Computer Physics Communications 225 (2018) 114–121

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A domain-decomposition method to implement electrostatic free boundary conditions in the radial direction for electric discharges ☆

A. Malagón-Romero *, A. Luque

IAA-CSIC, P.O. Box 3004, 18080 Granada, Spain

ARTICLE INFO

Article history:
Received 30 July 2017
Received in revised form 10 November 2017
Accepted 4 January 2018
Available online 31 January 2018

Keywords:
Electric discharge
Streamer
Domain decomposition
Poisson equation

ABSTRACT

At high pressure electric discharges typically grow as thin, elongated filaments. In a numerical simulation this large aspect ratio should ideally translate into a narrow, cylindrical computational domain that envelops the discharge as closely as possible. However, the development of the discharge is driven by electrostatic interactions and, if the computational domain is not wide enough, the boundary conditions imposed to the electrostatic potential on the external boundary have a strong effect on the discharge. Most numerical codes circumvent this problem by either using a wide computational domain or by calculating the boundary conditions by integrating the Green's function of an infinite domain. Here we describe an accurate and efficient method to impose free boundary conditions in the radial direction for an elongated electric discharge. To facilitate the use of our method we provide a sample implementation. Finally, we apply the method to solve Poisson's equation in cylindrical coordinates with free boundary conditions in both radial and longitudinal directions. This case is of particular interest for the initial stages of discharges in long gaps or natural discharges in the atmosphere, where it is not practical to extend the simulation volume to be bounded by two electrodes.

Program summary

Program Title: `poisson_sparse_fft.py`
Program Files doi: http://dx.doi.org/10.17632/x7f6czrnsh.1
Licensing provisions: CC By 4.0
Programming language: Python
Nature of problem: Electric discharges are typically elongated and their optimal computational domain has a large aspect ratio. However, the electrostatic interactions within the discharge volume may be affected by the boundary conditions imposed to the Poisson equation. Computing these boundary conditions using a direct integration of Green's function involves either heavy computations or a loss of accuracy.
Solution method: We use a Domain Decomposition Method to efficiently impose free boundary conditions to the Poisson equation. This code provides a stand-alone example implementation.

1. Introduction

Despite their prevalence in industry and in nature, electric discharges still hold many unknowns. For example, we do not yet understand precisely how a lightning channel starts, how it advances in its way to the ground or how exactly are bursts of X-rays produced as it progresses [1]. This is partly due to the short time and length scales involved in such processes which, combined with their jittery behavior, prevents the use of many diagnostic techniques. Due to these limitations, much of what we know about electric discharges comes from computer models which, at least within a simulation, are predictable and reveal arbitrarily small scales.

Consider streamer simulations. Streamers are thin filaments of ionized air that precede most electric discharges in long gaps at atmospheric pressure. The main challenge for simulating streamers is the wide separation between length scales: whereas the total length of the streamer channel at atmospheric pressure ranges from about one to some tens of centimeters, the ionization of air molecules is mostly confined to a layer thinner than one millimeter. Despite this difficulty, there are many numerical codes that explain most of the observed properties of streamers [2–7]. In the past decades these models have gradually improved and successfully overcome many of the challenges posed by streamer physics. However, they are still computationally intensive and often require days of runtime to produce meaningful simulations.

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: amaro@iaa.es (A. Malagón-Romero).

https://doi.org/10.1016/j.cpc.2018.01.003

In this work we look at one of the problems behind these long running times: the large aspect ratio of a single-channel discharge. Whereas the width of an atmospheric-pressure streamer is at most about one centimeter, its length spans many times this extension. In order to minimize the amount of work performed in a simulation, one strives to adapt the computational domain to the dimensions of the streamer, which means using a narrow cylindrical domain with a diameter only slightly larger than the streamer width. However, in such a narrow domain the electrostatic interaction between separate points in the channel is strongly affected by the boundary conditions imposed on the electric potential at the outer boundaries.

One approach to avoid this artifact while keeping a narrow domain around the streamer is to calculate the boundary values of the potential by direct integration of the electrostatic Green's function in free space [3,8–11]. These values are then imposed as inhomogeneous Dirichlet boundary conditions in the solution of the Poisson equation. In a cartesian grid with M cells in the radial direction and N cells in the axial direction the direct integration of the Green's function at each of the N nodes in the external boundary requires about MN^2 operations. Since the work employed by fast Poisson solvers scales as $MN\log(MN)$ (MN for multigrid solvers), the computation of boundary values by direct integration may easily dominate the work employed in the electrostatic calculations. This is mitigated in part by using a coarse-grained charge distribution in the integration. However, in that case there is a tradeoff between the degree of coarsening and the minimal radial extension of the domain required for a tolerable error.

Beyond this common approach used to solve Poisson's equation in electric discharges, some other methods have been developed. A family of these methods has been built upon the idea of the decoupling of local and far-field effects [12] and the computation of the boundary potential by means of a potential generated by a set of screening charges located in the outer surface of the computational domain [13]. Based on these two methods mentioned above, reference [14] uses a domain decomposition approach to exploit parallel computing capabilities; first, Poisson's equation subject to unbounded boundary conditions is solved in a set of disjoint patches. As a second step a coarse-grid representation of the space charge is obtained and Poisson's equation is again solved in a global coarse-grid whose solution is used to communicate far-field effects to local patches. Finally, Poisson's equation is solved in a fine grid using boundary conditions computed from the coarse-grid solution corrected with local field information.

A different family of methods uses the convolution with Green's function subject to free boundary conditions. They manage the singular behavior of Green's function by either regularizing it [15], or by replacing the singular component to the integrand of the convolution by an analytical contribution [16]. These methods have achieved an order of convergence greater than two.

Here we adapt to the cylindrical geometry of electric discharges the domain-decomposition method described by Anderson [17] (see also [18] for a review of similar techniques). As we discuss below, this method requires two calls to the Poisson solver but otherwise the leading term in its algorithmic complexity follows the scaling of the Poisson solver itself. Therefore for large grid sizes our approach is more efficient than the direct integration method. Furthermore, as we do not reduce the resolution, we do not introduce any numerical error in addition to the discretization error of the Poisson equation. We believe that the method we present is simple enough that it can be easily implemented on top of any existing streamer simulation code. To aid in this task we provide a standalone example in Python.

Some applications may also require free boundary conditions for the z-direction: for example, when the discharge develops far from the electrodes. In those cases one may also reduce the computational domain in the longitudinal direction while the core of the simulation remains inside the computational domain. We have considered this topic of interest in Appendix A where we have applied the domain decomposition method to obtain free boundary conditions also in the longitudinal direction. This extension requires an extra solution of Poisson's equation.

Note that streamers are not the only type of discharge that typically exhibits a large aspect ratio and that therefore our scheme is also applicable to other processes such as leaders and arcs.

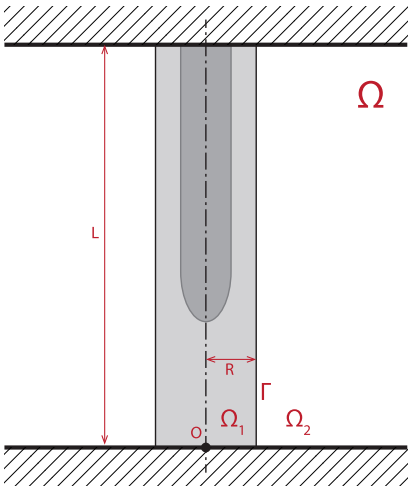

Fig. 1. Geometry of the discharge considered in this work. An elongated channel propagates between two conducting electrodes. The space between these electrodes, Ω is divided into two domains: the inner domain Ω_1 is our computational domain and contains all the space charge. The outer domain Ω_2 extends indefinitely outwards from the external boundary of Ω_1 and does not contain any space charge. The cylindrical surface Γ is the common boundary between Ω_1 and Ω_2.

2. Description of the method

2.1. Domain decomposition

The most convenient decomposition of the domain strongly depends on the problem at hand. The decomposition we present here is suitable for elongated discharges and probably some other applications but the procedure and the highlighted ideas are not restricted to this particular scheme.

We consider the geometry sketched in Fig. 1, where an elongated, cylindrically symmetrical streamer propagates between two planar electrodes. With minimal changes, our scheme can be extended to more complex geometries commonly employed in streamer simulations, such as protrusion–plane, protrusion–protrusion and sphere–plane. The electrostatic potential ϕ satisfies the Poisson equation with appropriate boundary conditions:

$$\begin{aligned} \Delta\phi &= f \quad \text{in} \quad \Omega, \\ \phi &= g \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{1}$$

where $f = -q/\epsilon_0$, with q being the charge density and ϵ_0 the vacuum permittivity. In principle an arbitrary boundary condition, here denoted by g, can be applied to the upper and lower electrodes. However, to simplify our discussion we limit ourselves to the most common case where $g = 0$, meaning $\phi = 0$ at $z = 0$ and $z = L$ (to impose a potential difference V between the two electrodes we simply add $\phi_{\text{inhom}} = zV/L$ to the solution of the homogeneous problem). The domain Ω is the space between the two electrodes, formally defined as

$$\Omega = \left\{ x \equiv (\rho, \theta, z) \in \mathbb{R}^3 / 0 \leq \rho, 0 \leq \theta < 2\pi, 0 \leq z \leq L \right\}. \tag{2}$$

Since our geometry is cylindrically symmetrical, we will henceforth omit the variable θ and consider the two-dimensional domain spanned by the variables (ρ, z).

Our purpose is to decompose the physical domain Ω into two, which we name Ω_1 and Ω_2, such that $\Omega = \overline{\Omega}_1 \cup \overline{\Omega}_2$ ($\overline{\Omega}_i$ is the closure of the set Ω_i), $\Omega_1 \cap \Omega_2 = \emptyset$ and $\text{supp}(f) \subset \Omega_1$, i.e. all the space charge is contained in Ω_1. The inner domain Ω_1, extending up to a given radius R, is our computational domain and therefore must be selected to be as narrow as possible.

Under this domain decomposition the problem (1) turns into two coupled problems:

$$\begin{aligned} \Delta\phi_i &= f \quad \text{in} \quad \Omega_i, \\ \phi_i &= 0 \quad \text{on} \quad \partial\Omega_i \setminus \Gamma, \\ \phi_i &= \phi_\Gamma \quad \text{on} \quad \partial\Gamma, \end{aligned} \tag{3}$$

where $i = 1, 2$ and $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ is the cylindrical surface at $\rho = R$ that separates the two domains.

Since at the interface Γ both, ϕ_1 and ϕ_2, are equal to the boundary value ϕ_Γ, they fulfill $\phi_1 = \phi_2$. But besides this condition, in order for ϕ_1 and ϕ_2 to be consistent with the solution ϕ of the original problem (1), they must also satisfy

$$\frac{\partial\phi_1}{\partial\rho} = \frac{\partial\phi_2}{\partial\rho} \text{ on } \Gamma. \tag{4}$$

2.2. Linearity

The linearity of the Poisson problems (3) with respect to their sources f allows us to decompose the potentials as

$$\phi_i = \bar{\phi}_i[\phi_\Gamma] + \tilde{\phi}_i[f_i], \tag{5}$$

where $\bar{\phi}_i[\phi_\Gamma]$ results from the boundary values ϕ_Γ at the interface Γ and $\tilde{\phi}_i[f_i]$ results from the original sources f restricted to Ω_i (we use $[\cdot]$ to denote a functional dependence). The precise definitions read

$$\begin{aligned} \Delta\bar{\phi}_i &= 0 \quad \text{in} \quad \Omega_i, \\ \bar{\phi}_i &= 0 \quad \text{on} \quad \partial\Omega_i \setminus \Gamma, \\ \bar{\phi}_i &= \phi_\Gamma \quad \text{on} \quad \partial\Gamma, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \Delta\tilde{\phi}_i &= f \quad \text{in} \quad \Omega_i, \\ \tilde{\phi}_i &= 0 \quad \text{on} \quad \partial\Omega_i \setminus \Gamma, \\ \tilde{\phi}_i &= 0 \quad \text{on} \quad \partial\Gamma. \end{aligned} \tag{7}$$

In terms of these components the flux equation (4) can be expressed as

$$\frac{\partial\bar{\phi}_1}{\partial\rho}[\phi_\Gamma] - \frac{\partial\bar{\phi}_2}{\partial\rho}[\phi_\Gamma] = -\frac{\partial\tilde{\phi}_1}{\partial\rho}[f] \text{ on } \Gamma, \tag{8}$$

where on the right hand side we have made use of $\tilde{\phi}_2 = 0$, since $f = 0$ in Ω_2.

2.3. Expansion in orthonormal solutions of the Laplace equation

The potentials $\bar{\phi}_i$ in (6) are solutions of the Laplace equation in cylindrical geometry and they can be expanded using an orthogonal basis of solutions (see e.g. [19]):

$$\bar{\phi}_1 = \sum_{m=1}^{\infty} \alpha_m I_0(k_m\rho)\sin(k_m z), \tag{9a}$$

$$\bar{\phi}_2 = \sum_{m=1}^{\infty} \beta_m K_0(k_m\rho)\sin(k_m z), \tag{9b}$$

where α_m and β_m are expansion coefficients, $k_m = m\pi/L$ and $I_n(x)$ and $K_n(x)$ are the modified n-order Bessel functions of the first and second kind respectively. Note that the set $S = \{\sin(k_m z)\}_{m=0}^{\infty}$ is an orthogonal basis of

$$\mathcal{L}^2([0, L]) = \left\{ f : [0, L] \mapsto \mathbb{R} : \int |f(z)|^2 dz < \infty \right\}, \tag{10}$$

therefore, ϕ_Γ can be expanded as:

$$\phi_\Gamma(z) = \sum_{m=1}^{\infty} a_m \sin(k_m z). \tag{11}$$

If ϕ_Γ is continuous and piecewise differentiable on $[0, L]$, $\phi'_\Gamma \in \mathcal{L}^2([0, L])$ and ϕ_Γ satisfies homogeneous Dirichlet boundary conditions, then the sine series converges to ϕ_Γ uniformly on $[0, L]$. Note that the term with $m = 0$ vanishes due to the homogeneous boundary conditions at $z = 0$ and $z = L$.

The boundary conditions at $z = 0$ and $z = L$ restrict the basis of solutions. Homogeneous Dirichlet boundary conditions are simpler because there is only need for sine functions. However, if we had some other boundary conditions such as homogeneous Neumann, the convenient basis should also include cosine functions to allow for non-zero values of the potential at $z = 0$ and $z = L$. Nevertheless, this basis is not orthogonal and this would make things slightly more complicated.

2.4. Continuity of the normal derivative

Imposing that $\bar{\phi}_1 = \bar{\phi}_2 = \phi_\Gamma$ at $\rho = R$ we solve for α_m and β_m and write (9) as

$$\bar{\phi}_1 = \sum_{m=1}^{\infty} a_m \frac{I_0(k_m\rho)}{I_0(k_m R)}\sin(k_m z), \tag{12a}$$

$$\bar{\phi}_2 = \sum_{m=1}^{\infty} a_m \frac{K_0(k_m\rho)}{K_0(k_m R)}\sin(k_m z). \tag{12b}$$

Using these expressions into the equation for the normal derivatives (8) we obtain

$$\sum_{m=1}^{\infty} a_m k_m \left[\frac{I_1(k_m R)}{I_0(k_m R)} + \frac{K_1(k_m R)}{K_0(k_m R)}\right]\sin(k_m z) = -\left.\frac{\partial\tilde{\phi}_1}{\partial\rho}\right|_{\rho=R}, \tag{13}$$

where we have made use of the identities $I'_0(x) = I_1(x)$, $K'_0(x) = -K_1(x)$. Using now the orthogonality of the basis S we obtain equations for the coefficients a_m:

$$\frac{L}{2} k_m a_m \left[\frac{I_1(k_m R)}{I_0(k_m R)} + \frac{K_1(k_m R)}{K_0(k_m R)}\right] = -\int_0^L dz \, \sin(k_m z) \left.\frac{\partial\tilde{\phi}_1}{\partial\rho}\right|_{\rho=R}. \tag{14}$$

In a space discretization based on a cartesian grid the integral in the latest expression is approximated by a finite sum with the form of a Discrete Sine Transform (DST). This leads to this final expression for the coefficients a_m

$$a_m = -\frac{2}{m\pi}\left[\frac{I_1(k_m R)}{I_0(k_m R)} + \frac{K_1(k_m R)}{K_0(k_m R)}\right]^{-1} \times \sum_{i=1}^{N} h \sin(k_m z_i) \left.\frac{\partial\tilde{\phi}_1}{\partial\rho}\right|_{\rho=R, z=z_i} + \mathcal{O}(h^2), \tag{15}$$

where h is the grid size and $\{z_i\}_{i=1}^{N}$ are the solution nodes in the z-direction. In a discrete problem the series in (11) is also truncated above $m = N$.

2.5. Algorithm

We are now ready to detail the domain-decomposition algorithm that allows us to solve the Poisson equation in the reduced computational domain Ω_1 with free boundary conditions:

1. Solve the Poisson equation in Ω_1 with the source term f and homogeneous Dirichlet boundary conditions at the boundary Γ. Call the result $\tilde{\phi}_1$.
2. Calculate the normal derivative of $\tilde{\phi}_1$ at Γ. Apply a DST and use expression (15) to obtain the coefficients a_m.
3. Use these coefficients to obtain the boundary values ϕ_Γ by means of a second DST and expression (11).
4. Solve again the Poisson equation in Ω_1 but now use ϕ_Γ as inhomogeneous Dirichlet boundary condition at Γ. The result, ϕ_1 is the solution of the Poisson equation with free boundary conditions.

To this algorithm we add the following remarks:

1. After obtaining the coefficients a_m one is tempted to use (12a) together with $\phi_1 = \bar{\phi}_1 + \tilde{\phi}_1$ to avoid solving the Poisson equation a second time. However, in a grid of $M \times N$ cells this procedure takes about MN^2 operations whereas solving the Poisson equation requires only $MN \log(MN)$ or MN operations.
2. The computational domain Ω_1 has to be as narrow as possible in order to reduce the computational cost of the simulation. Of course this narrowing is limited by the constraint that Ω_1 contains the support of the space charge density. In an electrostatic discharge the charge density typically decays smoothly away from the channel so in some cases one has to decide at which level it is safe to truncate the charge density with an acceptable error. Nevertheless, given the fast decay of the charge away from the channel, this is probably not a serious concern in most cases.

3. Tests and sample implementation

3.1. Tests

In order to test our scheme we consider now a simple setup where the Poisson equation has a closed-form solution. An example of such a configuration is a uniformly charged sphere located between two grounded, infinite planar electrodes. The electrostatic potential in this setup can be calculated by the method of images (see e.g. [19]) and equals the potential created in free space by an infinite series of spheres with alternating charges.

Suppose a sphere centered at $(\rho, z) = (0, z_0)$ with radius $a < \min(z_0, L - z_0)$ and total charge Q. At a point with cylindrical coordinates (ρ, z) the potential reads

$$\phi(\rho,z) = \phi_0(\rho,z) + \frac{Q}{4\pi\epsilon_0}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}\frac{(-1)^k}{\left[\rho^2+(z-z_0-2k(L-z_0))^2\right]^{1/2}}, \tag{16a}$$

with

$$\phi_0(\rho,z) = \frac{Q}{4\pi\epsilon_0} \times \begin{cases} \dfrac{1}{\left[\rho^2+(z-z_0)^2\right]^{1/2}} & \text{if } \rho^2+(z-z_0)^2 > a^2, \\ \dfrac{3a^2-\rho^2-(z-z_0)^2}{2a^3} & \text{if } \rho^2+(z-z_0)^2 \le a^2. \end{cases} \tag{16b}$$

Fig. 2 shows a comparison between the electric fields computed using expression (16b) and using the approach described in section 2. Here we took $a = 3$ mm, $L = 10$ mm, $Q = 10^{13}\,e$ (e is the elementary charge), $z_0 = L/2$. For the discretized solution we used $\Delta r = \Delta z = 10^{-2}$ mm and a radial extension of the computational domain $R = 5$ mm. We also include the electrostatic potential calculated by imposing homogeneous Neumann boundary conditions at the external boundary.

We see that the field calculated with the approach presented here is indistinguishable from the field from the method of images. The homogeneous Neumann conditions, on the other hand, produce an electric field that at the surface of the sphere deviates by about 15% from the other two in the worst case, i.e. with $R = 5$ mm. To investigate the convergence of the homogeneous Neumann solution we extended the computational domain by computing the field also for $R = 10$ mm and $R = 20$ mm. As we move the external boundary away, the solution with Neumann conditions approaches our reference solution (Method of Images). Essentially, bringing the external boundary closer to the charged sphere shields the electric field before so the Neumann condition is fulfilled. As we will see, applied to streamer simulations, this leads to slightly lower values of the electric field in the streamer head and therefore less ionization.

3.2. Order of accuracy

We have checked that the method described above does not change the order of accuracy of the discretization of the Poisson equation by constructing a closed-form solution of the Poisson equation that satisfies homogeneous Dirichlet boundary conditions in the upper and lower electrodes. We have used the potential

$$\phi = \sin\left(\pi\frac{z}{L}\right)e^{-\frac{r^2}{\sigma^2}-\frac{(z-z_0)^2}{\sigma^2}}, \tag{17}$$

whose Laplacian has the form

$$\Delta\phi = \frac{1}{L^2\sigma^4}\left\{4L\pi\sigma^2(z-z_0)\cos\left(\pi\frac{z}{L}\right) + \left[\pi^2\sigma^4+L^2\left(-4r^2+6\sigma^2-4(z-z_0)^2\right)\right]\right\} \times \sin\left(\pi\frac{z}{L}\right)e^{-\frac{r^2}{\sigma^2}-\frac{(z-z_0)^2}{\sigma^2}}. \tag{18}$$

Although this charge density is not strictly bounded, the contribution of charges excluded from the domain decays super-exponentially as the domain becomes wider and can thus be neglected as long as the external radius of the computational domain is significantly longer than σ.

We have solved the Poisson equation corresponding to the Laplacian (18) with $L = 1$ m, $\sigma = 0.1$ m and $z_0 = 0.5$ m within a cylindrical domain with a radius $R = 0.5$ m, where we imposed free boundary conditions with the method described above. In this manner we checked that the convergence in the ℓ^2-norm is of second order, the same as that of the finite difference scheme. This is as expected because $\tilde{\phi}_1$, its derivative in the radial direction and the Fourier coefficients (15) retain convergence of order $\mathcal{O}\left(h^2\right)$.

We are also interested in the convergence as we move the outer boundary. Following the example of the previous section, this time we change the radius of the sphere to 0.1 mm and the mesh spacing to 1 μm. Errors are presented in Table 1, and the convergence is as expected of second order. Therefore, the decomposition method does not cause errors of order less than two.

3.3. Sample implementation

A computer code that produces a figure similar to Fig. 2 is included with this work. The code is implemented in Python and to be executed it requires only the widely available scientific libraries NumPy and SciPy. To solve the discrete Poisson equation the code

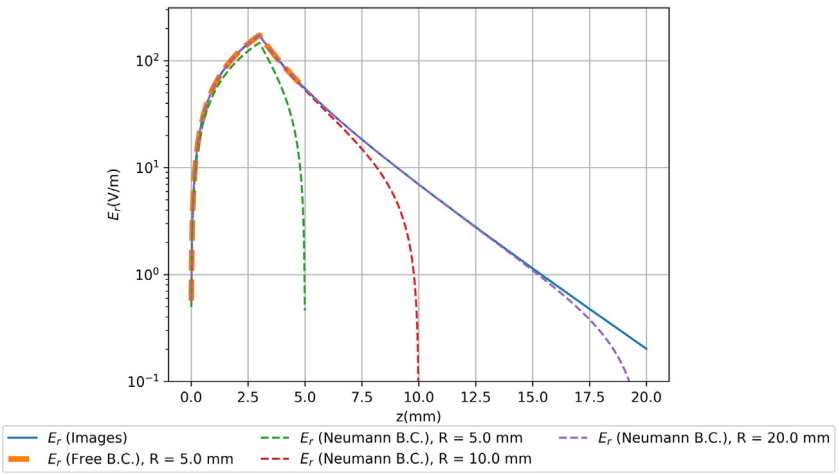

Fig. 2. Comparison between electric fields created by a uniformly charged sphere between two planar infinite electrodes calculated by the approach presented in this work, by the method of images and by imposing homogeneous Neumann boundary conditions at the external boundary of the computational domain. Note the overlap between the lines corresponding to free boundary conditions and to the method of images.

Table 1
Error obtained with change in outer radius.

Outer radius (mm)	$\|\epsilon\|_2$	$\frac{\|\epsilon\|_2}{\|\phi_{exact}\|_2}$
0.2	5.919×10^{-7}	5.037×10^{-6}
0.5	4.735×10^{-8}	3.812×10^{-7}
1	1.227×10^{-8}	9.838×10^{-8}

constructs a sparse matrix for the discrete Laplacian operator and invokes UMFPACK [20] (via SciPy) to solve the resulting linear system.

The code consists of a single file `Poisson_sparse_fft.py` and contains the following methods:

`compute_matrix`: Calculates the sparse matrix for the discrete Laplacian operator in a given cartesian grid and boundary conditions.

`apply_inhom_bc`: Modifies the right-hand-side of the linear system in order to apply inhomogeneous Dirichlet boundary conditions.

`DDM`: Applies the domain decomposition method described in Section 2 to solve the Poisson equation with free boundary conditions.

`MOI`: Calculates the electrostatic potential by means of the method of images, using (16).

`main`: This is the entry-point of the code: it uses the above methods to produce the output figure.

4. Streamer simulations

In elongated electric discharges, Neumann boundary conditions are often considered more appropriate than Dirichlet to be applied at r_{max} because there is not a physical electrode in the radial direction, and therefore there is no reason to keep constant the potential there. The development of electric discharges is driven by long range interactions and therefore boundary conditions certainly affect the solution inside the computational domain. These effects can be reduced by enlarging the domain in the radial direction in exchange of a higher computational cost. The procedure we have described allows us to keep the boundary r_{max} close to the core of the simulation without noticeable numerical effects on the electric discharge. The following simulations clearly illustrate the features mentioned.

We simulated the propagation of streamer discharges between two planar electrodes with a model that includes electron drift, impact ionization and dissociative attachment and is implemented using the CLAWPACK/PetClaw library [21,22]. The Poisson equation is solved using the Improved Stabilized version of BiConjugate Gradient solver from the PETSc numerical library [23,24]. For more details about the physical model see e.g. Refs. [4,25,26].

We selected an inter-electrode gap of $L = 2$ cm and a background electric field of 27 kV/cm. The streamer is initiated by a neutral ionization seed attached to the electrode on the central axis and centered at $z = L$. The peak electron density in this seed is 10^{14} cm^{-3} and the e-folding length is 0.7 mm.

As we are interested in the effect of the external boundary conditions, we run simulations both with free boundary conditions, implemented as described above, and with homogeneous Neumann conditions for the electrostatic potential (as mentioned above, it is generally assumed that Neumann boundary conditions introduce slightly smaller artifacts). We also use different radii of the computational domain, $R = 0.5$ cm, $R = 1$ cm and $R = 2$ cm. In Fig. 3 we show snapshots of the electric field resulting from these simulations at time $t = 30$ ns, shortly after the streamer branches in the simulations with free boundary conditions. Note however that the cylindrical symmetry of the simulations prevents proper branching. MovieS1 (available online) shows the complete evolution of the streamers.

In the plots we see that the simulations with free boundary conditions are essentially identical regardless of the lateral extension of the computational domain. The simulations with homogeneous Neumann conditions on the other hand depend artificially on the radius of the computational domain. The streamer barely develops with $R = 0.5$ cm and only the simulation with $R = 2$ cm reproduces accurately the branching time of the simulations with free boundary conditions. We conclude that, even in this case where the aspect ratio of the discharge is not extremely high, the computational gain from reducing the domain size (roughly a factor 4) more than compensates for the cost of solving twice the Poisson equation, resulting in an overall improvement of about a factor 2.

5. Discussion and conclusions

When they are not laterally constrained, most electrical discharges develop as elongated channels. Despite different physical conditions and ionization mechanisms this feature is common to streamers, leaders and arcs. The underlying reason for this shared property is that all these processes are affected by a Laplacian instability [27,28], whereby small bumps in a discharge front enhance

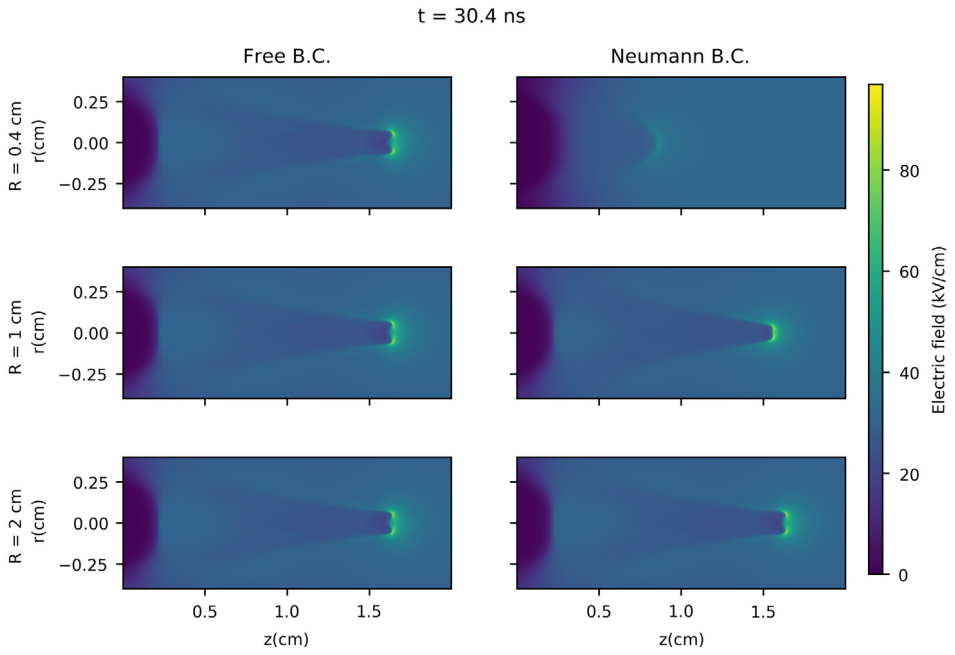

Fig. 3. Streamer simulations using free boundary conditions (left column) and homogeneous Neumann boundary conditions (right column) in the external boundary of the computational domain. For each selection of the boundary conditions we show three simulations where the computational domain extends to a radius $R = 0.5$ cm (top), $R = 1$ cm (middle) and $R = 2$ cm (bottom).

the electric field ahead and thus grow faster than the surrounding regions. This prevents the formation of wide, smooth discharges and creates branched discharge trees of many filaments [29].

Since this is the preferred shape of a discharge, it is reasonable to optimize our numerical models for elongated channels, selecting high-aspect-ratio computational domains. The method and the code that we have presented here can be used to achieve this efficiently and without losing accuracy.

We mention several possible extensions and refinements of this method. The first one, which is described in the appendix, consists in extending the free boundary conditions also to the upper and lower simulation boundaries. This can be useful for the investigation of discharges not attached to any electrode or, with appropriate modifications, attached to a single electrode.

A second extension is to adapt the method to run in parallel in several processors. If we parallelize the Poisson solver by vertically decomposing the domain, the application of the method described above requires collecting information about the initial solution around the external boundary and then performing a one-dimensional Fourier transform. The overhead of these steps is small compared to the operations required for the solution of the Poisson equation so the method can be efficiently parallelized.

Finally, one may ask about the suitability of this method for non-uniform meshes and, in particular, for adaptively refined meshes. Although the application of the method is in principle straightforward, a careful analysis is required to understand the error incurred due to a possibly coarser resolution around the boundary than around a localized charge density. This analysis, however, falls out of the scope of the present paper.

Note that although we have focused on the solution of the Poisson equation, this method can be easily generalized to other elliptic partial differential equations. Then, this can be applied to other components of streamer simulation codes such as the speeding-up of photoionization calculations by approximating the interaction integral by combining solutions of a set of partial differential equations, as proposed in Refs. [11,30,31].

Acknowledgment

This work was supported by the European Research Council (ERC) under the European Union H2020 programme/ERC grant agreement 681257.

A. Full free boundary conditions

The focus in this paper is the implementation of free boundary conditions in the outer boundary of a discharge confined between two parallel plates but the method described above can also be extended to implement free boundary conditions in all boundaries of a simulation with cylindrical symmetry. In this appendix we describe this extension.

A.1. Domain decomposition

The method described above allows us to solve the Poisson equation in the space between two infinite, parallel planes. To build upon this procedure we decompose now the full-space domain Ω into three disjoint subdomains, which we name Ω_0, Ω_1 and Ω_2 (see Fig. A.4). We assume now that the support of the charge distribution f is contained in Ω_0 and thus arrive at the three coupled problems

$$\begin{aligned} &\Delta\phi_0 = f \quad \text{in} \qquad \Omega_0, \\ &\phi_0 = \phi_{\Gamma_{0i}} \ \text{on} \ \partial\Gamma_{0i}, i = 1, 2, \end{aligned} \tag{A.1a}$$

and

$$\begin{aligned} &\Delta\phi_i = f \quad \text{in} \quad \Omega_i, \\ &\phi_i = \phi_{\Gamma_{0i}} \ \text{on} \ \partial\Gamma_{0i}, \end{aligned} \tag{A.1b}$$

where $i = 1, 2$ and $\Gamma_{0i} = \partial\Omega_0 \cap \partial\Omega_i$ are the surfaces $z = 0$ and $z = L$ respectively.

Since there are two interfaces, there are also two conditions for the continuity of the normal derivative:

$$\frac{\partial\phi_0}{\partial z} = \frac{\partial\phi_i}{\partial z} \ \text{on} \ \Gamma_i, i = 1, 2. \tag{A.2}$$

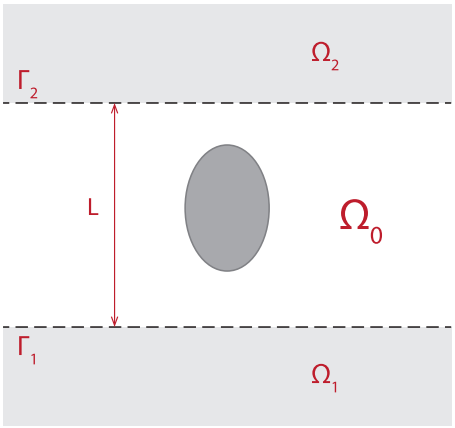

Fig. A.4. Geometry of the problem. Ω is divided into three subdomains: Ω_0, Ω_1 and Ω_2. The three extend indefinitely outwards and the latter two also downwards and upwards respectively.

A.2. Linearity

Linearity allows us to split the problem into

$$\begin{aligned} &\Delta\bar{\phi}_0 = 0 \quad \text{in} \qquad \Omega_0, \\ &\bar{\phi}_0 = \phi_{\Gamma_{0i}} \ \text{on} \ \partial\Gamma_{0i}, i = 1, 2, \end{aligned} \tag{A.3a}$$

$$\begin{aligned} &\Delta\tilde{\phi}_0 = f \quad \text{in} \qquad \Omega_0, \\ &\tilde{\phi}_0 = 0 \quad \text{on} \ \partial\Gamma_{0i}, i = 1, 2 \end{aligned} \tag{A.3b}$$

and for $i = 1, 2$,

$$\begin{aligned} &\Delta\bar{\phi}_i = 0 \quad \text{in} \quad \Omega_i, \\ &\bar{\phi}_i = \phi_{\Gamma_{0i}} \ \text{on} \ \partial\Gamma_{0i}, \end{aligned} \tag{A.4a}$$

$$\begin{aligned} &\Delta\tilde{\phi}_i = f \quad \text{in} \quad \Omega_i, \\ &\tilde{\phi}_i = 0 \quad \text{on} \ \partial\Gamma_{0i}. \end{aligned} \tag{A.4b}$$

Since there is no charge outside the computational domain, $\tilde{\phi}_i = 0$ for $i = 1, 2$. These equations are naturally subject to the condition that the potential vanishes at $-z, z, \rho \to \infty$.

Note now that the problem (A.3b) can be solved by the procedure described in the main text, since $\tilde{\phi}_0$ is the union of the solutions to the problems at (3).

In terms of these components, the flux equation (4) can be expressed as

$$\frac{\partial\bar{\phi}_0}{\partial z}\left[\phi_{\Gamma_{0i}}\right] - \frac{\partial\bar{\phi}_i}{\partial z}\left[\phi_{\Gamma_{0i}}\right] = -\frac{\partial\tilde{\phi}_0}{\partial z}\left[f\right] \ \text{on} \ \Gamma, \ \forall i. \tag{A.5}$$

A.3. Expansion in solutions of the Laplace equation

The potentials $\bar{\phi}_i$ in (A.4a) are solutions of the Laplace equation in cylindrical coordinates and they can be expanded using a set of solutions (see e.g. [19]). Since the domain is unbounded in the radial direction, instead of a series expansion we obtain an integral transform, which we can write in terms of the zero-order Hankel transform, which reads:

$$\bar{\phi}_0 = \int_0^\infty k\,\mathrm{d}k\left[A\left(k\right)e^{kz} + B\left(k\right)e^{-kz}\right]J_0\left(k\rho\right), \tag{A.6a}$$

$$\bar{\phi}_1 = \int_0^\infty k\,\mathrm{d}k\,C\left(k\right)e^{kz}J_0\left(k\rho\right), \tag{A.6b}$$

$$\bar{\phi}_2 = \int_0^\infty k\,\mathrm{d}k\,D\left(k\right)e^{-kz}J_0\left(k\rho\right), \tag{A.6c}$$

where $J_0\left(x\right)$ is the zero-order Bessel function of the first kind and the functions A, B, C, D weight the independent solutions to the Laplace equation. Note that, although the factor k can be absorbed into these functions, it appears explicitly in order to show the Hankel transform structure.

The function $\phi_{\Gamma_{0i}}$ can also be transformed as:

$$\phi_{\Gamma_{0i}}\left(\rho\right) = \int_0^\infty k\,\mathrm{d}k E_i\left(k\right)J_0\left(k\rho\right). \tag{A.7}$$

Casting these equations in the form of a Hankel transform is important because as the Hankel transform can be inverted (it is its own inverse) we can use the fact that, subject to some regularity assumptions,

$$\int_0^\infty k\,\mathrm{d}k\,F\left(k\right)J_0\left(k\rho\right) = 0 \iff F(k) = 0. \tag{A.8}$$

A.4. Continuity of the normal derivative

Imposing that $\bar{\phi}_0 = \bar{\phi}_i = \phi_{\Gamma_{0i}}$ at $z = 0$ and L, we solve for A, B, C, D using (A.8) and write (A.6) as

$$\begin{aligned} \bar{\phi}_0 = \int_0^\infty \frac{k\,\mathrm{d}k}{e^{2kL}-1}\Big[&\left(E_2e^{kL} - E_1\right)e^{kz} \\ &+ \left(-E_2 + E_1e^{kL}\right)e^{-k(z-L)}\Big]J_0\left(k\rho\right), \end{aligned} \tag{A.9a}$$

$$\bar{\phi}_1 = \int_0^\infty k\,\mathrm{d}k\,E_1e^{kz}J_0\left(k\rho\right), \tag{A.9b}$$

$$\bar{\phi}_2 = \int_0^\infty k\,\mathrm{d}k\,E_2e^{-k(z-L)}J_0\left(k\rho\right). \tag{A.9c}$$

Using these expressions into the equation for the normal derivative (A.5) we obtain

$$\int_0^\infty \frac{2e^{kL}k^2\,\mathrm{d}k}{e^{2kL}-1}\left(E_2 - E_1e^{kL}\right)J_0\left(k\rho\right) = -\left.\frac{\partial\tilde{\phi}_0}{\partial z}\right|_{z=0}, \tag{A.10a}$$

$$\int_0^\infty \frac{e^{kL}k^2\,\mathrm{d}k}{e^{2kL}-1}\left(E_2e^{kL} - E_1\right)J_0\left(k\rho\right) = -\left.\frac{\partial\tilde{\phi}_0}{\partial z}\right|_{z=L}. \tag{A.10b}$$

We can obtain the coefficients E_1 and E_2 going back to the k-space by means of the Hankel transform:

$$E_1\left(k\right) = \frac{1}{2}\left(e^{-kL}I_L - I_0\right), \tag{A.11a}$$

$$E_2\left(k\right) = \frac{1}{2}\left(I_L - e^{-kL}I_0\right), \tag{A.11b}$$

where

$$I_{0,L}\left(k\right) = -\frac{1}{k}\int_0^\infty \rho\,\mathrm{d}\rho\left.\frac{\partial\tilde{\phi}_0}{\partial z}\right|_{z=0,L}J_0\left(k\rho\right). \tag{A.12}$$

In this expression $\tilde{\phi}_0$ and its normal derivative are known from the algorithm described in 2.5. Therefore we can compute $I_{0,L}$ and, using (A.11) $E_{1,2}$. These functions in turn can be inserted in (A.7) to yield the boundary condition to impose on $\Gamma_{1,2}$.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cpc.2018.01.003.

References

[1] J.R. Dwyer, M.A. Uman, Phys. Rep. 534 (2014) 147. http://dx.doi.org/10.1016/j.physrep.2013.09.004.

[2] U. Ebert, C. Montijn, T.M.P. Briels, W. Hundsdorfer, B. Meulenbroek, A. Rocco, E.M. van Veldhuizen, Plasma Sour. Sci. Technol. 15 (2006) S118. http://dx.doi.org/10.1088/0963-0252/15/2/S14. arXiv:physics/0604023.

[3] N. Liu, V.P. Pasko, J. Phys. D 39 (2006) 327. http://dx.doi.org/10.1088/0022-3727/39/2/013.
[4] A. Luque, U. Ebert, J. Comput. Phys. 231 (2012) 904. http://dx.doi.org/10.1016/j.jcp.2011.04.019.
[5] N. Liu, J.R. Dwyer, H.C. Stenbaek-Nielsen, M.G. McHarg, Nature Commun. 6 (2015) 7540. http://dx.doi.org/10.1038/ncomms8540.
[6] J. Qin, V.P. Pasko, Geophys. Res. Lett. 42 (2015) 2031. http://dx.doi.org/10.1002/2015GL063269.
[7] J. Teunissen, U. Ebert, Afivo: a framework for quadtree/octree AMR with shared-memory parallelization and geometric multigrid methods, arXiv:1701.04329.
[8] N.Y. Babaeva, G.V. Naidis, J. Phys. D 29 (1996) 2423. http://dx.doi.org/10.1088/0022-3727/29/9/029.
[9] N.Y. Babaeva, G.V. Naidis, in: E.M. van Veldhuizen (Ed.), Electrical Discharges for Environmental Purposes: Fundamentals and Applications, Nova Science, New York, United States, 2000.
[10] N. Liu, V.P. Pasko, J. Geophys. Res. (Space Phys) 109 (2004) A04301. http://dx.doi.org/10.1029/2003JA010064.
[11] A. Bourdon, V.P. Pasko, N.Y. Liu, S. Célestin, P. Ségur, E. Marode, Plasma Sour. Sci. Technol. 16 (2007) 656. http://dx.doi.org/10.1088/0963-0252/16/3/026.
[12] C.R. Anderson, J. Comput. Phys. 62 (1986) 111. http://dx.doi.org/10.1016/0021-9991(86)90102-6.
[13] R.J. James, J. Comput. Phys. 25 (1977) 71–93. http://dx.doi.org/10.1016/0021-9991(77)90013-4.
[14] Gregory T. Balls, Phillip Colella, J. Comput. Phys. 180 (2002) 25–53. http://dx.doi.org/10.1006/jcph.2002.7068.
[15] Mads Mølholm Hejlesen, Johannes Tophøj Rasmussen, Philippe Chatelain Jens Honoré Walther, J. Comput. Phys. 252 (1) (2013) 458–467.
[16] Christopher R. Anderson, J. Comput. Phys. 314 (2016) 194–205. http://dx.doi.org/10.1016/j.jcp.2016.02.074.
[17] C.R. Anderson, in: T. Chan, R. Glowinski, J. Périaux, O. Widlund (Eds.), Domain Decomposition Methods, Proceedings in Applied Mathematics, SIAM, 1989 http://www.ddm.org/DD02/Proc-2.php.
[18] A. Bayliss, M. Gunzburger, E. Turkel, SIAM J. Appl. Math. 42 (2) (1982) 430–451. http://dx.doi.org/10.1137/0142032.
[19] J. Jackson, Classical Electrodynamics, John Wiley and sons, New York, USA, 1975.
[20] T.A. Davis, ACM Trans. Math. Software 30 (2) (2004) 196–199. http://dx.doi.org/10.1145/992200.992206.
[21] R. LeVeque, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
[22] A. Alghamdi, A. Ahmadia, D.I. Ketcheson, M.G. Knepley, K.T. Mandli, L. Dalcin, Proceedings of the 19th High Performance Computing Symposia, HPC '11, Society for Computer Simulation International, San Diego, CA, USA, 2011, pp. 96–103.
[23] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Web page, 2016, http://www.mcs.anl.gov/petsc.
[24] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.7, Argonne National Laboratory 2016, http://www.mcs.anl.gov/petsc.
[25] C. Montijn, W. Hundsdorfer, U. Ebert, J. Comput. Phys. 219 (2006) 801. http://dx.doi.org/10.1016/j.jcp.2006.04.017. arXiv:physics/0603070.
[26] U. Ebert, S. Nijdam, C. Li, A. Luque, T. Briels, E. van Veldhuizen, J. Geophys. Res. (Space Phys) 115 (2010) A00E43. http://dx.doi.org/10.1029/2009JA014867. arXiv:1002.0070.
[27] M. Arrayás, U. Ebert, W. Hundsdorfer, Phys. Rev. Lett. 88 (17) (2002) 174502. http://dx.doi.org/10.1103/PhysRevLett.88.174502. arXiv:nlin/0111043.
[28] G. Derks, U. Ebert, B. Meulenbroek, J. Nonlinear Sci. 18 (2008) 551. http://dx.doi.org/10.1007/s00332-008-9023-0.
[29] A. Luque, U. Ebert, New J. Phys. 16 (1) (2014) 013039. http://dx.doi.org/10.1088/1367-2630/16/1/013039. arXiv:1307.2378.
[30] P. Ségur, A. Bourdon, E. Marode, D. Bessieres, J.H. Paillol, Plasma Sour. Sci. Technol. 15 (2006) 648. http://dx.doi.org/10.1088/0963-0252/15/4/009.
[31] A. Luque, U. Ebert, C. Montijn, W. Hundsdorfer, Appl. Phys. Lett. 90 (8) (2007) 081501. http://dx.doi.org/10.1063/1.2435934. arXiv:physics/0609247.