rates of adaptive multiresolution techniques, but rather consider a fixed finest grid for which the finest physical scales are known to be resolved and study the influence of data compression. A detailed theoretical study on more general adaptive wavelet methods for elliptic equations can be found in [15].

numerical solution of Poisson equations on multiresolution grids for time-dependent problems, from its mathematical description to its effective implementation along with additional theoretical and practical considerations.

The paper is organized as follows. We present some theoretical principles of multiresolution analysis in Section 2 before conducting a mathematical description of Poisson equations discretized on multiresolution grids. In Section 3 we recall some key aspects of the multiresolution technique considered here and then we describe the numerical procedure conceived to represent elliptic operators on the adapted mesh. Numerical results coming from streamer discharge simulations are investigated in Section 4.

2 Data compression errors for Poisson equation on multiresolution grids

We carry out a mathematical description of a Poisson equation discretized on a multiresolution adapted grid and in particular the influence of data compression on the numerical accuracy of the approximations. However, we first need to briefly recall the general framework of biorthogonal wavelet bases and multiresolution analysis. More details on wavelet decomposition and multiresolution techniques for grid adaptation can be found in [14, 39].

2.1 Biorthogonal wavelet decomposition

Let us consider an open domain $\Omega \subset \mathbb{R}^d$, and two sequences of nested subspaces of $L^2(\Omega)$:

$$V_j \subset V_{j+1}, \quad \widetilde{V}_j \subset \widetilde{V}_{j+1}, \quad j \in \mathbb{N}_0, \tag{1}$$

such that $\overline{\bigcup_{j\in\mathbb{N}_0} V_j} = \overline{\bigcup_{j\in\mathbb{N}_0} \widetilde{V}_j} = L^2(\Omega)$. For each pair of subspaces we introduce the sets of scaling functions: $(\phi_{j,k})_{k\in\Delta_j}$ and $(\widetilde{\phi}_{j,k})_{k\in\Delta_j}$, $\Delta_j \subset \mathbb{Z}^d$, with compact supports of diameter proportional to 2^{-j}, and with the biorthogonal property:

$$\langle \phi_{j,k}, \widetilde{\phi}_{j,k'} \rangle = \delta_{k,k'}, \quad k, k' \in \Delta_j,\ j \in \mathbb{N}_0, \tag{2}$$

where the second subscript k indicates the localization $2^{-j}k$ of $\phi_{j,k}$ (resp., $\widetilde{\phi}_{j,k}$) within the subspace V_j (resp., $\widetilde{V}_j$). For a given $f \in L^2(\Omega)$, we can then define the following projector P_j onto $V_j \subset L^2(\Omega)$:

$$P_j f := \sum_{k\in\Delta_j} \langle f, \widetilde{\phi}_{j,k} \rangle \phi_{j,k}. \tag{3}$$

It can thus be seen that the base $(\phi_{j,k})_{k\in\Delta_j}$, together with (3), defines an approximation space V_j of resolution 2^{-j} in $L^2(\Omega)$. The same follows for $(\widetilde{\phi}_{j,k})_{k\in\Delta_j}$ and $\widetilde{V}_j$, considering the adjoint projector $\widetilde{P}_j$ onto $\widetilde{V}_j$.

Based on the definition of $(\phi_{j,k}, \widetilde{\phi}_{j,k})$, we can construct a pair of biorthogonal wavelets: $\psi_{j,k}$ and $\widetilde{\psi}_{j,k}$, with the pairwise biorthogonality property:

$$\langle \psi_{j,k}, \widetilde{\psi}_{j',k'} \rangle = \delta_{(j,k),(j',k')}, \quad k, k' \in \Delta_j,\ j, j' \in \mathbb{N}_0,$$

such that

$$\langle \phi_{j,k}, \widetilde{\psi}_{j,k'} \rangle = \langle \widetilde{\phi}_{j,k}, \psi_{j,k'} \rangle = 0. \tag{4}$$

The pairs $(\phi_{j,k}, \psi_{j,k})$ and $(\widetilde{\phi}_{j,k}, \widetilde{\psi}_{j,k})$ are usually known as, respectively, the *primal* and the *dual* scaling function and wavelet. The nestedness of the approximation spaces (1) involves that

$$P_j P_{j+1} = P_j, \quad \widetilde{P}_j \widetilde{P}_{j+1} = \widetilde{P}_j,$$

and we can thus define the projection

$$Q_j f = (P_{j+1} - P_j) f = \sum_{k \in \Delta_j} \langle f, \widetilde{\psi}_{j,k} \rangle \psi_{j,k}, \tag{5}$$

onto the complement space $W_j = V_{j+1} \cap \widetilde{V}_j^{\perp}$, known as the wavelet space, spanned by the base $(\psi_{j,k})_{k \in \Delta_j}$. Projector $\widetilde{Q}_j := \widetilde{P}_{j+1} - \widetilde{P}_j$ onto $\widetilde{W}_j = \widetilde{V}_{j+1} \cap V_j^{\perp}$ is similarly defined.

The projection $P_j f$ defined in (3) resolves function f up to the scale 2^{-j}, while finer details are discarded. For two successive spaces: V_{j+1} and V_j, the representation at scale $2^{-(j+1)}$: $P_{j+1} f$, can be thus reconstructed from a coarser one at 2^{-j}: $P_j f$, by adding according to (5) the omitted information when going from a coarse to a finer scale: $Q_j f$. Iterating from a given $J > 0$ induces then the following multi-scale representation:

$$P_J f = P_{J-1} f + [P_J f - P_{J-1} f] = \ldots = P_0 f + \sum_{j=0}^{J-1} Q_j f, \tag{6}$$

which similarly leads to the following wavelet decomposition of $L^2(\Omega)$:

$$f = P_0 f + \sum_{j=0}^{\infty} Q_j f. \tag{7}$$

2.2 Multiresolution analysis

Following the multiresolution finite volume scheme of [17], let us build a set of nested dyadic grids over $\Omega \subset \mathbb{R}^d$. With the abbreviated notation: $\Omega_\gamma := \Omega_{j,k}$, we consider regular disjoint partitions $(\Omega_\gamma)_{\gamma \in S_j}$ of Ω such that $\overline{\bigcup_{\gamma \in S_j} \Omega_\gamma} = \Omega$, for $j = 0, 1, \ldots, J$. Since each Ω_γ, $\gamma \in S_j$, is in general the union of a finite number of cells Ω_μ, $\mu \in S_{j+1}$, (2^d cells in the dyadic case) the sets S_j and S_{j+1} represent consecutive embedded grids over Ω. We thus denote $|\gamma| := j$ if $\gamma \in S_j$, while subscript $k \in \Delta_j$ corresponds to the position of the cell within S_j. For instance, in Cartesian coordinates we consider the univariate dyadic intervals in $\mathbb{R}$:

$$\Omega_\gamma = \Omega_{j,k} :=]2^{-j}k, 2^{-j}(k+1)[, \ \gamma \in S_j := \{(j,k) \text{ s.t. } j \in (0, 1, \ldots, J),\ k \in \mathbb{Z}\},$$

and the same follows for higher dimensions.

Defining the dual scaling function $\widetilde{\phi}_\gamma$ in (3) as

$$\widetilde{\phi}_\gamma := |\Omega_\gamma|^{-1} \chi_{\Omega_\gamma}, \tag{8}$$

we denote $\mathbf{f}_j := (f_\gamma)_{\gamma \in S_j}$, with $f_\gamma := \langle f, \widetilde{\phi}_\gamma \rangle$, as the spatial representation of $f(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$, on the grid S_j. Notice that f_γ is no other than the cell-average of $f : \Omega \to L^2(\Omega)$ in Ω_γ, scaled by $|\Omega_\gamma|$; $\mathbf{f}_j$ is hence equivalent to a finite volume representation of resolution 2^{-j} of f. For consecutive grid-levels the following representations of $f_{j+1} := P_{j+1} f \in V_{j+1}$ are thus perfectly equivalent,

$$f_{j+1} = \sum_{|\mu| = j+1} f_\mu \phi_\mu = \sum_{|\gamma| = j} f_\gamma \phi_\gamma + \sum_{|\gamma| = j} d_\gamma \psi_\gamma, \tag{9}$$

following (5), where the set of *details*: $(d_\gamma)_{\gamma \in S_j}$, $d_\gamma := \langle f, \widetilde{\psi}_\gamma \rangle$, gathers the wavelet coefficients in the wavelet space W_j.

Now, for a given scaling function $\widetilde{\phi}_\lambda$ at level j: $|\lambda| = j$, applying $\langle \cdot, \widetilde{\phi}_\lambda \rangle$ term-wise to (9) and considering the biorthogonality properties (2) and (4) yield

$$\sum_{|\mu| = |\lambda|+1} f_\mu \langle \phi_\mu, \widetilde{\phi}_\lambda \rangle = f_\lambda; \tag{10}$$

and we can hence define a fine-to-coarse operation given by

$$f_\lambda = |\Omega_\lambda|^{-1} \sum_{|\mu|=|\lambda|+1, \Omega_\mu \subset \Omega_\lambda} |\Omega_\mu| f_\mu, \tag{11}$$

as an immediate consequence of (8) into (10). Similarly, by applying $\langle \cdot, \widetilde{\phi}_\lambda \rangle$ term-wise to (9), this time with $|\lambda| = j+1$, a coarse-to-fine operation can be defined by evaluating

$$f_\lambda = \sum_{\Sigma_\gamma \cap \Sigma_\lambda \neq \varnothing} f_\gamma \langle \phi_\gamma, \widetilde{\phi}_\lambda \rangle + \sum_{\Sigma_\gamma \cap \Sigma_\lambda \neq \varnothing} d_\gamma \langle \psi_\gamma, \widetilde{\phi}_\lambda \rangle, \quad |\lambda| = |\gamma| + 1, \tag{12}$$

where $\Sigma_\gamma := \mathrm{supp} \widetilde{\psi}_\gamma$. Data between consecutive levels are thus uniquely related by inter-grid operations, (11) and (12), where the coefficients, known as *masks*, do not depend on data but only on the definition of the compactly supported wavelets and scaling functions [14, 39].

Following [17], we can equivalently construct the details as

$$d_\mu = \langle f, \widetilde{\psi}_\mu \rangle = f_\mu - \widehat{f}_\mu, \tag{13}$$

with

$$\widehat{f}_\mu := \sum_{\Sigma_\gamma \cap \Sigma_\mu \neq \varnothing} f_\gamma \langle \phi_\gamma, \widetilde{\phi}_\mu \rangle = \sum_{\gamma \in R_I(\mu)} \beta_{\mu,\gamma} f_\gamma, \quad |\mu| = |\gamma| + 1, \tag{14}$$

for a set of coefficients $(\beta_{\mu,\gamma})_{\gamma \in R_I(\mu)}$ and $R_I(\mu) := \{\gamma \text{ s.t.} (|\gamma| = |\mu| - 1 \wedge \Sigma_\gamma \cap \Sigma_\mu \neq \varnothing)\}$, such that

$$f_\gamma = |\Omega_\gamma|^{-1} \sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| \widehat{f}_\mu, \tag{15}$$

as a result of applying the fine-to-coarse operation (11) to (12). The definition of (13)-(14) amounts to consider a box wavelet $\widetilde{\psi}_\mu$ of order β resulting from the combination of B-splines of order 1 defined by the dual scaling function (8):

$$\widetilde{\psi}_\mu := \widetilde{\phi}_\mu - \sum_{\gamma \in R_I(\mu)} \beta_{\mu,\gamma} \widetilde{\phi}_\gamma,$$

after introducing (8) and (14) into (13). Hence, $\widehat{f}_\mu$ is an approximation of order β to f_μ in (14) based on the values f_γ, $|\gamma| = |\mu| - 1$, contained in a stencil $R_I(\mu)$ around Ω_μ.

From (9), a one-to-one correspondence is obtained between $\mathbf{f}_{j+1}$ and $\mathbf{f}_j$ by additionally considering the wavelet coefficients $(d_\gamma)_{\gamma \in S_j}$, and the same holds for the multi-scale representation (6) that considers $\mathbf{f}_0$ and the sets $(d_\gamma)_{\gamma \in S_{j'}}$, $j' = 0, \ldots, j$, to obtain $\mathbf{f}_{j+1}$. However, based on the construction (13) in [17] we consider instead the array $\mathbf{d}_{j+1} = (d_\mu)_{\mu \in \nabla_{j+1}}$, where the set $\nabla_{j+1} \subset S_{j+1}$ is obtained by removing for each $\gamma \in S_j$ one $\mu \in S_{j+1}$ such that $\Omega_\mu \subset \Omega_\gamma$. In this way we keep the one-to-one correspondence taking into account that the details involve the additional relation: $\sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} d_\mu = 0$, according to (13) and (15). We can thus write that

$$\mathbf{f}_{j+1} \longleftrightarrow (\mathbf{f}_j, \mathbf{d}_{j+1}),$$

and by iterating this decomposition, we finally obtain a multi-scale representation of $\mathbf{f}_J$ in terms of $\mathbf{m}_J := (\mathbf{f}_0, \mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_J)$:

$$\mathcal{M} : \mathbf{f}_J \longmapsto \mathbf{m}_J, \tag{16}$$

and similarly, its inverse $\mathcal{M}^{-1}$.

Given a set of indices $\Lambda \subset \nabla^J$, where $\nabla^J := \bigcup_{j=0}^J \nabla_j$ with $\nabla_0 := S_0$, we define a *thresholding* operator $\mathcal{T}_\Lambda$ that leaves unchanged the components d_λ of $\mathbf{m}_J$ if $\lambda \in \Lambda$, and replaces it by 0 otherwise. Defining the level-dependent threshold values $(\epsilon_0, \epsilon_1, \ldots, \epsilon_J)$, the set Λ is given by

$$\lambda \in \Lambda \text{ if } \|d_\lambda\|_{L^p} \geq \epsilon_{|\lambda|}. \tag{17}$$

Applying $\mathcal{T}_\Lambda$ on the multi-scale decomposition $\mathbf{m}_J$ of $\mathbf{f}_J$ amounts then to building the multiresolution approximation $\mathcal{A}_\Lambda \mathbf{f}_J$ to $\mathbf{f}_J$, where the operator $\mathcal{A}_\Lambda$ is given by

$$\mathcal{A}_\Lambda := \mathcal{M}^{-1}\mathcal{T}_\Lambda \mathcal{M},$$

in which all details of a certain level of regularity have been discarded.

2.3 Multiresolution error estimate

Taking into account the wavelet decomposition (7),

$$f = \sum_{j=-1}^{\infty} \sum_{|\lambda|=j} \langle f, \widetilde{\psi}_\lambda \rangle \psi_\lambda,$$

where $\psi_{-1,k} := \phi_{0,k}$ and $\widetilde{\psi}_{-1,k} := \widetilde{\phi}_{0,k}$, we can construct the array $\mathbf{\Psi}_{J,\lambda}$, $|\lambda| \leq J$, that corresponds to the primal wavelets ψ_λ cell-averaged at level J, *i.e.*, $\mathbf{\Psi}_{J,\lambda} := (\langle \psi_\lambda, \widetilde{\phi}_\gamma \rangle)_{\gamma \in S_J}$. Based on [17], we define the following normalized ℓ^2-norm:

$$\|\mathbf{f}_J\|_2^2 := 2^{-dJ} \sum_{\lambda \in S_J} (f_\lambda)^2,$$

which corresponds to the L^2-norm of a piecewise constant function. For compactly supported wavelets, there is a constant $C > 0$ such that

$$\|\mathbf{\Psi}_{J,\lambda}\|_2 \leq C \|\psi_\lambda\|_{L^2} \leq C 2^{-d|\lambda|/2},$$

and for the multiresolution approximation, we have that

$$\|\mathbf{f}_J - \mathcal{A}_\Lambda \mathbf{f}_J\|_2^2 = \|d_\lambda \mathbf{\Psi}_{J,\lambda}|_{\lambda \notin \Lambda}\|_2^2 \leq C \sum_{\lambda \notin \Lambda} \|d_\lambda\|_{L^2}^2 2^{-d|\lambda|} = C \sum_{\|d_\lambda\|_{L^2} \leq \epsilon_{|\lambda|}} \|d_\lambda\|_{L^2}^2 2^{-d|\lambda|},$$

since only some of the component of $\mathbf{f}_J - \mathcal{A}_\Lambda \mathbf{f}_J$ are non-zero, namely those corresponding to discarded details, and hence the approximation error is bounded by their sum. Considering a level-wise threshold parameter: $\epsilon_j := 2^{dj/2}\epsilon$, the next bound follows (where $\#(\cdot)$ returns the cardinality of a set):

$$\|\mathbf{f}_J - \mathcal{A}_\Lambda \mathbf{f}_J\|_2^2 \leq C \#(\nabla^J) \epsilon^2 = C \#(S_J) \epsilon^2 \leq C 2^{dJ} \epsilon^2,$$

with the cautious assumption that $\|d_\lambda\|_{L^2} = \epsilon_{|\lambda|}$ for all d_λ such that $\lambda \notin \Lambda$ (even though they might be much smaller than $\epsilon_{|\lambda|}$) as well as for the remaining components of $\mathbf{f}_J - \mathcal{A}_\Lambda \mathbf{f}_J$ (even though they are zero). Choosing $\epsilon := 2^{-dJ/2}\eta_{\mathrm{MR}}$ then yields

$$\|\mathbf{f}_J - \mathcal{A}_\Lambda \mathbf{f}_J\|_2 \leq C \eta_{\mathrm{MR}}, \tag{18}$$

with the level-dependent threshold values:

$$\epsilon_j = 2^{d(j-J)/2} \eta_{\mathrm{MR}}, \quad j = 0, 1, \ldots, J.$$

Bound (18) is similarly shown in [17] for both a uniform and ℓ^1 norms.

2.4 Poisson equation discretized on multiresolution grids

Given $V \in H^2(\Omega)$ and $f \in L^2(\Omega)$ that verify the following Poisson equation:

$$\partial_{\boldsymbol{x}}^2 V = f, \tag{19}$$

with $\boldsymbol{x} \in \Omega$, we have that

$$\mathbf{f}_J = (\langle f, \widetilde{\phi}_\gamma \rangle)_{\gamma \in S_J} = (\langle \partial^2_{\boldsymbol{x}} V, \widetilde{\phi}_\gamma \rangle)_{\gamma \in S_J}. \tag{20}$$

Considering that $\mathbf{f}_J \in \mathbb{R}^n$, $n = \#(S_J)$, and the space of square matrices of size n: $\mathcal{M}_n(\mathbb{R})$, let us introduce an operator $\mathbf{A} \in \mathcal{M}_n(\mathbb{R})$ such that following (20),

$$\mathbf{f}_J = \mathbf{A}\mathbf{V}_J + \mathcal{O}\left((\Delta x)^\alpha\right), \tag{21}$$

where $\Delta x := \operatorname{diam}(\Omega_\gamma|_{\gamma \in S_J})$ corresponds to the spatial resolution of the finest grid S_J, and $\mathbf{V}_J := (\langle V, \widetilde{\phi}_\gamma \rangle)_{\gamma \in S_J} \in \mathbb{R}^n$. Notice that this operator $\mathbf{A}$ is no other than a spatial discretization of the Laplace operator. It is therefore a positive definite, and hence non-singular matrix assuming appropriate boundary conditions at $\boldsymbol{x} \in \partial\Omega$ for the Poisson equation (19). In particular following (21), the unique solution $\mathbf{V}_d \in \mathbb{R}^n$ of system $\mathbf{A}\mathbf{V}_d = \mathbf{f}_J$ is an approximation of order α to $\mathbf{V}_J$. Within this framework, the next theorem can be proved.

Theorem 1. *Let $\mathbf{A}$ be a positive definite matrix in $\mathcal{M}_n(\mathbb{R})$ that verifies (21), and $\mathbf{f}^\epsilon_J := \mathcal{A}_\Lambda \mathbf{f}_J$. For $\mathbf{V}^\epsilon \in \mathbb{R}^n$ such that $\mathbf{A}\mathbf{V}^\epsilon = \mathbf{f}^\epsilon_J$, there is a constant $c > 0$ for which*

$$\|\mathbf{V}^\epsilon - \mathbf{V}_J\|_2 \leq c\left((\Delta x)^\alpha + \eta_{\mathrm{MR}}\right).$$

Proof. Proof follows straightforwardly by considering system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{x} = \mathbf{V}^\epsilon - \mathbf{V}_d$ and $\mathbf{b} = \mathbf{f}^\epsilon_J - \mathbf{f}^\epsilon_J$, together with (21) and bound (18). □

Given a spatial discretization of order α, the exact solution $\mathbf{V}_J$ of the Poisson equation can be therefore approximated according to a prescribed tolerance η_{MR}, even if the multiresolution analysis acts on the right-hand side function. In particular it follows that the exact solution $\mathbf{V}_d$ of the discrete Poisson equation $\mathbf{A}\mathbf{V}_d = \mathbf{f}_J$ is approximated by $\mathbf{V}^\epsilon$ in the same way $\mathbf{f}^\epsilon_J$ does for $\mathbf{f}_J$:

$$\|\mathbf{V}_d - \mathbf{V}^\epsilon\|_2 \leq C\eta_{\mathrm{MR}}.$$

Notice however that in a practical implementation the Laplacian will be discretized on an adapted grid, and hence we will not be solving system $\mathbf{A}\mathbf{V}^\epsilon = \mathbf{f}^\epsilon_J$. Nevertheless the next corollary establishes a necessary condition to guarantee the same numerical behavior described in Theorem 1.

Corollary 1. *Let $\widetilde{\mathbf{A}}$ be a positive definite matrix in $\mathcal{M}_n(\mathbb{R})$ such that for any $\mathbf{x} \in \mathbb{R}^n$,*

$$\|(\mathbf{A} - \widetilde{\mathbf{A}})\mathbf{x}\|_2 \leq C\eta_{\mathrm{MR}}. \tag{22}$$

For $\widetilde{\mathbf{V}} \in \mathbb{R}^n$ such that $\widetilde{\mathbf{A}}\widetilde{\mathbf{V}} = \mathbf{f}^\epsilon_J$, there is a constant $c > 0$ for which

$$\|\widetilde{\mathbf{V}} - \mathbf{V}_J\|_2 \leq c\left((\Delta x)^\alpha + \eta_{\mathrm{MR}}\right). \tag{23}$$

Operator $\widetilde{\mathbf{A}}$ corresponds to the Laplacian discretized on an adapted grid and projected onto the S_J-grid. By applying the multiresolution analysis on the right-hand side function and solving the discrete Poisson equation on the corresponding adapted grid, we obtain a solution $\widetilde{\mathbf{V}}$ that also verifies

$$\|\mathbf{V}_d - \widetilde{\mathbf{V}}\|_2 \leq C\eta_{\mathrm{MR}}, \tag{24}$$

as long as (22) is valid. Even though condition (22) seems to be quite strong, in practice we are only interested in the set of arrays $\mathbf{x} \in \mathbb{R}^n$ that reasonably approximate the solution V of a given Poisson equation (19). In this case property (22) is very likely satisfied and perhaps even overestimated, since the adapted grid and thus $\widetilde{\mathbf{A}}$ have been built based on the less regular function f, with respect to the actual solution V.

3 Numerical implementation

We now describe the numerical technique conceived to construct a Poisson solver within the present multiresolution framework. We consider the multiresolution implementation presented in [23]. For the sake of completeness some key aspects of this particular implementation will be first recalled, while more details and references can be found in [20].

3.1 Construction of multiresolution grids

The adapted grid is composed of a set of nested dyadic grids: S_j, $j = 0, 1, \ldots, J$, from the coarsest to the finest, generated by refining recursively a given cell depending on the local regularity of the variables of a given time-dependent problem at a given time. Function f in the Poisson equation (19) that depends directly on these variables (and hence varies also in time) may be additionally considered if necessary, as well as the solution V corresponding to the previous time-step to generate the grids. These grids are implemented in a multi-dimensional and Cartesian finite volume framework. Data compression is achieved by discarding the cells whose *details* are not in Λ according to (17). However, a *graded tree* Λ_ϵ is considered in practice instead of Λ because a certain data structure must be respected in order to carry out the multiresolution transform $\mathcal{M}$ in (16) (see [17] for more details). Notice that $\Lambda \subset \Lambda_\epsilon$ and error estimates like (18) follows straightforwardly with $\mathcal{A}_{\Lambda_\epsilon}$ instead of $\mathcal{A}_\Lambda$. Nevertheless, for the ease of reading we will keep the notation Λ in the following to refer to a graded tree.

A graded tree-structure is used to represent data in the computer memory (see also [47]). Recalling the standard tree-structure terminology: if $\Omega_\mu \subset \Omega_\gamma$ with $|\mu| = |\gamma| + 1$, we say that Ω_μ is a *child* of Ω_γ and that Ω_γ is the *parent* of Ω_μ. We thus define the *leaves* $\mathrm{L}(\Lambda)$ of a *tree* Λ as the set of cells Ω_λ, $\lambda \in \mathrm{L}(\Lambda)$, such that Ω_λ has no children in Λ. The sets ∇_j, $j = 0, 1, \ldots, J$, are distributed in N_R graded trees Λ_r, $r = 1, \ldots, N_\mathrm{R}$, where $N_\mathrm{R} := N_{\mathrm{R}x} N_{\mathrm{R}y} N_{\mathrm{R}z}$, and $N_{\mathrm{R}x}$, $N_{\mathrm{R}y}$, and $N_{\mathrm{R}z}$ stand for the number of graded trees or *roots* per direction. Denoting by $\mathrm{R}(\Lambda_r)$ the set that contains the graded tree Λ_r plus the missing cells Ω_λ in the construction of sets ∇_j, we similarly have that grid indices S_j, $j = 0, 1, \ldots, J$, are distributed in N_R sets $\mathrm{R}(\Lambda_r)$. The adapted grid is thus given by sets $\mathrm{L}(\Lambda_r)$, $r = 1, \ldots, N_\mathrm{R}$, with a total number of cells: $N_\mathrm{L} = \sum_{r=1}^{N_\mathrm{R}} \#(\mathrm{L}(\Lambda_r))$. If no adaptation is required: $\max N_\mathrm{L} = \#(S_J) = N_{\mathrm{R}x} N_{\mathrm{R}y} N_{\mathrm{R}z} 2^{dJ}$, that is, the size of the finest grid. Ghost cells called *phantoms* are added to the adapted grid at level interfaces, in order to always compute numerical fluxes at the highest grid-level between two neighboring cells [47].

Input parameters for the multiresolution implementation are: the maximum grid-level J corresponding to the finest spatial discretization; the number of roots per direction $N_{\mathrm{R}x}$, $N_{\mathrm{R}y}$, and $N_{\mathrm{R}z}$; and the threshold parameter η_MR which defines the numerical accuracy of the compressed representations following (18).

3.2 Construction of the discrete Laplace operator

Introducing the set $\mathrm{I}_\mathrm{L} := \{1, 2, \ldots, N_\mathrm{L}\} \subset \mathbb{Z}$, we define a bijective function $h : D(h) \to \mathrm{I}_\mathrm{L}$, with

$$D(h) := \bigcup_{r=1}^{N_\mathrm{R}} \mathrm{L}(\Lambda_r).$$

The set $\Theta_\mathrm{L} := (\Omega_\lambda)_{h(\lambda) \in \mathrm{I}_\mathrm{L}}$ corresponds then to the adapted grid, defined by the leaves of the tree representation. Taking into account the finite volume framework defined by the dual scaling function $\widetilde{\phi}_\gamma$ in (8), let us consider for a given function $u(\boldsymbol{x}) \in H^2(\Omega)$ and for each γ such that $\gamma \in D(h)$ the following numerical approximation:

$$\langle \partial^2_{\boldsymbol{x}} u, \widetilde{\phi}_\gamma \rangle = |\Omega_\gamma|^{-1} \sum_\mu |\Gamma_{\gamma,\mu}| F_{\gamma,\mu} + \mathcal{O}\left([\mathrm{diam}(\Omega_\gamma)]^\alpha\right), \quad \gamma \in D(h),$$

where the sum is made over all $\mu \neq \gamma$ such that the interface $\Gamma_{\gamma,\mu} := \overline{\Omega_\gamma} \cap \overline{\Omega_\mu}$ is not trivial and $F_{\gamma,\mu}$ accounts for the flux across each interface. Moreover, we can represent the flux computations by

$$|\Omega_\gamma|^{-1}|\Gamma_{\gamma,\mu}|F_{\gamma,\mu} = \sum_{\lambda \in R_F(\gamma)} \alpha_{\gamma,\lambda} u_\lambda, \tag{25}$$

where the stencil $R_F(\gamma) \subset S_{|\gamma|}$ and the set of coefficients $(\alpha_{\gamma,\lambda})_{\lambda \in R_F(\gamma)}$ establish the order α of the approximation. If the same scheme is considered throughout a given S_j, then for any $\mu \neq \gamma$ such that $|\gamma| = |\mu| = j$ the set of coefficients $(\alpha_{\gamma,\lambda})_{\lambda \in R_F(\gamma)}$ and $(\alpha_{\mu,\lambda})_{\lambda \in R_F(\mu)}$ are constant and component-wise equal.

The discrete Laplacian $\mathbf{A} = (a_{i,k})_{i,k \in \mathrm{I_L}}$ on a uniform grid S_J, $\#(S_J) = N_\mathrm{L}$, is hence computed by setting for each $i \in \mathrm{I_L}$, $\gamma = h^{-1}(i)$,

$$a_{h(\gamma),h(\lambda)} = \alpha_{\gamma,\lambda}, \qquad \forall \lambda \in R_F(\gamma), \tag{26}$$

and

$$a_{h(\gamma),k} = 0, \qquad \forall k \in \mathrm{I_L} \text{ s.t. } h^{-1}(k) \notin R_F(\gamma). \tag{27}$$

The finite volume flux representation (25) establishes that for a given interface $\Gamma_{\gamma,\mu}$ the following conservation property is verified: $F_{\gamma,\mu} + F_{\mu,\gamma} = 0$. Computing the flux $F_{\gamma,\mu}$ for Ω_γ amounts to evaluate also $F_{\mu,\gamma}$ for the neighboring cell Ω_μ. Let us denote $F^+_{\gamma,\mu}$ as the right flux for Ω_γ and $F^-_{\mu,\gamma}$ as the left flux for Ω_μ along the normal direction to $\Gamma_{\gamma,\mu}$. Similarly, $R^+_F(\gamma)$ stands for the stencil required to compute $F^+_{\gamma,\mu}$ and naturally $R^-_F(\mu) \equiv R^+_F(\gamma)$. Fluxes can therefore be computed only once at each interface and the same property is exploited to save computations while constructing operator $\mathbf{A}$: for each $i \in \mathrm{I_L}$, $\gamma = h^{-1}(i)$,

$$a_{h(\gamma),h(\lambda)} = \alpha_{\gamma,\lambda}, \qquad \forall \lambda \in R^+_F(\gamma), \tag{28}$$

instead of (26), and

$$a_{h(\mu),h(\lambda)} = -a_{h(\gamma),h(\lambda)}, \quad \overline{\Omega_\gamma} \cap \overline{\Omega_\mu} \neq \varnothing, \forall \lambda \in R^+_F(\gamma). \tag{29}$$

The sparsity of the resulting matrix depends directly on the stencil $R^+_F(\cdot)$ related to the flux computation scheme, while the computational complexity of the procedure is of $\mathcal{O}(\#(S_J))$.

We are nevertheless interested in building the Laplacian $\widetilde{\mathbf{A}} = (\widetilde{a}_{i,k})_{i,k \in \mathrm{I_L}}$ represented on a multiresolution adapted grid, meaning that $N_\mathrm{L} < \#(S_J)$. The principle is the same, as we construct $\widetilde{\mathbf{A}}$ by computing its elements following (27), (28), and (29) with $\widetilde{a}_{i,k}$ instead of $a_{i,k}$. Notice that for a given γ such that $|\gamma| = j$ all fluxes are computed at the same grid S_j in (25). In the case of adapted grids the latter involves that fluxes are computed on a locally uniform grid defined by $R_F(\gamma)$. Ghost cells are thus locally introduced so that for a given γ all cells $\lambda \neq \gamma$ such that $\lambda \in R_F(\gamma)$ are available. Given an adapted tree Λ_r, let us denote by $\mathrm{P}(\Lambda_r)$ the set of phantoms related to the tree Λ_r; that is, all cells with index λ such that for every leaf Ω_γ in Θ_L, $\lambda \in R_F(\gamma)$ but $\lambda \notin \mathrm{R}(\Lambda_r)$. Notice that by construction a phantom is always a child of a leaf. The variable values on these ghost cells are then computed based on the cells contained in the adapted representation $\mathrm{R}(\Lambda_r)$. Using the inter-level operation (14), variables at phantoms are defined by

$$\widehat{u}_\mu = \sum_{|\mu|=|\gamma|+1, \gamma \in R_I(\mu)} \beta_{\mu,\gamma} u_\gamma, \tag{30}$$

where the stencil $R_I(\mu) \subset S_{|\mu|-1}$ and the coefficients $(\beta_{\mu,\gamma})_{\gamma \in R_I(\mu)}$ are defined by the β vanishing moments of the compactly supported dual wavelet. In practice (30) corresponds to a polynomial interpolation of order β with the particular property that the inter-level operation (11) is verified [17]:

$$u_\gamma = |\Omega_\gamma|^{-1} \sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| \widehat{u}_\mu. \tag{31}$$

Recalling that a phantom stands at the place of a discarded cell whose detail was computed based on the same inter-level operations, we have that $\widehat{u}_\mu$ involves an approximation error of $\mathcal{O}(\epsilon_{|\mu|})$ (according to (13)) and the multiresolution error framework remains perfectly valid. Moreover, this construction guarantees a consistent and conservative representation at inter-grid interfaces.

To summarize, given a certain leaf Ω_γ, for each λ such that $\lambda \in R_F(\gamma)$ there are three possibilities:

1. Cell Ω_λ is a leaf and thus belongs to the adapted grid, *i.e.*,
$$\lambda \in \bigcup_{r=1}^{N_\mathrm{R}} \mathrm{L}(\Lambda_r); \tag{32}$$
2. Cell Ω_λ is a phantom, *i.e.*,
$$\lambda \in \bigcup_{r=1}^{N_\mathrm{R}} \mathrm{P}(\Lambda_r); \tag{33}$$
3. Cell Ω_λ belongs to the set of adapted grids but it is not a leaf, *i.e.*,
$$\lambda \in \bigcup_{r=1}^{N_\mathrm{R}} \mathrm{R}(\Lambda_r) \wedge \lambda \notin \bigcup_{r=1}^{N_\mathrm{R}} \mathrm{L}(\Lambda_r). \tag{34}$$

Operator $\widetilde{\mathbf{A}}$ is then a matrix in $\mathcal{M}_n(\mathbb{R})$, $n = N_\mathrm{L}$, (with N_L much smaller than $\#(S_J)$) whose coefficients $(\widetilde{a}_{i,k})_{i,k\in \mathrm{I}_\mathrm{L}}$ are computed following (27)–(29) with $\widetilde{a}_{i,k}$ instead of $a_{i,k}$ whenever (32) is verified, together with the coefficients defined by the inter-level operation (30) (resp., (11)) when (33) (resp., (34)) is verified. Additionally, if a leaf Ω_γ shares an interface $\Gamma_{\gamma,\mu}$ with a leaf of higher resolution, coefficients $\widetilde{a}_{h(\gamma),k}$ are defined considering the weighted sum (31) of the coefficients computed for the corresponding phantoms, children of Ω_γ, at $\Gamma_{\gamma,\mu}$. The latter is equivalent to saying that at inter-level interfaces the matrix coefficients are always evaluated at the highest resolution. For the sake of completeness, the complete scheme to construct the discrete Laplacian is detailed in A. The algorithm (with computational complexity $\mathcal{O}(N_\mathrm{L})$) considers several multiplications and combinations of constant coefficients based on the ideas previously exposed in such a way that we solve for the variables defined only on the adapted grid (the leaves) but strongly coupling inter-grid relations. In practice operator $\widetilde{\mathbf{A}}$ is stored using a standard CSR (Compressed Sparse Row) format for sparse matrices.

4 Streamer discharge simulations

Classical fluid model for streamers in air at atmospheric pressure is given by drift-diffusion equations consistently coupled with a Poisson equation [4, 31]:

$$\left.\begin{aligned}
&\partial_t n_\mathrm{e} + \partial_{\boldsymbol{x}} \cdot (n_\mathrm{e}\, \boldsymbol{v}_\mathrm{e}) - \partial_{\boldsymbol{x}} \cdot (D_\mathrm{e}\, \partial_{\boldsymbol{x}} n_\mathrm{e}) = n_\mathrm{e}\nu_\mathrm{i} - n_\mathrm{e}(\nu_{\mathrm{a}2} + \nu_{\mathrm{a}3}) - n_\mathrm{e} n_\mathrm{p} \beta_\mathrm{ep} + n_\mathrm{n}\gamma + S_\mathrm{ph},\\
&\partial_t n_\mathrm{p} + \partial_{\boldsymbol{x}} \cdot (n_\mathrm{p} \boldsymbol{v}_\mathrm{p}) - \partial_{\boldsymbol{x}} \cdot (D_\mathrm{p}\, \partial_{\boldsymbol{x}} n_\mathrm{p}) = n_\mathrm{e}\nu_\mathrm{i} - n_\mathrm{e} n_\mathrm{p} \beta_\mathrm{ep} - n_\mathrm{n} n_\mathrm{p} \beta_\mathrm{np} + S_\mathrm{ph},\\
&\partial_t n_\mathrm{n} + \partial_{\boldsymbol{x}} \cdot (n_\mathrm{n} \boldsymbol{v}_\mathrm{n}) - \partial_{\boldsymbol{x}} \cdot (D_\mathrm{n}\, \partial_{\boldsymbol{x}} n_\mathrm{n}) = n_\mathrm{e}(\nu_{\mathrm{a}2} + \nu_{\mathrm{a}3}) - n_\mathrm{n} n_\mathrm{p} \beta_\mathrm{np} - n_\mathrm{n}\gamma,
\end{aligned}\right\} \tag{35}$$

$$\varepsilon_0\, \partial_{\boldsymbol{x}} \cdot \boldsymbol{E} = -q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}), \quad \boldsymbol{E} = -\partial_{\boldsymbol{x}} \phi, \tag{36}$$

where $\boldsymbol{x} \in \mathbb{R}^d$, n_i is the density of charged species i (e: electrons, p: positive ions, n: negative ions), ϕ and $\boldsymbol{E}$ stand, respectively, for the electric potential and field, and $\boldsymbol{v}_\mathrm{i} = \mu_\mathrm{i} \boldsymbol{E}$ is the drift velocity. We denote by D_i and μ_i the diffusion coefficient and the mobility of charged species i, q_e is the absolute value of the electron charge, and ε_0 is the permittivity of free space. Moreover, ν_i is the impact

ionization coefficient, ν_{a2} and ν_{a3} are the two-body and three-body electron attachment coefficients, β_{ep} and β_{np} are, respectively, the electron–positive ion and negative–positive ion recombination coefficients, and γ is the detachment coefficient. All these coefficients depend on the local reduced electric field E/N_{air} and thus vary in time and space, where $E = |\boldsymbol{E}|$ is the electric field magnitude and N_{air} is the air neutral density. For test studies presented in this paper, the transport parameters for air are taken from [38]; detachment and attachment coefficients, respectively, from [5, 30]; and other reaction rates, also from [38]. Diffusion coefficients for ions are derived from mobilities using classical Einstein relations. Our reference density for air is $N_{air} = 2.688 \times 10^{19}\,\mathrm{cm}^{-3}$.

For positive streamers a sufficient number of seed-electrons needs to be present in front of the streamer head as the direction of electron motion is opposed to the streamer propagation (see [9] and references therein). Photoionization is in general an accepted mechanism to produce such seed-electrons in nitrogen-oxygen mixtures. It is therefore introduced into the drift-diffusion system (35) as a source term (S_{ph}) that needs to be evaluated in general at each time-step for all points of the computational domain. Computation of S_{ph} is detailed in B which requires the iterative solution of six elliptic equations given by (41) with boundary conditions (42). Iterating three times amounts then to solve 18 Poisson-like equations per time-step.

In what follows we will first assess the theoretical validity of the mathematical description conducted in §2.4 and the numerical implementation described in §3.2 for Poisson equations discretized on multiresolution adapted grids. This study will be conducted on a simplified model with known analytical solution that mimics the spatial configuration typically found in streamer discharges. In a second part we will present double-headed streamer simulations modeled by (35)–(36), for which we will evaluate the performance of different linear solvers implemented to solve the Poisson equations discretized on multiresolution grids. Finally, dynamic grid adaptation will be analyzed for the numerical simulation of two interacting positive streamers in a configuration that leads to streamer merging [35, 8].

4.1 Numerical validation

We first investigate the validity of bound (23) and thus (24). That is, the solution of a Poisson equation discretized on a multiresolution adapted grid is an approximation of $\mathcal{O}(\eta_{MR})$ of the exact solution computed on a uniform grid of equal (finest) resolution, where η_{MR} is the accuracy parameter used by the multiresolution analysis on the right-hand side of the Poisson equation. Given a set of constant parameters: a, b, and σ, let us consider the exponential function $\phi(\boldsymbol{x})$ on a two-dimensional domain $\Omega \subset \mathbb{R}^2$,

$$\phi(\boldsymbol{x}) = g(\boldsymbol{x}) + b = a \exp\left(-|\boldsymbol{x}|^2/\sigma^2\right) + b, \quad \boldsymbol{x} = (x, y) \in \Omega \tag{37}$$

that verifies the following Poisson equation:

$$\partial^2_{\boldsymbol{x}} \phi(\boldsymbol{x}) = \rho(\boldsymbol{x}), \quad \rho(\boldsymbol{x}) = \frac{4}{\sigma^2}\left(\frac{|\boldsymbol{x}|^2}{\sigma^2} - 1\right) g(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{38}$$

with boundary conditions,

$$\phi(\boldsymbol{x}) = g(\boldsymbol{x}) + b, \quad \boldsymbol{x} \in \partial\Omega. \tag{39}$$

Using a standard, second-order centered scheme, we discretize equation (38) on the region: $[-0.5, 0.5] \times [-0.5, 0.5]$, and we consider the set of parameters: $a = 10$, $b = 20$, and $\sigma = 0.005$. The value of σ has been chosen such that function $\rho(\boldsymbol{x})$ exhibits similar steep gradients as those found in a developed streamer head modeled by (35)–(36). Since $g(\boldsymbol{x})$ decays rapidly toward the boundaries, we consider Dirichlet boundary conditions in (39): $\phi(\boldsymbol{x}) = b$, whereas symmetric boundary conditions are taken at $y = 0$ in order to consider only half of the computational domain: $[-0.5, 0.5] \times [0, 0.5]$.

Figure 1 shows normalized L^2-errors between the analytical solution (37) and the numerical solution of the Poisson equation (38) discretized on an adapted grid, obtained with several threshold

values η_{MR}. The resulting linear systems were solved using MUMPS [3, 2], a direct linear system solver. The finest spatial discretization is denoted by Δx, and it is set by the choice of the maximum level J in the multiresolution analysis and the number of roots per direction: $N_{\mathrm{R}x}$ and $N_{\mathrm{R}y}$. In this case, $\Delta x = 1/(N_{\mathrm{R}x}2^J)$ with $J = 5, 6, \ldots, 13$, $N_{\mathrm{R}x} = 10$, and $N_{\mathrm{R}y} = 5$. For streamer discharge simulations an accurate resolution of the electric field: $\boldsymbol{E} = -\partial_{\boldsymbol{x}}\phi$, is essential for good physical descriptions. Therefore we have also computed $\boldsymbol{E}$ with a second order, centered approximation, and compared it against its analytical counterpart: $\boldsymbol{E} = 2\boldsymbol{x}\, g(\boldsymbol{x})/\sigma^2$. In both cases, for ϕ and $\boldsymbol{E}$, the numerical errors behave like a second order spatial approximation even if the solutions are computed on an adapted grid, especially for relatively coarse discretizations or sufficiently fine multiresolution threshold values. For finer resolutions, the numerical errors coming from the adaptive multiresolution become more dominant and the numerical errors are effectively bounded by the threshold parameter η_{MR}. Bounds (23) and (24) prove then to describe accurately the behavior of the numerical approximations when solving a Poisson equation on a multiresolution adapted grid.

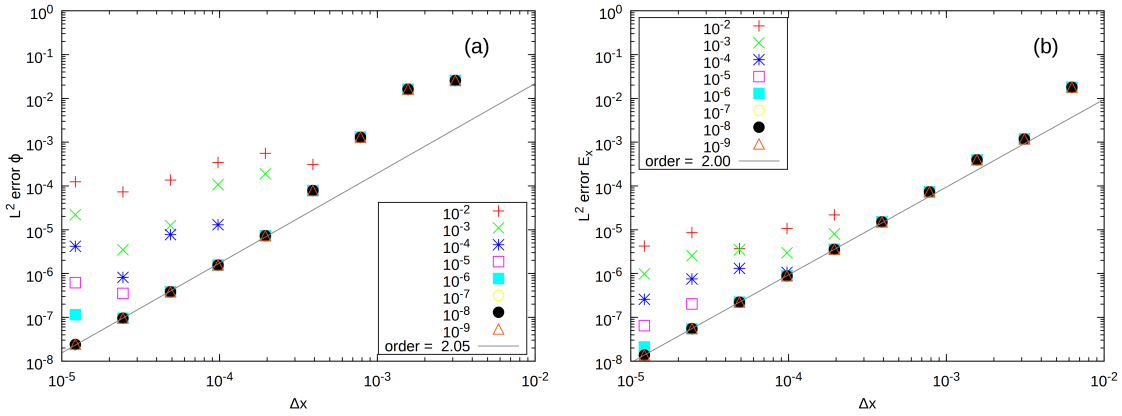

Figure 1: L^2-errors between analytical and numerical solutions ϕ of Poisson equation (38) (a) and component E_x of $\boldsymbol{E} = -\partial_{\boldsymbol{x}}\phi$ (b) for several threshold values η_{MR}.

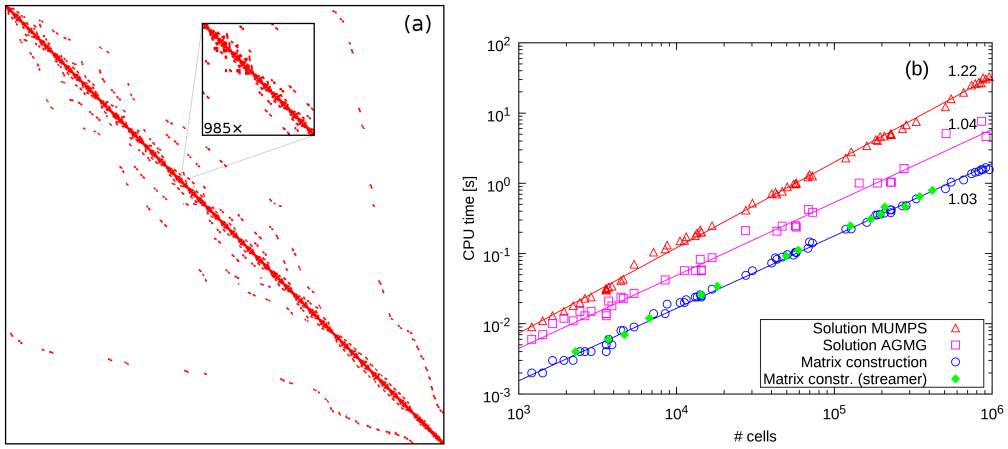

Figure 2: Matrix construction and solution: (a) discrete Laplacian $\widetilde{\mathbf{A}}$ on a multiresolution grid; and (b) CPU times to build $\widetilde{\mathbf{A}}$ and solve the corresponding linear systems for several numbers of cells (slopes of data fits are indicated).

These tests allow us also to verify that the discrete Laplacian is consistently constructed following the procedure established in §3.2, and correctly implemented in practice. The matrix representation $\widetilde{\mathbf{A}}$ is shown in Figure 2(a). Notice that matrix $\widetilde{\mathbf{A}}$ cannot be symmetric, unless no grid adaptation is performed. However $\widetilde{\mathbf{A}}$ is in general quasi-symmetric. For instance, for this particular problem approximately 89 % of symmetry is retrieved in terms of non-zero elements of the matrix. Figure 2(b) illustrates the computational complexity of the matrix construction, which behaves like $\mathcal{O}(N_{\mathrm{L}})$. The different measures were obtained performing several computations with different finest grid-levels $J \in [9, \ldots, 13]$, and multiresolution parameters $\eta_{\mathrm{MR}} \in [10^{-2}, \ldots, 10^{-9}]$. We have also indicated in Figure 2(b) the CPU times to solve the resulting linear system with MUMPS, as well as with an algebraic multigrid solver: AGMG [43, 41, 44] (with tolerances set to 10^{-9}), which also have computational complexity of $\mathcal{O}(N_{\mathrm{L}})$. Finally, we have also verified that building the matrix representation behaves the same way for the more complex streamer configuration presented in the following.

4.2 Performance of linear solvers

We present in what follows a brief study on the performance of several software packages currently available in the literature to solve linear systems of general type: $\mathbf{Ax} = \mathbf{b}$, stemmed in our case from the discretization of a Poisson equation on a multiresolution adapted grid. Two groups of linear solvers were considered: *direct* and *iterative* solvers. Direct methods to solve sparse linear systems of type $\mathbf{Ax} = \mathbf{b}$ involves in general two separate phases: 1) *factorization* of the system matrix $\mathbf{A}$, and 2) *solution* by forward elimination followed by backward substitution to obtain the solution for a given right-hand side $\mathbf{b}$. Main advantages of direct methods are their generality and robustness. Furthermore, direct methods provide an efficient way to solve multiple systems sharing the same system matrix $\mathbf{A}$ but different right-hand sides $\mathbf{b}$, since the most expensive phase (the factorization) needs to be computed only once. On the other hand the main limitation of direct methods is the amount of memory required to store the factorization matrices that may have many (10–100 times) more elements than the original matrix, and may become constraining for large two-dimensional problems and even prohibitive for three-dimensional configurations.

The need of iterative solvers then naturally arises as they exhibit relatively modest memory requirements. However the choice of appropriate algorithms, as well as a fine-tuning of solver parameters to particular linear systems are essential for a satisfactory performance. The number of iterations may also be substantially decreased with a good initial guess of the solution. This feature is somehow exploited in this work in the context of time evolving problems because the previous solution may serve well as an initial guess for the current time-step. Additionally, the system matrix $\mathbf{A}$ changes practically every time-step due to grid adaptation. Iterative solvers are hence favored over direct ones which would require costly factorizations of $\mathbf{A}$ at each time-step. A promising class of iterative methods are multigrid solvers capable to solve sparse linear system of size N with a computational complexity of $\mathcal{O}(N)$. In particular algebraic multigrid methods (AMG) do not require an explicit grid geometry and work directly on matrix entries. They are therefore well-suited to our purposes since system $\mathbf{Ax} = \mathbf{b}$ (coming form a discrete Poisson equation on a multiresolution grid) has completely lost any reminiscence of its original geometric layout.

4.2.1 Test configuration

Let us consider the propagation of a double-headed streamer at atmospheric pressure. In this configuration positive and negative streamers emerge from an initial germ of charged species. Drift-diffusion equations (35) together with Poisson equation (36) are solved following the time-space adaptive scheme introduced in [21]. The latter is based on a decoupled numerical solution of (35) and (36) in such a way that each problem is solved separately by a dedicated solver. Both numerical approximations are assembled according to a second order scheme in time. The latter also considers a time-stepping procedure with error control such that a prescribed accuracy $\eta_{\mathcal{T}}$ is

attained. Variables are represented at cell centers except for the electric field and the velocities which are staggered, while the entire problem is solved on an adapted grid dynamically obtained by multiresolution analysis. The latter is performed on the species densities, variables of the time-dependent transport equations (35). Notice that the right-hand side of the Poisson equation (36) is a linear combination of the species densities so that the theoretical framework in § 2.4 remains valid.

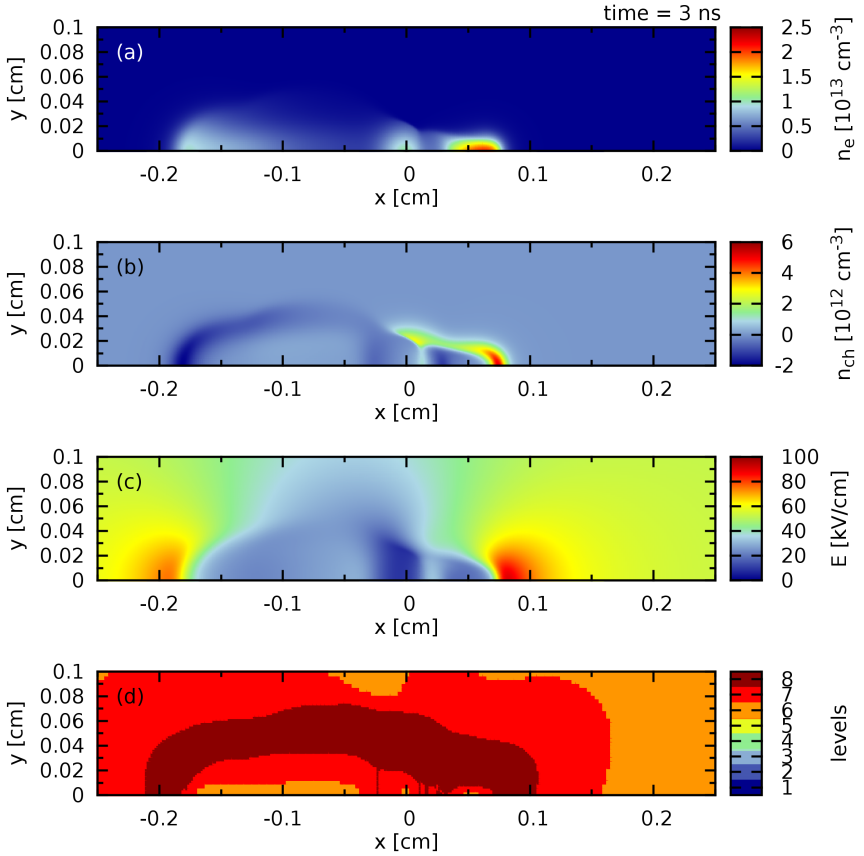

Figure 3: Double-headed streamer at 3.0 ns: (a) electron density n_e; (b) net charged-species density n_{ch}; (c) magnitude of the electric field $E = |\boldsymbol{E}|$; and (d) grid-levels of the adapted mesh. Only part of the computational domain is shown.

Numerical simulations in the present study were carried out for a space-time accuracy tolerance of $\eta_{MR} = \eta_{\mathcal{T}} = 10^{-4}$ with a space resolution of 3.9 μm corresponding to a finest grid level: $J = 8$ with $N_{Rx} = 10$, and $N_{Ry} = 3$. This set of parameters guarantees a sufficiently fine time-space representation of the physics, and numerical results disclosing practically the same behavior with higher spatial resolutions and tighter accuracy tolerances. The computational domain is given by $[-0.5, 0.5] \times [0, 0.3]$ cm in a Cartesian configuration. A homogeneous electric field $\boldsymbol{E} = (48.0, 0)$ kV/cm is introduced via Dirichlet boundary conditions for the Poisson equation at $x = \pm 0.5$ cm, whilst Neumann boundary conditions are applied at $y = 0.3$ cm. A plane of symmetry is imposed at $y = 0$, thus only one half of the streamer is actually simulated. The double-headed streamer is initiated by placing a Gaussian plasma cloud so that the initial conditions for the transport equations (35) are given by

$$n_p(\boldsymbol{x}, 0) = n_e(\boldsymbol{x}, 0) = n_{max} \exp\left(-|\boldsymbol{x}|^2/\sigma^2\right) + n_{0p,e}, \quad n_n(\boldsymbol{x}, 0) = n_{0n},$$

with $\sigma = 0.02\,\mathrm{cm}$, $n_{\max} = 10^{13}\,\mathrm{cm}^{-3}$, and a small homogeneous pre-ionization background of $n_{0\mathrm{n,e}} = 5 \times 10^{-5}\,\mathrm{cm}^{-3}$ and $n_{0\mathrm{p}} = 10^{-4}\,\mathrm{cm}^{-3}$. All tests were conducted starting from the same solution at 3.0 ns when the double-headed streamer is already well developed but no interference with the boundaries is evidenced. The electron density, the net charge species density: $n_{\mathrm{ch}} = n_{\mathrm{p}} - n_{\mathrm{n}} - n_{\mathrm{e}}$, the magnitude of the electric field, and the levels of the adapted grid for the double-headed streamer at 3.0 ns are presented in Figure 3. The total number of cells is of 197784, distributed over five grid-levels from a resolution of $62.5\,\mu$m at level $j = 4$ up to $3.9\,\mu$m at $J = 8$. A data compression of about 10 % is thus achieved with respect to a uniform grid with the finest spatial resolution.

4.2.2 Analysis of results

We have considered both direct and iterative solvers readily available in various software packages. Some of the main characteristics of each one of them are described in C. Most of present day linear solvers are developed with a special attention on enhanced parallel capabilities. Nevertheless, thanks to significant data compression achieved by multiresolution adaptation, the linear systems under consideration have typically about 10^5 unknowns with approximately 10^6 non-zero elements in the system matrix. Therefore to simplify our study we have focused our attention on sequential performance of these solvers. We have performed the numerical experiments on *anila*, a two-processor computer installed at Masaryk University. Each processor is an Intel Xeon CPU E5410 @ 2.33GHz with a total available computer memory of 24 GB. The computer runs on a 64-bit version of Fedora 18 GNU/Linux system. All codes with the various linear solvers were compiled using compilers from GCC (version 4.7.2). Memory requirements of each solver were obtained by tracing the memory profiles of running programs with `top` command, executed in batch mode with a delay-time interval set to 0.01 s. In order to discriminate memory requirements for the linear solvers from the overall program memory usage, a reference program was executed in which calls to the solver were replaced by FORTRAN (GNU extension) `SLEEP` command.

Table 1: Direct solvers: CPU computing time, L^2-error of ϕ and $|\boldsymbol{E}|$ with respect to solutions computed with MUMPS, and memory requirements for each solver.

MUMPS	Memory: 193 MB	
CPU(s)	L^2-error ϕ	L^2-error $\lvert\boldsymbol{E}\rvert$
3.96	0	0
PaStiX	Memory: 259 MB	
CPU(s)	L^2 error ϕ	L^2 error $\lvert\boldsymbol{E}\rvert$
5.38	1.84×10^{-13}	1.45×10^{-12}

Table 1 gathers computation data obtained with two direct solvers: MUMPS [3] and PaStiX [28] in terms of CPU time and memory usage. The total number of unknowns for the Poisson equations considered in this problem is given by the number of cells considered, 197784 in this case, while the discrete Laplacian has 1078534 non-zero entries. In what follows we consider as reference solution the solution to the Poisson equation (36): ϕ, computed with MUMPS. As before we also analyze the approximation to the electric field: $\boldsymbol{E} = -\partial_{\boldsymbol{x}}\phi$. Both solvers yield practically the same solution, while better performances are obtained with MUMPS in terms of both CPU time and memory for this particular problem. Data for three iterative solvers are presented in Table 2 for two algebraic multigrid solvers: AGMG and BoomerAMG [29] (contained in the *hypre* library), and for GMRES [48] preconditioned with BoomerAMG (also contained in *hypre*). In all cases a fine-tuning of computing parameters have been previously carried out so that Table 2 includes the best performances obtained with each of these solvers for this particular problem. A key parameter for iterative solvers is given by the relative and absolute tolerances that in particular serve as stopping criteria to the iterative procedures. In this study we have set both tolerances

Table 2: Iterative solvers: number of iterations (#iter) for relative tolerances: $tol = 10^{-6}, \ldots, 10^{-14}$, CPU computing time, L^2-error of ϕ and $|\boldsymbol{E}|$ with respect to solutions computed with MUMPS, and memory requirements for each solver.

AGMG			Memory: 82 MB	
tol	#iter	CPU(s)	L^2-error ϕ	L^2-error $\|\boldsymbol{E}\|$
10^{-6}	3	0.50	1.65×10^{-5}	1.40×10^{-4}
10^{-7}	4	0.55	1.20×10^{-5}	2.94×10^{-5}
10^{-8}	8	0.78	1.80×10^{-6}	4.20×10^{-6}
10^{-9}	10	0.89	1.43×10^{-7}	4.10×10^{-7}
10^{-10}	14	1.10	2.03×10^{-8}	4.74×10^{-8}
10^{-11}	16	1.21	2.24×10^{-9}	7.44×10^{-9}
10^{-12}	19	1.38	2.09×10^{-11}	9.65×10^{-11}
10^{-13}	20	1.43	1.28×10^{-11}	3.68×10^{-11}
10^{-14}	24	1.64	1.59×10^{-12}	3.86×10^{-12}

***hypre* BoomerAMG**			Memory: 100 MB	
tol	#iter	CPU(s)	L^2 error ϕ	L^2 error $\|\boldsymbol{E}\|$
10^{-6}	3	1.23	7.36×10^{-4}	2.46×10^{-3}
10^{-7}	6	1.51	1.81×10^{-5}	5.99×10^{-5}
10^{-8}	8	1.71	3.97×10^{-6}	1.63×10^{-5}
10^{-9}	11	1.98	9.56×10^{-8}	7.44×10^{-7}
10^{-10}	14	2.27	9.04×10^{-9}	9.87×10^{-8}
10^{-11}	17	2.55	5.45×10^{-10}	5.18×10^{-9}
10^{-12}	20	2.83	6.24×10^{-11}	1.02×10^{-9}
10^{-13}	24	3.21	6.28×10^{-12}	2.58×10^{-11}
10^{-14}	27	3.52	4.73×10^{-13}	3.84×10^{-12}

***hypre* BoomerAMG + GMRES**			Memory: 146 MB	
tol	#iter	CPU(s)	L^2 error ϕ	L^2 error $\|\boldsymbol{E}\|$
10^{-6}	2	1.24	9.09×10^{-4}	2.36×10^{-3}
10^{-7}	5	1.57	2.65×10^{-5}	1.28×10^{-4}
10^{-8}	8	1.90	1.15×10^{-6}	1.49×10^{-5}
10^{-9}	10	2.13	6.19×10^{-8}	8.56×10^{-7}
10^{-10}	12	2.34	4.32×10^{-9}	6.09×10^{-8}
10^{-11}	14	2.58	5.77×10^{-10}	3.33×10^{-9}
10^{-12}	15	2.69	3.58×10^{-10}	8.20×10^{-10}
10^{-13}	17	2.93	3.57×10^{-11}	7.10×10^{-11}
10^{-14}	19	3.15	3.13×10^{-12}	7.73×10^{-12}

equal to an accuracy tolerance, denoted as *tol*. To improve the numerical performances the initial guess corresponds to the solution computed during the previous time-step. For tolerances higher or equal to 10^{-5} convergence is attained right-away with the initial guess for all three solvers. In all cases better performances are obtained with these iterative solvers with respect to direct ones even with very fine accuracy tolerances *tol*. Even though GMRES converges in a less number of iterations for different values of *tol* with respect to the algebraic multigrid solvers, it does not yield faster computations taking into account that for this problem preconditioning is the most expensive

part. BoomerAMG and GMRES/BoomerAMG therefore involve similar computing times.

Considering the solution obtained with MUMPS as the reference solution all these iterative solvers scale well in terms of the accuracy of the approximations, set by the tolerance parameter *tol*. Notice that these numerical errors must be taken into account when using iterative solvers, in particular into bounds (23) and (24). The criterion adopted in this work is that numerical errors coming from the solution of the linear systems must be smaller than the multiresolution ones so that (23) and (24) remain valid. The latter could be enforced by setting in general: $tol < \eta_{\mathrm{MR}}$, while in this particular case a safer choice might be given by $tol \leq 10^{-3} \times \eta_{\mathrm{MR}}$ according to the values contained in Table 2. Among the solvers tested in this study, AGMG revealed itself as the most performing package both in terms of CPU time and memory requirements to solve this particular problem. However it is important to remark that the overall performance of all these solvers are clearly problem-dependent. In this regard the *hypre* library provides a user-friendly and unified interface to various solution schemes, very appropriate to handle different types of problems.

4.3 Application to the study of the interaction of two positive streamers

While the previous illustrations have served to evaluate the performance of the numerical strategy, we consider now an interesting application with more complex dynamics for which the present technique can be further assessed. We study the interaction of two positive streamers initiated to develop side by side. Because the heads of both streamers carry space charge of the same polarity, their mutual interaction should essentially be an electrostatic repulsion. However, it was found that streamers in such a configuration may attract each other and eventually merge [10, 18, 42]. This attraction is mainly the result of the enhancement of photoionization source in the space between the streamer heads [35, 8]. In particular, based on an extensive parametric numerical study, we have shown in [8] that for initial separations of two streamers smaller or comparable to the absorption length of photoionization, merging will start when the ratio of the streamer characteristic width and their mutual separation attains a certain value. We describe here some numerical aspects omitted in our previous study that illustrate how the numerical strategy developed in this work ensures a fine time-space resolution of the complex interaction dynamics of both discharges.

Table 3: Data compression (DC) and number of cells at different grid-levels at sample time instances.

time(ns)	DC(%)	Number of cells at grid-levels						
		2	3	4	5	6	7	8
0.0	0.031	15240	162	272	527	1341	7728	169904
1.0	1.219	780	4671	61359	605154	22297	23160	49424
2.0	1.264	620	4440	57625	633483	23447	27668	47744
3.0	1.291	566	4346	56348	642443	26479	33016	48752
4.0	1.325	528	4528	54141	648333	34030	41660	50672
5.0	1.360	512	4712	51905	652364	42504	56536	47136
6.0	1.407	544	4768	49464	653958	61267	64120	51056
7.0	1.449	458	5288	47652	650502	93008	63196	51728
8.0	1.495	338	5791	46530	644986	131087	66656	45328
9.0	1.567	219	5388	48316	637538	188227	70380	35744
10.0	1.702	18	5296	48622	624126	294256	73264	25280

Let us consider two positive streamers modeled by (35)–(36) propagating in a homogeneous electric field. As before the system of equations is solved with the time-space adaptive scheme in-

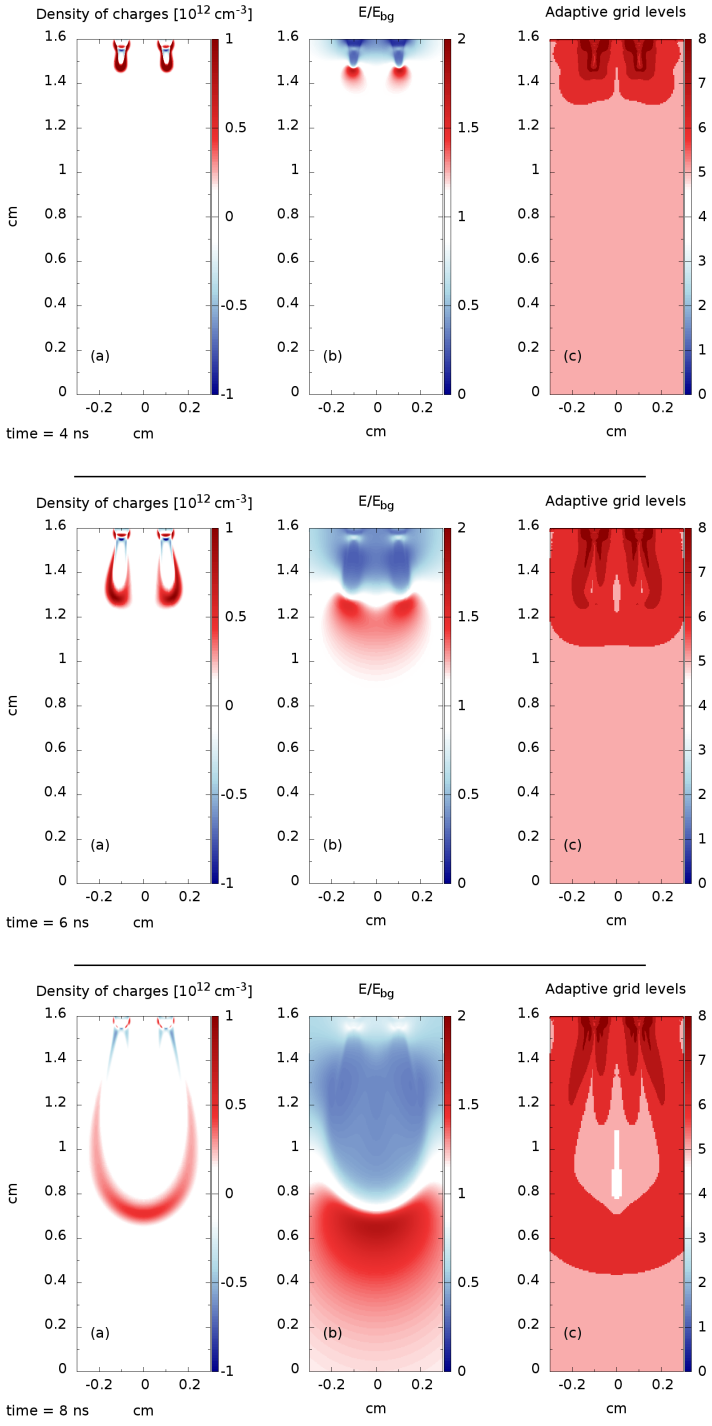

Figure 4: Time evolution of the net charge density (a), magnitude of the electric field (b), and dynamic grid adaptation (c) for two interacting positive streamers at ground pressure with an applied electric field of $\boldsymbol{E}_{\mathrm{bg}} = (0, -48)\,\mathrm{kV/cm}$ at time instances: 4.0, 6.0 and 8.0 ns. Only part of the computational domain is shown.

troduced in [21] with the Poisson equation discretized on the adapted grid following the numerical technique established in § 3.2. The resulting linear systems are solved with MUMPS. The computational domain is given by $[0, 3.0] \times [-1.6, 1.6]$ cm in a Cartesian configuration. A space-time accuracy tolerance of $\eta_{\mathrm{MR}} = \eta_{\mathcal{T}} = 10^{-4}$ was chosen with a space resolution of $3.9\,\mu$m corresponding to a finest grid level of $J = 8$ with $N_{\mathrm{R}x} = 30$ and $N_{\mathrm{R}y} = 32$. The finest spatial resolution is equivalent to that of a uniform grid with 8192×7680 cells. A homogeneous electric field of $\boldsymbol{E}_{\mathrm{bg}} = (0, -48)\,\mathrm{kV/cm}$ is introduced via Dirichlet boundary conditions for the Poisson equation (36) at $y = \pm 1.6\,\mathrm{cm}$, whilst Neumann boundary conditions are applied at $x = 3.0\,\mathrm{cm}$. A plane of symmetry is imposed at $x = 0$. The positive streamer is initiated by placing a Gaussian seed with a maximum of $10^{13}\,\mathrm{cm}^{-3}$ and a characteristic width of $0.02\,\mathrm{cm}$, centered at $0.1\,\mathrm{cm}$ from the symmetry axis. The time evolution of the net charge density, the magnitude of the electric field, and the dynamic grid adaptation at time instances: 4.0, 6.0, and 8.0 ns are shown in Figure 4. Population of different grid-levels at sample times is detailed in Table 3 together with the corresponding data compression (DC), defined as the percentage of active cells with respect to the equivalent number of cells for the finest discretization, given in this case by 62914560. We recall that no grid overlapping is considered in this implementation, that is, both the time-dependent PDEs as well as the Poisson equations are solved on the adapted grid consisting of cells at different grid-levels as shown in Table 3. The coarsest resolution allowed in this simulation (at grid-level $j = 1$) corresponds to a spatial resolution of $0.05\,\mathrm{cm}$ (note that this level was actually not populated during the simulation, therefore it is not listed in Table 3).

From Figure 4 and Table 3 we observe that the finest level is first populated at the vicinity of the initial Gaussian seed and follows the propagation of onsetting streamers (see Figure 4 corresponding to time 4 ns). At the next instance shown (Figure 4 at 6 ns), the propagating front is fully described in a region contained within levels 6 and 7. This is because both streamer heads had expanded and the finest scale is thus no longer necessary. Once the streamer heads have merged (Figure 4 at 8 ns) and therefore only one head is propagating, only level 6 is required. It is interesting to notice that behind the head, *i.e.*, inside the plasma channel where neither sharp gradients nor strong discharge activity are present, the grid is coarsened down to level 4. Nevertheless, the finest resolution is attained and kept throughout the simulation close to the initial Gaussian seeds where we can observe persistence of highly localized space charge as well as strong spatial variation of the electric field. Despite a decreasing population of the finest level after 7 ns (see Table 3) overall data compression is slowly increasing due to the fact that discharge activity is gradually filling larger regions of the computational domain.

5 Concluding remarks

We have presented a brief, yet complete study on the numerical solution of Poisson equations on adapted grids generated by multiresolution analysis arising in the numerical simulation of multiscale propagating fronts. In this context we have recast numerical estimates on the multiresolution errors introduced during the solution of Poisson equations on these adapted grids. On this basis we have developed a numerical procedure to represent the discrete Laplace operator on the adapted grid by reconstructing locally uniform-grid regions at inter-grid interfaces by means of ghost cells and inter-level operations. This approach constitutes a new alternative to the standard level-wise numerical solution of Poisson equations considered in most of the adaptive mesh refinement techniques for time-dependent problems in the literature. In this way the numerical solution of a Poisson equation amounts to considering a linear system completely independent of the grid generation or any other grid-related data structure or geometric consideration, while the multiresolution framework guarantees numerical approximations within an accuracy tolerance as well as consistency and conservation properties throughout the set of grids. Here we have focused our attention on Poisson equations, however the present technique remains valid for more general elliptic PDEs like Poisson equations with time- and/or space-varying coefficients.

To assess the validity of these theoretical and numerical developments we have investigated them in the context of the numerical simulation of streamer discharges. This application involves an intensive use of Poisson solvers and accurate solutions of Poisson equations are essential to the correct reproduction of the physics. In a first step we have carefully evaluated the theoretical bounds previously derived in a simpler configuration with analytical solution. A much more complex and complete model was then considered to simulate the propagation of a double-headed streamer discharge in air at atmospheric pressure. We have thus conducted a study on the performance and capabilities of various direct and iterative linear solvers for this problem that allows us to further validate the current implementation and serves as well as a guide for other applications. In particular we have evaluated the potentialities of algebraic multigrid solvers, well-suited to this kind of implementation with no geometric counterpart. The robustness of the numerical strategy has been further assessed for the simulation of interacting positive streamers, an interesting application in plasma physics. Further developments include optimizing the numerical construction of the discrete Laplace operators by conceiving, for instance, better data structures or by updating only the matrix entries modified by the grid adaptation. Taking into account that in this implementation solving these linear problems becomes a separate aspect from the multiresolution analysis itself, parallel computing capabilities may be directly inherited from the software packages available in the literature. However, an intelligent conjunction with multiresolution parallelism must be sought to achieve overall satisfactory results. These issues constitute particular topics of our current research.

Acknowledgments

This research was supported by a fundamental project grant from ANR (French National Research Agency - ANR Blancs): *Séchelles* (project leader S. Descombes - 2009-2013) and by a DIGITEO RTRA project: *MUSE* (project leader M. Massot - 2010-2014). M. D. was partially supported by the Applied Mathematics Program of the DOE Office of Advance Scientific Computing Research under U.S. Department of Energy under contract No. DE-AC02-05CH11231 at LBNL, and acknowledges support of the Laboratoire EM2C for a visiting stay in France. Z. B. acknowledges support from project CZ.1.05/2.1.00/03.0086 funded by the European Regional Development Fund and support of Ecole Centrale Paris.

A Pseudo-code of the algorithm

Computing the multiresolution representation $\mathbf{f}_J^\epsilon \in \mathbb{R}^n$, $n = N_{\mathrm{L}}$, defines an adapted grid given by the set of leaves: $\Theta_{\mathrm{L}} = (\Omega_\lambda)_{h(\lambda)\in \mathrm{I}_{\mathrm{L}}}$ of size N_{L}. The algorithm to construct the discrete Laplacian: $\widetilde{\mathbf{A}} \in \mathcal{M}_n(\mathbb{R})$, can be schematically described as follows in a Cartesian framework where interfaces are given by $\Gamma_{\gamma,\mu}^{d'}$, $d' = 1, \ldots, d$. This scheme supports space discretizations and compactly supported wavelets of arbitrary order.

for $i = 1 \to N_{\mathrm{L}}$ **do**
  Current leaf: Ω_γ s.t. $\gamma = h^{-1}(i)$.
  for $d' = 1 \to d$ **do**
    Current neighbor: Ω_μ s.t. $\Gamma_{\gamma,\mu}^{d'} = \overline{\Omega_\gamma} \cap \overline{\Omega_\mu}$.
    if $\mu \in D(h)$ **then** $\{\Omega_\mu$ is a leaf, *i.e.*, (32).$\}$
      $i' = h(\mu)$.
      for $\lambda \in R_F^+(\gamma)$ **do**
        if $\lambda \in D(h)$ **then** $\{\Omega_\lambda$ is a leaf, *i.e.*, (32).$\}$
          $k = h(\lambda)$.
          $\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + \alpha_{\gamma,\lambda}$.
          $\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - \alpha_{\gamma,\lambda}$.

else if $\lambda \in \bigcup_{r=1}^{N_{\mathrm{R}}} \mathrm{P}(\Lambda_r)$ **then** {Ω_λ is a phantom, *i.e.*, (33).}
for $\widehat{\lambda}$ s.t. $\widehat{\lambda} \in R_I(\lambda)$ **do**
if $\widehat{\lambda} \in D(h)$ **then** {$\Omega_{\widehat{\lambda}}$ is a leaf, *i.e.*, (32).}
$k = h(\widehat{\lambda})$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + \beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - \beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
else {$\Omega_{\widehat{\lambda}}$ is within the tree, *i.e.*, (34).}
for $\widehat{\lambda}'$ s.t. $\Omega_{\widehat{\lambda}'} \subset \Omega_{\widehat{\lambda}}$ **do**
if $\widehat{\lambda}' \in D(h)$ **then** {$\Omega_{\widehat{\lambda}'}$ is a leaf, *i.e.*, (32).}
$k = h(\widehat{\lambda}')$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + |\Omega_{\widehat{\lambda}}|^{-1}|\Omega_{\widehat{\lambda}'}|\beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - |\Omega_{\widehat{\lambda}}|^{-1}|\Omega_{\widehat{\lambda}'}|\beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
else
for $\widehat{\lambda}''$ s.t. $\Omega_{\widehat{\lambda}''} \subset \Omega_{\widehat{\lambda}'}$ **do**
if $\widehat{\lambda}'' \in D(h)$ **then** {$\Omega_{\widehat{\lambda}''}$ is a leaf, *i.e.*, (32).}
$k = h(\widehat{\lambda}'')$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + |\Omega_{\widehat{\lambda}}|^{-1}|\Omega_{\widehat{\lambda}''}|\beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - |\Omega_{\widehat{\lambda}}|^{-1}|\Omega_{\widehat{\lambda}''}|\beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
else
Continue up to leaves.
end if
end for
end if
end for
end if
else {Ω_λ is within the tree, *i.e.*, (34).}
for λ' s.t. $\Omega_{\lambda'} \subset \Omega_\lambda$ **do**
if $\lambda' \in D(h)$ **then** {$\Omega_{\lambda'}$ is a leaf, *i.e.*, (32).}
$k = h(\lambda')$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + |\Omega_\lambda|^{-1}|\Omega_{\lambda'}|\alpha_{\gamma,\lambda}$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - |\Omega_\lambda|^{-1}|\Omega_{\lambda'}|\alpha_{\gamma,\lambda}$.
else
Continue up to leaves.
end if
end for
end if
end for
else if $\mu \in \bigcup_{r=1}^{N_{\mathrm{R}}} \mathrm{P}(\Lambda_r)$ **then** {Ω_μ is a phantom, *i.e.*, (33).}
for $\lambda \in R_F^+(\gamma)$ **do**
if $\lambda \in D(h)$ **then** {Ω_λ is a leaf, *i.e.*, (32).}
$k = h(\lambda)$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + \alpha_{\gamma,\lambda}$.
for $\widehat{\mu}$ s.t. $\Omega_\mu \subset \Omega_{\widehat{\mu}} \wedge \overline{\Omega_\gamma} \cap \overline{\Omega_{\widehat{\mu}}} \neq \varnothing$ **do**
$i' = h(\widehat{\mu})$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - |\Omega_{\widehat{\mu}}|^{-1}|\Omega_\mu|\alpha_{\gamma,\lambda}$.
end for
else if $\lambda \in \bigcup_{r=1}^{N_{\mathrm{R}}} \mathrm{P}(\Lambda_r)$ **then** {Ω_λ is a phantom, *i.e.*, (33).}
for $\widehat{\lambda}$ s.t. $\widehat{\lambda} \in R_I(\lambda)$ **do**

if $\widehat{\lambda} \in D(h)$ **then** $\{\Omega_{\widehat{\lambda}}$ is a leaf, *i.e.*, (32).$\}$
$k = h(\widehat{\lambda})$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + \beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
for $\widehat{\mu}$ s.t. $\Omega_\mu \subset \Omega_{\widehat{\mu}} \wedge \overline{\Omega_\gamma} \cap \overline{\Omega_{\widehat{\mu}}} \neq \varnothing$ **do**
$i' = h(\widehat{\mu})$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - |\Omega_{\widehat{\mu}}|^{-1}|\Omega_\mu|\beta_{\lambda,\widehat{\lambda}}\alpha_{\gamma,\lambda}$.
end for
else $\{\Omega_{\widehat{\lambda}}$ is within the tree, *i.e.*, (34).$\}$
Continue up to leaves.
end if
end for
else $\{\Omega_\lambda$ is within the tree, *i.e.*, (34).$\}$
Continue up to leaves.
end if
end for
else $\{\Omega_\mu$ is within the tree, *i.e.*, (34).$\}$
for $\widehat{\gamma}$ s.t. $\Omega_{\widehat{\gamma}} \subset \Omega_\gamma \wedge \overline{\Omega_{\widehat{\gamma}}} \cap \overline{\Omega_\mu} \neq \varnothing$ **do**
Current neighbor: $\Omega_{\mu'}$ s.t. $\Omega_{\mu'} \subset \Omega_\mu \wedge \Gamma^{d'}_{\widehat{\gamma},\mu'} = \overline{\Omega_{\widehat{\gamma}}} \cap \overline{\Omega_{\mu'}}$.
$\Omega_{\mu'}$ is a leaf: $i' = h(\mu')$.
for $\lambda \in R_F^+(\widehat{\gamma})$ **do**
if $\lambda \in D(h)$ **then** $\{\Omega_\lambda$ is a leaf, *i.e.*, (32).$\}$
$k = h(\lambda)$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + |\Omega_\gamma|^{-1}|\Omega_{\widehat{\gamma}}|\alpha_{\widehat{\gamma},\lambda}$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - \alpha_{\widehat{\gamma},\lambda}$.
else if $\lambda \in \bigcup_{r=1}^{N_\mathrm{R}} \mathrm{P}(\Lambda_r)$ **then** $\{\Omega_\lambda$ is a phantom, *i.e.*, (33).$\}$
for $\widehat{\lambda}$ s.t. $\widehat{\lambda} \in R_I(\lambda)$ **do**
if $\widehat{\lambda} \in D(h)$ **then** $\{\Omega_{\widehat{\lambda}}$ is a leaf, *i.e.*, (32).$\}$
$k = h(\widehat{\lambda})$.
$\widetilde{a}_{i,k} = \widetilde{a}_{i,k} + |\Omega_\gamma|^{-1}|\Omega_{\widehat{\gamma}}|\beta_{\lambda,\widehat{\lambda}}\alpha_{\widehat{\gamma},\lambda}$.
$\widetilde{a}_{i',k} = \widetilde{a}_{i',k} - \beta_{\lambda,\widehat{\lambda}}\alpha_{\widehat{\gamma},\lambda}$.
else $\{\Omega_{\widehat{\lambda}}$ is within the tree, *i.e.*, (34).$\}$
Continue up to leaves.
end if
end for
else $\{\Omega_\lambda$ is within the tree, *i.e.*, (34).$\}$
Continue up to leaves.
end if
end for
end for
end if
end for
end for

B Photoionization model

The photoionization source term S_ph is evaluated using the three-group SP_3 model developed in [9] with Larsen's boundary conditions [32, 33]. This model considers $N_g = 3$ effective monochromatic radiative transfer equations. As no scattering of photons is taken into account and since the time scale of photon propagation is considered short with respect to the streamer propagation, at each instant of the streamer simulation the photon distribution function $\Psi_l(\boldsymbol{x}, \boldsymbol{\Omega})$ at position $\boldsymbol{x}$ and

direction $\mathbf{\Omega}$ fulfills a radiative transfer equation of the form:

$$\mathbf{\Omega}\cdot\partial_{\boldsymbol{x}}\Psi_l(\boldsymbol{x},\mathbf{\Omega})+\lambda_l p_{\mathrm{O}_2}\Psi_l(\boldsymbol{x},\mathbf{\Omega})=\frac{1}{4\pi}\frac{p_{\mathrm{q}}}{p+p_{\mathrm{q}}}\left(\xi\frac{\nu_{\mathrm{u}}}{\nu_{\mathrm{i}}}\right)\frac{\nu_{\mathrm{i}}n_{\mathrm{e}}}{c\,\xi}, \tag{40}$$

where $l\in\{1,\ldots,N_g\}$ indicates discrete wavelengths, λ_l is the absorption coefficient, p_{O_2} is the partial pressure of molecular oxygen (150 Torr at atmospheric pressure), p is the total pressure, $p_{\mathrm{q}}=30$ Torr is the quenching pressure, $\xi=0.1$ is the photoionization efficiency, ν_{u} is the effective excitation coefficient for N_2 states responsible for ionizing radiation, and ν_{i} and n_{e} are previously introduced ionization coefficient and electron density. The term $(\xi\nu_{\mathrm{u}}/\nu_{\mathrm{i}})$ is given as a function of the reduced electric field in [53, 34]. Finally, c stands for the speed of light. Let us emphasize that monochromatic equations (40) have different absorption coefficients but they all have the same source term that depends on the local reduced electric field E/N_{air}, varying therefore in time and space.

The SP_3 approximation of (40) leads to a set of two elliptic equations for functions $\phi_{1,l}(\boldsymbol{x})$ and $\phi_{2,l}(\boldsymbol{x})$ [32]:

$$\left.\begin{aligned}\partial^2_{\boldsymbol{x}}\phi_{1,l}(\boldsymbol{x})-\frac{\lambda_l^2p_{\mathrm{O}_2}^2}{\kappa_1^2}\phi_{1,l}(\boldsymbol{x})&=-\frac{\lambda_l p_{\mathrm{O}_2}}{\kappa_1^2}\frac{p_{\mathrm{q}}}{p+p_{\mathrm{q}}}\left(\xi\frac{\nu_{\mathrm{u}}}{\nu_{\mathrm{i}}}\right)\frac{\nu_{\mathrm{i}}n_{\mathrm{e}}}{c\,\xi},\\ \partial^2_{\boldsymbol{x}}\phi_{2,l}(\boldsymbol{x})-\frac{\lambda_l^2p_{\mathrm{O}_2}^2}{\kappa_2^2}\phi_{2,l}(\boldsymbol{x})&=-\frac{\lambda_l p_{\mathrm{O}_2}}{\kappa_2^2}\frac{p_{\mathrm{q}}}{p+p_{\mathrm{q}}}\left(\xi\frac{\nu_{\mathrm{u}}}{\nu_{\mathrm{i}}}\right)\frac{\nu_{\mathrm{i}}n_{\mathrm{e}}}{c\,\xi},\end{aligned}\right\} \tag{41}$$

with $\kappa_{1,2}=(1/7)(3\pm2\sqrt{6/5})$. Equations (41) are coupled through the boundary condition. On a boundary surface with neither reflection nor emission, functions $\phi_{1,l}(\boldsymbol{x})$ and $\phi_{2,l}(\boldsymbol{x})$ must verify the following conditions [32, 33]:

$$\left.\begin{aligned}\partial_{\boldsymbol{x}}\phi_{1,l}(\boldsymbol{x})\cdot\boldsymbol{n}_s&=-\lambda_l p_{\mathrm{O}_2}\alpha_1\phi_{1,l}(\boldsymbol{x})-\lambda_l p_{\mathrm{O}_2}\beta_2\phi_{2,l}(\boldsymbol{x}),\\ \partial_{\boldsymbol{x}}\phi_{2,l}(\boldsymbol{x})\cdot\boldsymbol{n}_s&=-\lambda_l p_{\mathrm{O}_2}\alpha_2\phi_{2,l}(\boldsymbol{x})-\lambda_l p_{\mathrm{O}_2}\beta_1\phi_{1,l}(\boldsymbol{x}),\end{aligned}\right\} \tag{42}$$

where $\boldsymbol{n}_s$ is the outward unit normal to the boundary surface, $\alpha_{1,2}=(5/96)(34\pm11\sqrt{6/5})$, and $\beta_{1,2}=(5/96)(2\pm\sqrt{6/5})$. Because $0<\beta_{1,2}\ll\alpha_{1,2}$ the coupling in (41) is weak. A simple strategy to solve (41) together with the boundary conditions (42) consist in solving the equations independently, that is, with $\beta_{1,2}=0$ to then iterate and correct the initial approximations with the inclusion of the $\beta_{1,2}$ coefficients. Convergence is attained very rapidly after few iterations (typically three). The isotropic part of the photon distribution function $\Psi_l(\boldsymbol{x})$ is then written as a linear combination of $\phi_{1,l}(\boldsymbol{x})$ and $\phi_{2,l}(\boldsymbol{x})$ [32]:

$$\Psi_l(\boldsymbol{x})=\frac{\gamma_2\phi_{1,l}(\boldsymbol{x})-\gamma_1\phi_{2,l}(\boldsymbol{x})}{\gamma_2-\gamma_1},$$

with $\gamma_{1,2}=(5/7)(1\pm3\sqrt{5/6})$. The photoionization source term $S_{\mathrm{ph}}(\boldsymbol{x})$ can be finally calculated as [9]:

$$S_{\mathrm{ph}}(\boldsymbol{x})=\sum_{l=1}^{N_g}A_l\xi p_{\mathrm{O}_2}c\Psi_l(\boldsymbol{x}),$$

where parameters A_l together with λ_l are given in Table 4.

C Software packages

In this paper we have considered the following packages:

MUMPS (release 4.10.0), direct solver: MUMPS stands for MUltifrontal Massively Parallel Solver[2] [3, 2], a package to solve linear systems of equations where the system matrix $\mathbf{A}$ is a square

[2]MUMPS home page: `http://graal.ens-lyon.fr/MUMPS/`

Table 4: Parameters for three group photoionization model [9].

l	$A_l\,[\mathrm{cm}^{-1}\,\mathrm{Torr}^{-1}]$	$\lambda_l[\mathrm{cm}^{-1}\,\mathrm{Torr}^{-1}]$
1	0.0067	0.0447
2	0.0346	0.1121
3	0.3059	0.5994

sparse matrix that can be either non-symmetric, symmetric positive definite, or general symmetric. MUMPS implements a direct method based on a multifrontal approach which performs a direct LU-factorization: $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L}$ is a lower triangular matrix and $\mathbf{U}$ an upper triangular one. If the matrix is symmetric then the factorization: $\mathbf{A} = \mathbf{LDL}^T$ is performed, where $\mathbf{D}$ is a block diagonal matrix with blocks of order 1 or 2 on the diagonal. MUMPS is distributed as a public domain software.

PaStiX (release 5.2), direct solver: PaStiX[3] is a parallel multi-threaded library for the solution of large linear systems of equations [28]. For non-symmetric matrices, an LU-decomposition is performed If the system matrix $\mathbf{A}$ is symmetric positive-definite it uses Cholesky ($\mathbf{A} = \mathbf{LL}^T$) or Cholesky-Crout ($\mathbf{A} = \mathbf{LDL}^T$) factorization. PaStiX is distributed under the CeCILL-C license[4].

hypre (release 2.8.0b), iterative solver: *hypre* is a software library of high performance preconditioners and solvers for the solution of large sparse linear systems of equations on massively parallel computing architectures[5]. The library provides advanced parallel preconditioners and parallel multigrid solvers for both structured and unstructured grid setups through a variety of conceptual interfaces, each of them appropriate to a different class of problem. Through the *Linear-Algebraic System Interface* we have access to BoomerAMG [29], an algebraic multigrid solver that may be also used as a preconditioner for GMRES [48]. *hypre* is a free software, distributed under the terms of the GNU Lesser General Public License[6].

AGMG (release 3.1.1), iterative solver: AGMG[7] is an implementation of the algebraic multigrid method developed in [43, 41, 44]. This method solves systems of linear equations and it is in particular expected to be efficient for large systems arising from the discretization of scalar second order elliptic PDEs. It is purely algebraic, no information has to be supplied besides the system matrix and the right-hand side array. Version 3.1.1 (11/2011) was released under GNU GPLv3 (or later) license.

References

References

[1] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, and M.L. Welcome. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.*, 142(1):1–46, 1998.

[3] PaStiX home page: `http://pastix.gforge.inria.fr/`
[4] Licence français de logiciel libre: `http://www.cecill.info/`
[5] *hypre* home page: `http://www.llnl.gov/CASC/linear_solvers/`
[6] `http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html`
[7] AGMG home page: `http://homepages.ulb.ac.be/~ynotay/AGMG/`

[2] P.R. Amestoy, I.S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001.

[3] P.R. Amestoy, I.S. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods in Appl. Mech. Eng.*, 184(2-4):501–520, 2000.

[4] N.Y Babaeva and G.V. Naidis. Two-dimensional modelling of positive streamer dynamics in non-uniform electric fields in air. *J. Phys. D-Appl. Phys.*, 29:2423–2431, 1996.

[5] M.S. Benilov and G.V. Naidis. Modelling of low-current discharges in atmospheric-pressure air taking account of non-equilibrium effects. *J. Phys. D-Appl. Phys.*, 36(15):1834–1841, 2003.

[6] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:67–84, 1989.

[7] M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[8] Z. Bonaventura, M. Duarte, A. Bourdon, and M. Massot. Derivation of a merging condition for two interacting streamers in air. *Plasma Sources Sci. Technol.*, 21(5):052001, 2012.

[9] A. Bourdon, V.P. Pasko, N.Y. Liu, S. Celestin, P. Ségur, and E. Marode. Efficient models for photoionization produced by non-thermal gas discharges in air based on radiative transfer and the Helmholtz equations. *Plasma Sources Sci. Technol.*, 16(3):656–678, 2007.

[10] T.M.P. Briels, J. Kos, E.M. van Veldhuizen, and U. Ebert. Circuit dependence of the diameter of pulsed positive streamers in air. *J. Phys. D-Appl. Phys.*, 39(24):5201–5210, 2006.

[11] K. Brix, S. Melian, S. Müller, and M. Bachmann. Adaptive multiresolution methods: Practical issues on data structures, implementation and parallelization. *ESAIM: Proc.*, 34:151–183, 2011.

[12] R. Bürger, R. Ruiz-Baier, K. Schneider, and M. Sepúlveda. Fully adaptive multiresolution schemes for strongly degenerate parabolic equations in one space dimension. *ESAIM: Math. Model. Numer. Anal.*, 42:535–563, 2008.

[13] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[14] A. Cohen. *Wavelet Methods in Numerical Analysis*, volume 7. Elsevier, Amsterdam, 2000.

[15] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations: Convergence rates. *Math. Comp.*, 70:27–75, 2001.

[16] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5):485–560, 1992.

[17] A. Cohen, S.M. Kaber, S. Müller, and M. Postel. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comp.*, 72:183–225, 2003.

[18] S.A. Cummer, N. Jaugey, J.B. Li, W.A. Lyons, T.E. Nelson, and E.A. Gerken. Submillisecond imaging of sprite development and structure. *Geophys. Res. Lett.*, 33(4):L04104, 2006.

[19] M. Domingues, S. Gomes, O. Roussel, and K. Schneider. Adaptive multiresolution methods. *ESAIM: Proc.*, 34:1–96, 2011.

[20] M. Duarte. *Méthodes numériques adaptatives pour la simulation de la dynamique de fronts de réaction multi-échelles en temps et en espace*. PhD thesis, Ecole Centrale Paris, France, 2011.

[21] M. Duarte, Z. Bonaventura, M. Massot, A. Bourdon, S. Descombes, and T. Dumont. A new numerical strategy with space-time adaptivity and error control for multi-scale streamer discharge simulations. *J. Comput. Phys.*, 231:1002–1019, 2012.

[22] M. Duarte, S. Descombes, C. Tenaud, S. Candel, and M. Massot. Time-space adaptive numerical methods for the simulation of combustion fronts. *Combust. Flame*, (160):1083–1101, 2013.

[23] M. Duarte, M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, and F. Laurent. New resolution strategy for multi-scale reaction waves using time operator splitting, space adaptive multiresolution and dedicated high order implicit/explicit time integrators. *SIAM J. Sci. Comput.*, 34(1):A76–A104, 2012.

[24] U. Ebert, F. Brau, G. Derks, W. Hundsdorfer, C.-Y. Kao, C. Li, A. Luque, B. Meulenbroek, S. Nijdam, V. Ratushnaya, L. Schäfer, and S. Tanveer. Multiple scales in streamer discharges, with an emphasis on moving boundary approximations. *Nonlinearity*, 24(1):C1–C26, 2011.

[25] J.L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Comput. Method Appl. Mech. Eng.*, 195(4447):6011–6045, 2006.

[26] A. Harten. Adaptive multiresolution schemes for shock computations. *J. Comput. Phys.*, 115:319–338, 1994.

[27] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.*, 48:1305–1342, 1995.

[28] P. Hénon, P. Ramet, and J. Roman. PaStiX: A high-performance parallel direct solver for sparse symmetric definite systems. *Parallel Comput.*, 28(2):301–321, 2002.

[29] V.E. Henson and U.M. Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41:155–177, 2002.

[30] I.A. Kossyi, A.Y. Kostinsky, A.A. Matveyev, and V.P. Silakov. Kinetic scheme of the non-equilibrium discharge in nitrogen-oxygen mixtures. *Plasma Sources Sci. Technol.*, 1(3):207–220, 1992.

[31] A.A. Kulikovsky. Positive streamer between parallel plate electrodes in atmospheric pressure air. *J. Phys. D-Appl. Phys.*, 30:441–450, 1997.

[32] E.W. Larsen, G. Thommes, A. Klar, M. Seaid, and T. Gotz. Simplified P-N approximations to the equations of radiative heat transfer and applications. *J. Comput. Phys.*, 183(2):652–675, 2002.

[33] N.Y. Liu, S. Celestin, A. Bourdon, V.P. Pasko, P. Ségur, and E. Marode. Application of photoionization models based on radiative transfer and the Helmholtz equations to studies of streamers in weak electric fields. *Appl. Phys. Lett.*, 91(21):211501, 2007.

[34] N.Y. Liu and V.P. Pasko. Effects of photoionization on propagation and branching of positive and negative streamers in sprites. *J. Geophys. Res.*, 109:A04301, 2004.

[35] A. Luque, U. Ebert, and W. Hundsdorfer. Interaction of streamer discharges in air and other oxygen-nitrogen mixtures. *Phys. Rev. Lett.*, 101(7):075005, 2008.

[36] D.F. Martin, P. Colella, and D. Graves. A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions . *J. Comput. Phys.*, 227(3):1863–1886, 2008.

[37] C. Montijn, W. Hundsdorfer, and U. Ebert. An adaptive grid refinement strategy for the simulation of negative streamers. *J. Comput. Phys.*, 219(2):801–835, 2006.

[38] R. Morrow and J.J. Lowke. Streamer propagation in air. *J. Phys. D-Appl. Phys.*, 30:614–627, 1997.

[39] S. Müller. *Adaptive Multiscale Schemes for Conservation Laws*, volume 27. Springer-Verlag, 2003.

[40] S. Müller. Multiresolution schemes for conservation laws. In R. DeVore *et al.*, editor, *Multiscale, Nonlinear and Adaptive Approximation*, pages 379–408. Springer Berlin Heidelberg, 2009.

[41] A. Napov and Y. Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM J. Sci. Comput.*, 34:A1079–A1109, 2012.

[42] S. Nijdam, C.G.C. Geurts, E.M. van Veldhuizen, and U. Ebert. Reconnection and merging of positive streamers in air. *J. Phys. D-Appl. Phys.*, 42(4):045201, 2009.

[43] Y. Notay. An aggregation-based algebraic multigrid method. *Electronic Trans. Numer. Anal.*, 37:123–146, 2010.

[44] Y. Notay. Aggregation-based algebraic multigrid for convection-diffusion equations. *SIAM J. Sci. Comput.*, 34:A2288–A2316, 2012.

[45] S. Pancheshnyi, P. Ségur, J. Capeillère, and A. Bourdon. Numerical simulation of filamentary discharges with parallel adaptive mesh refinement. *J. Comp. Phys.*, 227(13):6574–6590, 2008.

[46] Y.P. Raizer. *Gas Discharge Physics*. Springer-Verlag, 1991.

[47] O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *J. Comput. Phys.*, 188(2):493–523, 2003.

[48] Y. Saad and M. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comp.*, 7(3):856–869, 1986.

[49] C. Safta, J. Ray, and H.N. Najm. A high-order low-Mach number AMR construction for chemically reacting flows. *J. Comput. Phys.*, 229(24):9299–9322, 2010.

[50] R. Témam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires. II. *Arch. Rational Mech. Anal.*, 33:377–385, 1969.

[51] R. Teyssier. Cosmological hydrodynamics with adaptive mesh refinement - A new high resolution code called RAMSES. *A&A*, 385(1):337–364, 2002.

[52] T. Unfer, J.-P. Boeuf, F. Rogier, and F. Thivet. Multi-scale gas discharge simulations using asynchronous adaptive mesh refinement. *Comput. Phys. Commun.*, 181(2):247–258, 2010.

[53] M.B. Zheleznyak, A.K. Mnatsakanyan, and S.V. Sizykh. Photo ionization of nitrogen and oxygen mixtures by radiation from a gas discharge. *High Temp.*, 20(3):357–362, 1982.