Czech Technical University in Prague
Faculty of Mechanical Engineering

Université Paris 13
Institut Galilée, LAGA

Dissertation Thesis

Numerical Simulation of Streamer Propagation on Unstructured Dynamically Adapted Grids

Ing. Jan Karel

Doctoral Study Programme:
Mathematical and Physical Engineering

Field: *Applied Mathematics*
Date: *December 2, 2014*

Supervisors:
Prof. Ing. Jaroslav Fořt, CSc.
Prof. Fayssal Benkhaldoun

Jury:
Fayssal Benkhaldoun, PU, Université Paris 13, France (Supervisor)
Anne Bourdon, DR, Ecole Centrale de Paris, France (Reviewer)
Jaroslav Fort, PU, FME, CTU Prague, Czech Republic, (Supervisor)
Jiri Furst, HDR, FME, CTU Prague, CTU Prague, Czech Republic
Jan Halama, HDR, FME, CTU Prague, Czech Republic, Czech Republic
Laurence Halpern, PU, Université Paris 13, France
Philippe Helluy, PU, Université de Strasbourg, France (Reviewer)
Richard Liska, PU, FNSPE, CTU Prague, Czech Republic

2013 Prague

Acknowledgments

I would like to thank to the supervisors Professor Ing. Jaroslav Fořt, CSc. (Czech Technical University in Prague, Faculty of Mechanical Engineering) and Professor Fayssal Benkhaldoun (Université Paris 13, LAGA). I also would like to thank Professor Laurence Halpern for welcoming me at LAGA and Professor Khaled Hassouni (Université Paris 13, LSPM) and Doc. Ing. Jiří Fürst, PhD. (CTU in Prague, Faculty of Mechanical Engineering) for valuable advices and for participation in the jury. I also would like to thank Professor Philippe Helluy, Professor Richard Liska and Director of Research at CNRS Anne Bourdon for they acceptation to be reviewers of the work and also for their participation in the jury. And finaly I also would like to thank to Doc. Ing. Jan Halama, PhD for his participation in the jury.

The work has been partly supported by the Marie Curie Actions of the European Commission in the frame of the DEASE project (MEST-CT-2005-021122) and project of the Grant of Czech Technical University in Prague No SGS10/244/OHK2/3T/12.

Název práce: **Numerická Simulace Propagace Streameru na Dynamicky Adaptovaných Sítích**

Anotace

Tato dizertační práce se zabývá numerickou simulací propagace streameru (elektrický výboj ve vysokonapěťovém elektrickém poli). Pro popis streameru je použit minimální model, který se skládá ze soustavy transportních rovnic pro elektricky nabité částice spárovaných s Poissonovou rovnicí pro elektrický potenciál. V práci simulujeme obecný pohyb streameru ve 3D. Tento obecný pohyb je prezentován rozvětvením streameru, kterého se dosáhne pomocí lokálních poruch v elektrickém poli.

Pro numerickou simulaci streameru jsme vyvinuli metodu založenou na dynamické adaptaci sítě, jejíž vlastnosti byly otestovány na jednodušších problé-mech ve 2D (menší časová náročnost). I když jde o jiný typ problému (rovinný výboj), pro vývoj metody je dostatečný a umožňuje snadný přechod do 3D.

Klíčová slova: numerická simulace, pohyb streameru, rozvětvení streameru, minimální model, dynamická adaptace sítě

Title: **Numerical Simulation of Streamer Propagation on Unstructured Dynamically Adapted Grids**

Abstract

The aim of this thesis is a numerical simulation of a streamer propagation (electric discharge in a high voltage electric field). The minimal model is used for the streamer description. The model consists of a system of convection-diffusion-reaction equations for electric particles coupled with Poisson's equation for an electric potential. We simulate a general streamer motion in 3D which is presented by streamer branching. It is caused by local disturbances in the electric field.

We have developed a method based on a dynamically adaptation of grids for the simulation. The properties of the method are tested on simpler problems in 2D (less time consuming). This approach is sufficient for the development of the method even if it is different type of problem (planar discharge) and it allows a simple transition to 3D.

Key words: numerical simulation, streamer motion, streamer branching, minimal model, dynamical grid adaptation

Contents

List of Symbols

A	[cm^{-2}Torr^{-2}]	fitting parameter
D_e	[cm^2·s^{-1}]	diffusion coefficient
$D_{\sigma_{ij}}$		diamond cell
$\vec{E}$	[Vcm^{-1}]	intensity of electric field
I		ion emission
N	[cm^{-3}]	neutral gas density
n_e	[cm^{-3}]	electron density
n_i (n_p)	[cm^{-3}]	positive ion density
n_n	[cm^{-3}]	negative ion density
$\dfrac{p_0}{p}$		ratio of the atmospheric pressure to a gas pressure
q	[C]	elementary charge
$\vec{v}_e$	[cm· s^{-1}]	electron velocity
$\vec{v}_p$	[cm· s^{-1}]	positive ion velocity
$\vec{v}_n$	[cm· s^{-1}]	negative ion velocity
V	$[V]$	electric potential
t	[s]	time
$\triangle t$	[s]	time step
T_i		mesh cell
$x,\ y,\ y$	[cm]	coordinates
α	[cm^{-1}]	ionization coefficient
ϵ_0	[Fm^{-1}]	permittivity
β_{ep}	[cm^3·s^{-1}]	electron-positive ion recombination coefficient
β_{np}	[cm^3·s^{-1}]	negative-positive ion recombination coefficient
λ	[cm^{-1}Torr^{-1}]	fitting parameter
η_2	[cm^{-1}]	two-body attachment coefficient
η_3	[cm^{-1}]	three-body attachment coefficient
μ_i	[cm^2], [cm^3]	volume of subdomain
ψ		Barth-Jespersen limiter
σ_{ij}		face of a cell T_i
Ω		domain
$\mathfrak{n}$		unit normal vector
FVM		finite volume method
FEM		finite element method

Chapter 1

Introduction

When a neutral gas is exposed to high intensity electric field, non-equilibrium ionization processes (discharges) occur. The ionization causes that neutral particles are decomposed to ions and free electrons. Meanwhile the gas temperature remains close to ambient temperature, the electron temperature increases to a high level ($T_e > 10000$ K). The discharges can appear in various forms depending on spatiotemporal conditions such as the intensity of the electric field, volume and pressure of the medium etc. One can distinguish stationary and unsteady discharges. The dark, glow or ark discharges belong to the stationary ones. Leaders and streamers are transient unsteady discharges. We are interested in streamers that are filamentary discharges which dynamics are controlled by highly localized and nonlinear space charge regions. The streamers can be used in several **technical applications**, e.g.:
streamers are used for treatment of contaminated media like exhaust gases, polluted water or bio gas because the streamers emit the reactive radicals . The streamers are also used in the processing of electronics, especially for etching and deposition of thin films. They can be applied in surface modification (e.g. hardening, corrosion resistance) too.
The atmospheric pressure plasma jets belong among other utilizations of streamers. They can be used for biomedical applications because they can produce various biocidal agents such as UV radiation, reactive species and charged particles. The atmospheric plasma jets can be easily handled and can treat a target from a distance without thermal damage. Therefore they have a potential to be used in diverse fields of biomedicine.
Another utilization of the weakly ionized gas is to control a boundary layer in airflow at high speeds because achievements in aerospace allow to create hypersonic air vehicles at a Mach number greater than 6. The traditional flow control mechanism is often ineffective at such high velocities due to significant physical-chemical modification of the airflow.
The streamers can be undesirable too. They can cause e.g. transmission line outgoes at power system when a streamer appears in a disturbed electric field around an insulator string. The interest is how to protect the insulator string from the streamer.

The generation of streamer discharges is described by coupling of electrostatic with motion of charged particles (electrons, positive and negative ions). The electrostatic is represented by Poisson's equation for the electric potential and the particles motion is described by a set of convection-diffusion-reaction equations. The charged particles are considered as continuum. It means that we deal with the particles in a form of densities and we can use continuum mechanics for them. The motion of charged particles causes a local nonlinearity in the electric field which is an essential element for creation of the streamer discharge. Another important phenomenon in the streamer motion is photoionization which decompose neutral particles to charged ones and it also locally changes the electric field. The photoionization effect is very often described by Helmholtz or Eddington models or it is replaced by background densities of the electrons and the positive ions in many studies. The charged particles react with surrounding environment in chemical reaction during the streamer propagation. It produce chemically active species which are interesting for many applications.

Governing equations

The streamer motion can be describe by different models which differ in a complexity. The streamer propagation is essentially formulated in a form of fluid models that couple unsteady transport equations for charged species and Poisson's equation for the electric potential.

One of these approaches is two-fluid Euler-Poisson system which is considered e.g. in [6]. This approach consist of a continuity equation and a convection-diffusion-reaction equation for the electrons and the positive ions (one pair for each specie)

$$\begin{aligned}
&(n_i)_t + \nabla \cdot (n_i u_i) = 0,\\
&m_i \left[(n_i u_i)_t + \nabla \cdot (n_i u_i \otimes u_i) \right] + \nabla p_i \left(n_i \right) = q n_i E,\\
&(n_e)_t + \nabla \cdot (n_e u_e) = 0,\\
&m_e \left[(n_e u_e)_t + \nabla \cdot (n_e u_e \otimes u_e) \right] + \nabla p_e \left(n_i \right) = -q n_e E,\\
&\nabla^2 V = -\frac{q}{\epsilon_0} \left(n_i - n_e \right), \quad \vec{E} = -grad(V),
\end{aligned}$$

where $n_{i,e}$ denotes positive ion (electron) density, $u_{i,e}$ is its velocity. V is the electric potential, $\vec{E}$ is the intensity of the electric field q is the elementary charge and ϵ_0 is the dielectric constant. $p_{i,e}$ denotes particle pressure of ions (electrons).

Another approach, which is considered in many papers, deal with three convection-diffusion-reaction equations for charged species coupled with Poisson's equation for the electric potential

$$\begin{aligned}
&\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e \vec{v}_e) - \nabla \cdot (D_e \nabla n_e) = S_{ph} + n_e \alpha |\vec{v}_e| - n_e \beta_{att} |\vec{v}_e| - n_e n_p \beta_{ep},\\
&\frac{\partial n_p}{\partial t} + \nabla \cdot (n_p \vec{v}_p) = S_{ph} + n_e \alpha |\vec{v}_e| - n_e n_p \beta_{ep} - n_n n_p \beta_{np},\\
&\frac{\partial n_n}{\partial t} + \nabla \cdot (n_n \vec{v}_n) = n_e \beta_{att} |\vec{v}_e| - n_n n_p \beta_{np},\\
&\nabla^2 V = -\frac{q}{\epsilon_0} \left(n_p - n_n - n_e \right), \quad \vec{E} = -grad(V),\\
&\nabla^2 S_{ph} - \left(\lambda p_{O_2} \right)^2 S_{ph} = -A p_{O_2}^2 I,
\end{aligned}$$

where $n_e,\ n_p,\ n_n$ are electron, positive and negative ion densities, $\vec{v}_e,\ \vec{v}_p,\ \vec{v}_n$ are drift velocities of the particles and D_e is a diffusion coefficient. V is the electric potential and $\vec{E}$ denote the intensity of the electric field. The source terms on the right hand side of convection-diffusion-reaction equations represent ionization (terms multiplied by a constant α), attachment (terms multiplied by a constant β_{att}), electron-positive ion recombination (β_{ep}) and negative-positive ion recombination (β_{np}). The source term S_{ph} denotes the photoionization and is expressed by a Helmholtz equation (see above). A and λ in the Helmholtz equation are fitting parameters and we can find them e.g. in [16]. p_{O^2} is a partial pressure of the oxygen and I represents photon emission which can be described by relation $I = A \cdot n_e \cdot \vec{v}_e \cdot \lambda$. We can see the precise expression for drift velocities, diffusion coefficient and coefficients in source terms in relations (B.1) - (B.8).

There is the simplest approach, so-called **minimal model**, among the approaches. It's the simplest model which is able to capture the streamer propagation and consists of a convection-diffusion-reaction equation for the electron density and an ordinary differential equation for the positive ion density. These two equations are coupled with the Poisson's equation for the electric potential [7], [8]

$$
\begin{aligned}
&\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e \vec{v}_e) - \nabla \cdot (D_e \nabla n_e) = n_e \alpha |\vec{v}_e|, \\
&\frac{\partial n_p}{\partial t} = n_e \alpha |\vec{v}_e|, \\
&\nabla^2 V = -\frac{q}{\epsilon_0} (n_p - n_e), \quad \vec{E} = -grad(V).
\end{aligned}
$$

The motion of the positive ions is neglected in the minimal model because the velocity of the electrons is much higher than the positive ion velocity. Only the ionization is considered in source term and the photoionization is replaced by background densities for electrons and positive ions.

All approaches have the same relation between transport equations of charged particles and the electric field. The electric potential depends on charged particles because their densities have an influence on the right hand side of Poisson's equation. On the other hand the material properties (drift velocities, diffusion coefficient etc.) depend on the intensity of the electric field (see appendix B) which is a negative gradient of the electric potential. It leads to the coupled system of equations which can't be solved separately.

Numerical simulation

Because of the streamer nature which is a filament of a plasma, the first attempts how to simulate streamer motion were 1D simulations where the 3D nature of the electric potential (Poisson's equation) was solved by a disk method. This approach gives sufficient information about distribution of electric field, electron drift velocity, charge etc. along the streamer axis but it isn't able to tells us the distributions in a space. Therefore there is a need to simulate streamer in more dimensions.

A plane to plane electrode system or a pin to plane electrode system is very often considered in streamer simulations. The computational domain is very simple when the plane to plane system is taken into account. It allows to simulate the streamer as a 2D axisymmetric case. The system of governing equations is then written in a cylindrical coordinates in such a case. The computational domain is in this case a rectangle and the initial nonlinearity in the electric field is introduced by distribution of the electron and the positive ion densities. The cartesian grid is suitable for such simple computation domain (see e.g. [7], [8], [9]) and the finite difference method is mainly used. The finite difference method has advantages that the cell vertex schemes are simple and mostly come from Taylor expansion. To avoid the high numerical viscosity in these schemes, they can be easily extended to obtain higher orders of accuracy. The influence of higher orders schemes on streamer motion is studied e.g. in [13] where the first order upwind scheme, the second order MUSCL scheme family and QUICKEST scheme family (3^{rd} order and 5^{th} order) are compared.

When we would like to solve streamer motion in more general geometries, it is better to use unstructured grids. The simplest case of a such geometry is the pin to plane electrode system where the shape of the pin electrode is necessary to capture. The finite volume method (FVM) is better choice for computation on unstructured grids but on the other hand it's difficult to extend these schemes to higher orders of accuracy (higher than the second order). Another possibility is to use the finite element method (FEM) on the unstructured grids. The comparison of FVM and FEM methods is discussed e.g. in [12]. It seems that the computational cost is higher for the FEM than for the FVM. It's necessary to skip to 3D if we want to simulate the streamer motion in more complex geometries (e.g. different shapes of electrodes) or more general motion which is caused by local disturbances in the electric field. The streamer propagation in the plane to plane electrode system computed on 3D cartesian grid is e.g. in [8]. The attempts of general streamer motion (streamer branching), which is introduced by local seeds of plasma spots, are discussed e.g. in [16] and an unstructured mesh is considered at this approach.

Mesh adaptation

The difficulty in a numerical simulation of the streamer propagation comes from a very narrow region where the variables governing the physics of the motion vary in several orders of magnitude. Therefore we need a fine grid in this region. A mesh size would be too large if the mesh will be fine in a whole computational domain and the computational time and memory cost would be too high for a such mesh. A way how to avoid this disadvantage in the simulation is a dynamical mesh adaptation. Almost all of streamer simulations use adapted meshes now. We can see the mesh adaptation on 2D cartesian grids e.g. in [7],[8], [9]. In [7], they treat independently with refinement procedures for continuity equations and for Poisson's equation to capture the streamer head front in a cartesian grid. The 3D mesh adaptation is used e.g. in [9] where the streamer motions is computed in a plane to plane electrode system. The streamer motion is also simulated on 2D unstructured grids where the finite-element method is applied together with using of an adaptive mesh generator [10], [11]. The adaptation criterion deals several variables, ratio of the gradient of the electron density and the magnitude of the electron density$|\text{grad} n_e|/n_e$, the magnitude of the net charge density $\rho = e(n_i - n_e)$ and the magnitude of the ionization source term. Advantage of the mesh adaptation on a structured grid is that we can use relatively simple schemes easily extended to higher orders of accuracy but on the other hand we obtain non-conform grids. The mesh adaptation on unstructured grids gives conform grids because a conformity step is applied after cells refinement. The mesh adaptation is especially necessary in the 3D computation where the mesh size could oversize several millions of cells. That leads to high memory cost and the computation takes several weeks to obtain results.

There are still lots of challenges in the streamer simulation regardless of the great progress in the streamer simulation. Most of the streamer simulations published in the literature take into account the axisymmetric approach and geometries which aren't able to capture the structure observed in practice simulations or in the nature. The full 3D simulation is needed to solve the streamer branching or the streamer propagation in complex geometries.

1.1 Goals of the Thesis

The main aim of the thesis is to simulate the streamer branching. Therefore we have to develop fully 3D method with dynamical grid adaptation. The physical model can be simplified to the minimal model of the streamer. The unstructured grid is a proper choice even with the disadvantage of lower order schemes to simulate a general motion of the streamer.
3D computation is very time consuming. Numerical method have to be developed and tested on simplified cases. The proper intermediate step is a 2D case (planar discharge) even if it's a different type of problem than the filamentary discharge. We want to test properties of the scheme (numerical tests of convergence), the adaptation algorithm and proper formulation of conditions.
Computing code for an unstructured grid needs series of support utilities for description and manipulation with a grid (e.g. description of connectivity). Mainly the algorithm of the grid adaptation needs a special care. Modern computer languages (e.g. C++) offer suitable tools for this purposes. Applicability of such programs exceeds range of the streamer motion simulation. They can be used in many applications of finite volume method for different mathematical models (Euler equations, Navier-Stokes equation, flow in porous media, shallow water equations etc.) and different type of problems (combustion, flow with shock waves, multi-face flow etc.) where the dynamical mesh adaptation can increase the quality of results.

We therefore formulate following goals of the thesis:

- Simulation of streamer branching by introducing temporal disturbances;
- Development of a numerical method for a 3D streamer motion calculation described by the minimal model and based on dynamic adaptation of unstructured grids.

Intermediate steps:

- Development of 2D method (in a form which allows simple transition to 3D) for numerical experiments to verify the properties of the method;
- To develop a general platform based on object programming in C++ language for simulation of various physical problems on unstructured grids with a dynamical mesh adaptation in 2D and 3D.

Chapter 2

Governing equations

The modeling of streamer propagation mechanism in atmospheric pressure discharges received a large attention and was the subject of a large research effort during the last two decades. Until very recently, streamer propagation was essentially described through fluid models that couple non-stationary transport equations for charged species (electron and ions) with a Poissons (elliptic) equation for the electrical potential. Usually the transport equations of charged species consist of time-dependent convection-diffusion-reaction equations. In principle, when using advanced physical model, the source terms in these equations include a non-local component induced by the photoionization effect that was often invoked as the main propagation driver at least for streamer in field-free space. In this case, a radiation transport model has to be used and a non local radiation transfer equation has to be coupled to the transport and Poisson's equations. This results in a very high computational costs and recently an advanced physical model based on Eddinton's approximation allowed a very efficient and physically sound description of the photoionization phenomenon. It remains that very often a background seed electron field was used to simulate the electrons generated by photoionization phenomena and responsible for the avalanches that takes place at the streamer tip and that are responsible for streamer propagation. The photoionization effect is deeply studied e.g. in [15] where several approximation is taken into account. There are also more complex boundary conditions for the Poisson's equation in [15].

The simplest **minimal model** of discharge motion [7], [8] is taken into account in our work . It consists of a convection-diffusion-reaction for the electron density, an ordinary differential equation for the positive ion density coupled by the Poisson's equation for the elecric potential

$$
\begin{aligned}
\frac{\partial n_e}{\partial t} + \operatorname{div}\left(n_e \vec{v}_e - D_e \vec{\nabla} n_e\right) &= S_e, \\
\frac{\partial n_i}{\partial t} &= S_i^+,
\end{aligned}
\tag{2.1}
$$

where t is the time, n_e denotes the electron density, n_i the positive ion density, $\vec{v}_e$ the electron drift velocity, D_e the diffusion coefficient. $S_e = S_i^+$ are source terms. The system (2.1) is coupled with the Poisson's equation for the electric potential V

$$\Delta V = Q, \tag{2.2}$$

where $Q = -\frac{e}{\epsilon}(n_i - n_e)$ (2D tests were computed with $Q = -\frac{e}{\epsilon}\frac{(n_i - n_e)}{10^4}$), ϵ is the dielectric constant, e the electron charge. The intensity of the electric field is computed as a negative gradient of the electric potential

$$\vec{E} = -\vec{\nabla} V. \tag{2.3}$$

The electron drift velocity $\vec{v}_e$ is a function of the electric field $\vec{E}$ and depends on the ratio $||\vec{E}||/N$ where N is the neutral gas density ($N = 2.5 \cdot 10^{19}\ cm^{-3}$) [5]

$$\begin{array}{lll}
\text{for } \dfrac{\|\vec{E}\|}{N} > 2 \cdot 10^{-15}, & \vec{v}_e = - \left[7.4 \cdot 10^{21} \cdot \dfrac{\|\vec{E}\|}{N} + 7.1 \cdot 10^6 \right] \cdot \dfrac{\vec{E}}{\|\vec{E}\|}, \\
\text{for } 10^{-16} < \dfrac{\|\vec{E}\|}{N} \leq 2 \cdot 10^{-15}, & \vec{v}_e = - \left[1.03 \cdot 10^{22} \cdot \dfrac{\|\vec{E}\|}{N} + 1.3 \cdot 10^6 \right] \cdot \dfrac{\vec{E}}{\|\vec{E}\|}, \\
\text{for } 2.6 \cdot 10^{-17} < \dfrac{\|\vec{E}\|}{N} \leq 10^{-16}, & \vec{v}_e = - \left[7.2973 \cdot 10^{21} \cdot \dfrac{\|\vec{E}\|}{N} + 1.63 \cdot 10^6 \right] \cdot \dfrac{\vec{E}}{\|\vec{E}\|}, \\
\text{for } \dfrac{\|\vec{E}\|}{N} \leq 2.6 \cdot 10^{-17}, & \vec{v}_e = - \left[6.87 \cdot 10^{22} \cdot \dfrac{\|\vec{E}\|}{N} + 3.38 \cdot 10^4 \right] \cdot \dfrac{\vec{E}}{\|\vec{E}\|}.
\end{array} \tag{2.4}$$

The diffusion coefficient D_e is a function of the electron drift velocity and the electric field [5]

$$D_e = \left[0.3341 \cdot 10^9 \cdot \left(\frac{\|\vec{E}\|}{N} \right)^{0.54069} \right] \cdot \frac{\|\vec{v}_e\|}{\|\vec{E}\|}. \tag{2.5}$$

The source terms depend on the electron drift velocity and the electron density

$$S_e = \frac{\alpha}{N} \cdot \|\vec{v}_e\| \cdot n_e \cdot N, \tag{2.6}$$

$$S_i^+ = S_e \tag{2.7}$$

where the ratio $\frac{\alpha}{N}$ [cm^2] is computed by following formula

$$\begin{array}{lll}
\text{if } \dfrac{\|\vec{E}\|}{N} > 1.5 \cdot 10^{-15}, & \dfrac{\alpha}{N} = 2 \cdot 10^{-16} \cdot \exp\left(\dfrac{-7.248 \cdot 10^{-15}}{\|\vec{E}\|/N} \right), \\
\text{else,} & \dfrac{\alpha}{N} = 6.619 \cdot 10^{-17} \cdot \exp\left(\dfrac{-5.593 \cdot 10^{-15}}{\|\vec{E}\|/N} \right).
\end{array} \tag{2.8}$$

Initial conditions

The electric field is established between two planar electrodes. High voltage 25000 V is applied at the anode and the cathode is grounded. The disturbance in the electric field between these two planar electrodes is given by an initial Gaussian pulse. The photo-ionization effect, which is necessary to run streamer, is substitute by a background density:

2 dimensions

$$\begin{aligned} &n_e\left(x,y,0\right) = 10^{16}\cdot e^{-\frac{(x-0.2)^2+(y-0.25)^2}{\sigma^2}} + 10^9\ [\mathrm{cm}^{-3}],\quad \sigma = 0.01, \\ &n_i\left(x,y,0\right) = n_e\left(x,y,0\right), \end{aligned} \tag{2.9}$$

3 dimensions

$$\begin{aligned} &n_e\left(x,y,z,0\right) = 10^{12}\cdot e^{-\frac{(x-0.2)^2+(y-0.25)^2}{\sigma^2}} + 10^8\ [\mathrm{cm}^{-3}],\quad \sigma = 0.01, \\ &n_i\left(x,y,z,0\right) = n_e\left(x,y,0\right). \end{aligned} \tag{2.10}$$

Boundary conditions

The Neumann's homogeneous condition is considered at all boundaries for the electron density

$$\frac{\partial n_e}{\partial \vec{n}} = 0. \tag{2.11}$$

Following Dirichlet's conditions are prescribed for the electric potential at the anode and the cathode

$$\begin{aligned} V &= 25000\ [\mathrm{V}], && \text{for anode,} \\ V &= 0\ [\mathrm{V}], && \text{for cathode.} \end{aligned} \tag{2.12}$$

The Neumann's homogeneous condition is prescribed at the rest of boundaries

$$\frac{\partial V}{\partial \vec{n}} = 0. \tag{2.13}$$

Chapter 3

Numerical Approximation

The streamer propagation phenomenon is described through the time-space evolution of physical characteristics such as electron density, ion density, electric field and space charge. All these characteristics experience a several orders of magnitude variation in a very narrow region located at the microdischarge tip (the so called streamer region). Further, the transport equations and the electric field equation are strongly coupled through source-terms, transport fluxes and space charge. The propagation dynamics intimately depends on the space-time distribution of the microdischarge characteristics in this tinny region with a characteristics length of few micrometers. Researchers therefore concentrate on the development of high order schemes, e.g., FCT, QUICK, (W)ENO, MUSCLE, improved Sharfeter-Gummel, etc., for 1D geometries and 2D-axisymmetrical geometries. The use of highly non-symmetrical electrode system, along with the non-planar nature of the streamer propagation motivated some groups to adapt some of these numerical schemes to the case of non-structured grids.
Nevertheless the streamer simulation remains costly and time consuming. That's the reason why a simple upwind scheme extended by a Van Leer's type MUSCL algorithm together with Barth-Jespersen limiter is used in this work. The dynamical mesh adaptation improve the accuracy of the simple scheme. There is possible to use another limiters such as Van Albada limiter (see section 4.1.2). The Barth-Jespersen limiter is chosen because it's a quite simple limiter and doesn't depend on parameters which should be tuned.
As it was mentioned before, streamer propagation is represented by unsteady convection diffusion reaction equations for electric particles coupled with Poisson's equation for electrical potential. The unsteady equations are solved by an explicit scheme on unstructured grid. The discretized Poisson's equation leads to a system of algebraic linear equations which is solved with a direct solver at each time step.

3.1 Numerical Method

The system of convection-diffusion-reaction equations for the charge particle densities (electrons, positive ions) and the Poisson's equation for the electric potential are treated by the finite volume method on unstructured meshes. The algorithm is

- new values of the electron density n_e and the ion density n_i are computed by an explicit scheme;
- the electrical potential is established from the new values $n_e,\ n_i$ by numerical solution of the Poisson's equation;
- the electric field $\vec{E}$ is computed from the electric potential;
- the electron drift velocity $\vec{v}_e$ and the diffusive coefficient D_e is computed as functions of the electric field $\vec{E}$;
- a new time layer is computed => new values of $n_e,\ n_i$;
- a new mesh is made by a dynamical mesh adaptation after several iterations;
- values of unknowns are interpolated form the old mesh to the new one.

3.1.1 Discretization of Electron Density Equation

The finite volume method is used for discretization of the electron density equation. The domain Ω is divided in a system of subdomains μ_i, $\Omega \equiv \bigcup_i \mu_i$, which do not overlap each other and have common boundaries. Integration over volume μ_i gives (the treatment is the same in 3D and leads to the same general form)

$$\iint_{\mu_i} \frac{dn_e}{dt} dV + \iint_{\mu_i} \operatorname{div}\left(n_e \vec{v}_e\right) dV - \iint_{\mu_i} \operatorname{div}\left(D_e \vec{\nabla} n_e\right) dV = \iint_{\mu_i} S_e dV. \tag{3.1}$$

By using Green's formula and dividing by the volume, the equation (3.1) leads to

$$\frac{dn_e}{dt} + \frac{1}{\mu_i} \oint_{\partial\mu_i} n_e \vec{v}_e \vec{\mathfrak{n}} ds - \frac{1}{\mu_i} \oint_{\partial\mu_i} D_e \vec{\nabla} n_e \vec{\mathfrak{n}} ds = S_e, \tag{3.2}$$

with the unit normal vector $\vec{\mathfrak{n}}$. Now we approximate the curvilinear integral by a summation. So one obtains for a volume μ_i

$$\frac{dn_e}{dt} = -\frac{1}{\mu_i} \sum_{j=1}^{m} \left(n^n_{e_{ij}} \vec{v}_{e_{ij}} \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| - D_{e_{ij}} \vec{\nabla} n^n_{eij} \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \right) + S^N_{e_i}, \tag{3.3}$$

where m is number of faces of volume μ_i, $\vec{\mathfrak{n}}_{ij}$ is the unit normal vector of the face σ_{ij} (face between volumes μ_i and μ_j) and $|\sigma_{ij}|$ is its length. Other variables denoted by subscript ij represent variables on the face σ_{ij}. At the end, the time derivative is approximated by the three steps Runge - Kutta method

$$\begin{aligned} n^{(0)}_{e_i} &= n^n_{e_i}, \\ n^{(k)}_{e_i} &= n^n_e + \alpha_{(k)} \operatorname{Res}\left(n^{(k-1)}_{e_i}\right), \\ n^{n+1}_{e_i} &= n^{(3)}_{e_i}, \end{aligned} \tag{3.4}$$

with $k = 1, 2, 3$ and the coefficients $\alpha_{(1)} = 0.5,\ \alpha_{(2)} = 0.5,\ \alpha_{(3)} = 1$. The residual in equation (3.5) is given by

$$\mathrm{Res}(n_{e_i}^{(k-1)}) = -\frac{\triangle t}{\mu_i} \sum_{j=1}^{m} \left(n_{e_{ij}}^{(k-1)} \vec{v}_{e_{ij}} \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| - D_{e_{ij}} \vec{\nabla} n_{eij}^{(k-1)} \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \right) + \triangle t \cdot S_{e_i}^{(k-1)}, \tag{3.5}$$

where $\triangle t$ is a time step.

3.1.1.1 Discretization of Convective Terms

The convective flux in electron density equation is computed by simple upwind scheme

$$n_{e_{ij}} = \begin{cases} n_{e_i} & \text{if } (\vec{v}_{e_{ij}} \cdot \vec{\mathfrak{n}}_{ij}) \geq 0, \\ n_{e_j} & \text{in other case,} \end{cases} \tag{3.6}$$

with assumption that the normal vector $\vec{\mathfrak{n}}_{ij}$ is oriented from the cell T_i to the cell T_j. Computation of the interface drift velocity $\vec{v}_{e_{ij}}$ will be explained later.

Higher order approximation in space

The upwind scheme has a first order of accuracy. It leads to a high numerical dissipation which is necessary to compensate by higher order algorithm. The scheme is therefore extended by a Van Leer's type MUSCL algorithm together with Barth-Jespersen limiter.

One introduce $\widetilde{n}_{e_i},\ \widetilde{n}_{e_j}$ in the equation (3.6) instead of $n_{e_i},\ n_{e_j}$

$$\begin{aligned} \widetilde{n}_{e_i} &= n_{e_i} + \psi_i \cdot \left(\vec{\nabla} n_{e_i} \cdot \vec{r}_i \right), \\ \widetilde{n}_{e_j} &= n_{e_j} + \psi_j \cdot \left(\vec{\nabla} n_{e_j} \cdot \vec{r}_j \right), \end{aligned} \tag{3.7}$$

where $\vec{\nabla} n_{e_i},\ \vec{\nabla} n_{e_j}$ are gradients of electron density in cells $T_i,\ T_j$. We can compute the gradients assuming that the electron density is piecewise linear function and its value n_{e_i} is in the center of gravity of the cell T_i. This linear function is computed by the least square method which include all neighbors of the cell T_i (neighbors with at least one common vertex). $\vec{r}_i$ is a vector coming from the center of gravity of the cell T_i to the center of gravity of a face σ_{ij}. In the same way, $\vec{r}_j$ is a vector coming from the center of gravity of the cell T_j to the center of gravity of the face σ_{ij}. ψ_i and ψ_j are Barth-Jespersen limiter function.
The gradient $\vec{\nabla} n_{e_i} = ((n_e)_x,\ (n_e)_y)$ from equation (3.7) is computed by following formulas in 2D

$$(n_e)_x = \frac{I_{yy} \cdot J_x - I_{xy} \cdot J_y}{D}, \qquad (n_e)_y = \frac{I_{xx} \cdot J_y - I_{xy} \cdot J_x}{D}, \tag{3.8}$$

where $I_{xx},\ I_{yy},\ I_{xy},\ J_x,\ J_y,\ D$ are given by equation (A.1).

We obtain a little bit more complicated relations in 3D for components of the gradient $\vec{\nabla} n_{e_i} = ((n_e)_x,\ (n_e)_y,\ (n_e)_z)$

$$(n_e)_x = \frac{\left(I_{yy}I_{zz} - I_{yz}^2\right) J_x + \left(I_{xz}I_{yz} - I_{xy}I_{zz}\right) J_y + \left(I_{xy}I_{yz} - I_{xz}I_{yy}\right) J_z}{D}, \tag{3.9}$$

$$(n_e)_y = \frac{\left(I_{xz}I_{yz} - I_{xy}I_{zz}\right) J_x + \left(I_{xx}I_{zz} - I_{xz}^2\right) J_y + \left(I_{xy}I_{xz} - I_{yz}I_{xx}\right) J_z}{D}, \tag{3.10}$$

$$(n_e)_z = \frac{\left(I_{xy}I_{yz} - I_{xz}I_{yy}\right) J_x + \left(I_{xy}I_{xz} - I_{yz}I_{xx}\right) J_y + \left(I_{xx}I_{yy} - I_{xy}^2\right) J_z}{D}, \tag{3.11}$$

where $I_{xx},\ I_{yy},\ I_{zz},\ I_{xy},\ I_{xz},\ I_{yz},\ J_x,\ J_y,\ J_z,\ D$ are defined by equations (A.2)-(A.4).

The Barth-Jespersen limiter is given by following formula

$$\psi_i = \begin{cases} min\left(1,\ \frac{n_{e_{max}} - n_{e_i}}{\triangle_2}\right), & \triangle_2 > 0, \\ 1, & \triangle_2 = 0, \\ min\left(1,\ \frac{n_{e_{min}} - n_{e_i}}{\triangle_2}\right), & \triangle_2 < 0, \end{cases} \tag{3.12}$$

where $n_{e_{max}}$ is maximum of electron density from values in the cell T_i and all neighboring cells with common face, $n_{e_{min}}$ is minimum of electron density from values in the cell T_i and all neighboring cells with common face. $\triangle_2$ is given by relation $\triangle_2 = \vec{\nabla} n_{e_i} \cdot \vec{r}$ with $\vec{r}$ which comes from the cell's center of gravity to the center of gravity of one of faces of the cell T_i. Final ψ_i is the minimum over the faces.

3.1.1.2 Discretization of Dissipative Terms

The curvilinear integral of the diffusive term in the equation (3.2) is approximated by numerical integration around a cell boundary, so one can write

$$\oint_{\partial\mu_i} D_e \vec{\nabla} n_e^n \vec{n} ds = \sum_{j=0}^{m} D_{eij} \vec{\nabla} n_{eij}^n \vec{\mathfrak{n}}_{ij} |\sigma_{ij}|, \tag{3.13}$$

where $\vec{\nabla} n_{eij}^n$ represents gradient of the electron density on the face σ_{ij}, $\vec{\mathfrak{n}}_{ij}$ is a unit normal vector of the face σ_{ij} and $|\sigma_{ij}|$ is its measure. The constant m indicate the total number of a cell's faces. Computation of the diffusion coefficient D_{eij} is described in the equation (3.32).

Gradient Computation in 2D
Approximation of the electron density gradient $\vec{\nabla} n_{eij}$ in 2D is done on the diamond cell (see Fig. 3.1). The diamond cell is constructed by connection of centers of gravity ($L,\ R$) of cells

$T_i,\ T_j$ which share the face σ_{ij} and its endpoints $A,\ B$. One obtains the co-volume $D_{\sigma_{ij}}$ by this construction.

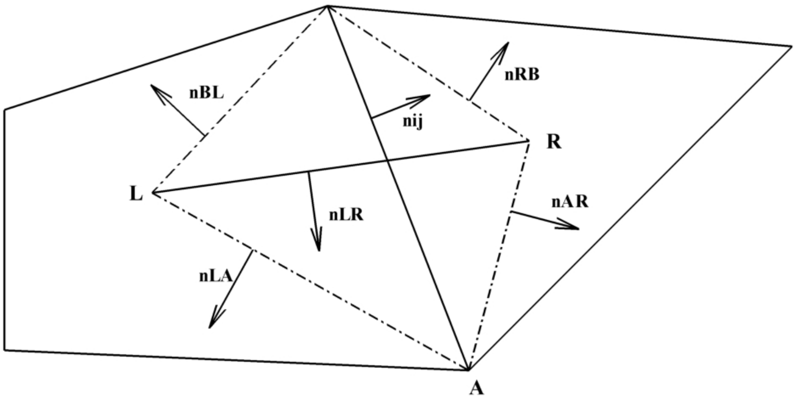

Figure 3.1: Diamond cell in 2D

One can assume that the electron density gradient is constant on the co-volume $D_{\sigma_{ij}}$. According to Green-Gauss theorem the approximation leads to

$$\vec{\nabla} n_{eij} = \frac{1}{\mu\left(D_{\sigma_{ij}}\right)}\left[\frac{1}{2}\left(n_e(A)+n_e(R)\right)\vec{\mathfrak{n}}_{AR}|\sigma_{AR}| + \frac{1}{2}\left(n_e(R)+n_e(B)\right)\vec{\mathfrak{n}}_{RB}|\sigma_{RB}| + \right.$$
$$\left. + \frac{1}{2}\left(n_e(B)+n_e(L)\right)\vec{\mathfrak{n}}_{BL}|\sigma_{BL}| + \frac{1}{2}\left(n_e(L)+n_e(A)\right)\vec{\mathfrak{n}}_{LA}|\sigma_{LA}|\right], \tag{3.14}$$

where $n_e(A)$ $(n_e(B),\ \cdots)$ represents value of the electron density in the point A $(B,\ \cdots)$, $\vec{\mathfrak{n}}_{AR}$ is a unit normal vector of the co-volume face σ_{AR} and $|\sigma_{AR}|$ is its length. The others co-volume faces and their normal vectors are labeled analogically (see Fig. 3.1).
When we factor one half out of the square bracket and modify the expression inside the square brackets, we obtain

$$\vec{\nabla} n_{eij} = \frac{1}{2\mu\left(D_{\sigma_{ij}}\right)}\Big[\left(\vec{\mathfrak{n}}_{AR}|\sigma_{AR}| + \vec{\mathfrak{n}}_{LA}|\sigma_{LA}|\right) n_e(A) + \left(\vec{\mathfrak{n}}_{RB}|\sigma_{RB}| + \vec{\mathfrak{n}}_{BL}|\sigma_{BL}|\right) n_e(B) +$$
$$+ \left(\vec{\mathfrak{n}}_{AR}|\sigma_{AR}| + \vec{\mathfrak{n}}_{RB}|\sigma_{RB}|\right) n_e(R) + \left(\vec{\mathfrak{n}}_{BL}|\sigma_{BL}| + \vec{\mathfrak{n}}_{LA}|\sigma_{LA}|\right) n_e(L)\Big]. \tag{3.15}$$

The following equalities are obvious

$$
\begin{aligned}
\vec{\mathfrak{n}}_{LA}|\sigma_{LA}| + \vec{\mathfrak{n}}_{AR}|\sigma_{AR}| &= \vec{\mathfrak{n}}_{LR}|\sigma_{LR}|,\\
\vec{\mathfrak{n}}_{AR}|\sigma_{AR}| + \vec{\mathfrak{n}}_{RB}|\sigma_{RB}| &= \vec{\mathfrak{n}}_{ij}|\sigma_{ij}|,\\
\vec{\mathfrak{n}}_{RB}|\sigma_{RB}| + \vec{\mathfrak{n}}_{BL}|\sigma_{BL}| &= -\vec{\mathfrak{n}}_{LR}|\sigma_{LR}|,\\
\vec{\mathfrak{n}}_{BL}|\sigma_{BL}| + \vec{\mathfrak{n}}_{LA}|\sigma_{LA}| &= -\vec{\mathfrak{n}}_{ij}|\sigma_{ij}|.
\end{aligned}
\tag{3.16}
$$

If we substitute relations (3.16) in the equation (3.15), it gives

$$
\vec{\nabla} n_{eij} = \frac{1}{2\mu\left(D_{\sigma_{ij}}\right)} \Big[\left(n_e(A) - n_e(B)\right) \vec{\mathfrak{n}}_{LR}|\sigma_{LR}| + \left(n_e(R) - n_e(L)\right) \vec{\mathfrak{n}}_{ij}|\sigma_{ij}| \Big]. \tag{3.17}
$$

The electron density values $n_e(A),\ n_e(B)$ in the face σ_{ij} end points are computed by the least square method

$$
n_e(A) = \sum_{p=1}^{N(A)} \alpha_p(A) n_{ep}, \quad n_e(B) = \sum_{p=1}^{N(B)} \alpha_p(B) n_{ep}, \tag{3.18}
$$

where n_{ep} is value of the electron density in the cell T_p, $N(A)$ $(N(B))$ is number of cells including vertex A (B), $\alpha_p(A)$ $(\alpha_p(B))$ are weights coming from the least square method. One can write (for vertex A)

$$
\alpha_p(A) = \frac{1 + \lambda_x \left(x_p - x_A\right) + \lambda_y \left(y_p - y_A\right)}{N(A) + \lambda_x R_x + \lambda_y R_y}, \tag{3.19}
$$

where the weights parameters are given by formulas (A.5).

Gradient approximation in 3D

The diamond cell $D_{\sigma_{ij}}$ in 3D is constructed by the centers of gravity $L,\ R$ of cells $T_i,\ T_j$ which contain the common face σ_{ij}. The rest of vertices of the diamond cell are vertices of the face σ_{ij} (see Fig. 3.2). The electron density gradient $\vec{\nabla} n_{eij}$ is approximated by following formula which coming from the equations (A.6) - (A.9)

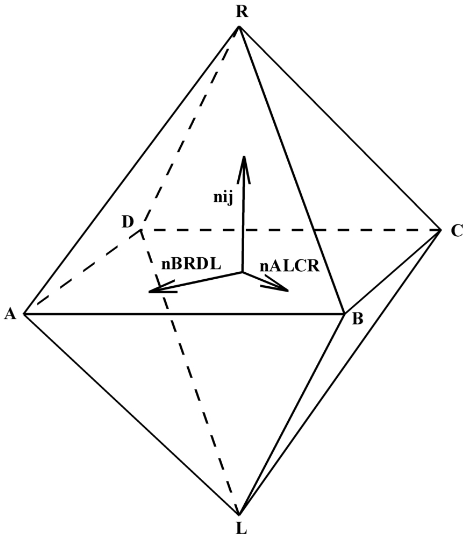

Figure 3.2: Diamond cell in 3D

$$\vec{\nabla} n_{eij} = \frac{1}{3\mu\left(D_{\sigma_{ij}}\right)}\Big[\left(n_e(A) - n_e(C)\right)\vec{\mathfrak{n}}_{BRDL}|\sigma_{BRDL}| + \\ + \left(n_e(B) - n_e(D)\right)\vec{\mathfrak{n}}_{ALCR}|\sigma_{ALCR}| + \left(n_e(R) - n_e(L)\right)\vec{\mathfrak{n}}_{ij}|\sigma_{ij}|\Big]. \tag{3.20}$$

The electron density values $n_e(A),\ n_e(B),\ n_e(C),\ n_e(D)$ are computed by the least square method

$$\begin{aligned} n_e(A) &= \sum_{p=1}^{N(A)} \alpha_p(A) n_{ep}, \quad n_e(B) = \sum_{p=1}^{N(B)} \alpha_p(B) n_{ep}, \\ n_e(C) &= \sum_{p=1}^{N(C)} \alpha_p(C) n_{ep}, \quad n_e(D) = \sum_{p=1}^{N(D)} \alpha_p(D) n_{ep}. \end{aligned} \tag{3.21}$$

The weights $\alpha_p(A)$ is given by

$$\alpha_p(A) = \frac{1 + \lambda_x\left(x_p - x_A\right) + \lambda_y\left(y_p - y_A\right) + \lambda_z\left(z_p - z_A\right)}{D}, \tag{3.22}$$

where $\lambda_x,\ \lambda_y,\ \lambda_z,\ D$ are expressed in relations (A.10) - (A.12).

3.1.2 Discretization of the Positive ion Density

The equation for ion density is in the minimal model only an ordinary differential equation and its numerical approximation is represented by three steps Runge - Kutta method (equivalent of equation (3.5))

$$\begin{aligned} n_{i_i}^{(0)} &= n_{i_i}^{n}, \\ n_{i_i}^{(k)} &= n_i^n + \alpha_{(k)} \mathrm{Res}\left(n_{i_i}^{(k-1)}\right), \\ n_{i_i}^{n+1} &= n_{i_i}^{(3)}, \end{aligned} \tag{3.23}$$

where the coefficients $\alpha_{(k)}$ are the same as in the equation (3.5) and the residual is expressed by relation

$$\mathrm{Res}(n_{i_i}^{(k-1)}) = \triangle t \cdot S_{e_i}^{(k-1)}. \tag{3.24}$$

3.1.3 Discretization of Poisson's Equation

Discretization of Poisson's equation leads to a system of linear equations with a sparse matrix. Such system of equation can be solved by direct methods or by iterative methods. Iterative methods gives possibilities as Jacobi iterative method, Gauss-Seidel iterative method or super relaxation method which is improved Gauss-Seidel iterative method. There are also gradients method such as Biconjugate gradients method or *gmrs* method (generalized minimum residual method). The *gmrs* method together with proper preconditioning is very often used in numerical simulations (solving of implicit scheme, etc.). Gauss elimination or LU factorization belong among direct methods. The LU factorization is more suitable when right hand side of the system of equation is changed. Because of dynamic mesh adaptation, the matrix of the system of equations is changed after a given number of iterations and the right hand side is changed each iteration. This leads to "quasi-stationary" matrix of equations and therefore LU factorization is the proper method in our case. We use MKL library from Intel to solve the system of liner equations by LU factorization.
The Poisson's equation is discretized by a central type approximation which leads to a system of linear equations

$$\boldsymbol{A} \cdot \vec{V}^{n+1} = \vec{b}^{n+1}. \tag{3.25}$$

$\boldsymbol{A}$ is a matrix of coefficients, $\vec{V}$ is a vector of unknowns (its dimension is equaled to the total number of cells) and $\vec{b}$ is a vector of right hand side. The updated values n_e^{n+1}, n_i^{n+1} are used for the evaluation of the source term in equation (2.2). A row i in the matrix $\boldsymbol{A}$ corresponds to the cell T_i. We use a similar finite volume method approximation as for diffusive terms in the equation for electron density (3.13).

$$\frac{1}{\mu_i} Q = \frac{1}{\mu_i} \iint_{\mu_i} \triangle V dV = \frac{1}{\mu_i} \oint_{\partial \mu i} \vec{\nabla} V \mathfrak{n} ds \approx \frac{1}{\mu_i} \sum_{j=1}^{m} \vec{\nabla} V_{ij} \mathfrak{n}_{ij} |\sigma_{ij}|. \tag{3.26}$$

An approximation of the gradient $\vec{\nabla} V_{ij}$ is also made by the diamond cell (see Fig. 3.1, Fig. 3.2). According to the equation (3.17), one can write for **2D case**

$$\vec{\nabla} V_{ij} = \frac{1}{2\mu(\sigma_{ij})} \left[(V_A - V_B) \cdot \vec{\mathfrak{n}}_{LR} |\sigma_{LR}| + (V_R - V_L) \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \right]. \tag{3.27}$$

When we substitute the equation (3.27) to a relation

$$\triangle V = \frac{1}{\mu_i} \sum_{j=1}^{m} \vec{\nabla} V_{ij} \vec{\mathfrak{n}}_{ij} |\sigma_{ij}|, \tag{3.28}$$

we obtain the final form of approximation of the Poisson's equation in **2D**

$$\begin{aligned} \triangle V = \sum_{j=1}^{m} \Bigg[& \frac{1}{2\mu_i \mu\left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{LR} |\sigma_{LR}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_A - \\ & - \frac{1}{2\mu_i \mu\left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{LR} |\sigma_{LR}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_B + \\ & + \frac{1}{2\mu_i \mu\left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_R - \\ & - \frac{1}{2\mu_i \mu\left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \Bigg], \end{aligned} \tag{3.29}$$

where electrical potential values $V_A,\ V_B$ are computed by the least square method (see equation (3.18)) with weights computed according to equation (3.19).

For **3D case**, the electrical potential gradient is approximated according to equation (3.20), so one can write

$$\begin{aligned} \vec{\nabla} V_{eij} = \frac{1}{3\mu\left(D_{\sigma_{ij}}\right)} \Big[& (V_A - V_C) \vec{\mathfrak{n}}_{BRDL} |\sigma_{BRDL}| + \\ & + (V_B - V_D) \vec{\mathfrak{n}}_{ALCR} |\sigma_{ALCR}| + (V_R - V_L) \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \Big]. \end{aligned} \tag{3.30}$$

If we substitute the equation (3.30) to the equation (3.28), the numerical approximation of the electrical potential in **3D** leads to

$$
\begin{aligned}
\triangle V = \sum_{j=1}^{m} \Bigg[& \frac{1}{3\mu_i \mu \left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{BRDL} |\sigma_{BRDL}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_A - \\
& - \frac{1}{3\mu_i \mu \left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{BRDL} |\sigma_{BRDL}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_C + \\
& + \frac{1}{3\mu_i \mu \left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{ALCR} |\sigma_{ALCR}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_B - \\
& - \frac{1}{3\mu_i \mu \left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{ALCR} |\sigma_{ALCR}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_D + \\
& + \frac{1}{3\mu_i \mu \left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_R - \\
& - \frac{1}{3\mu_i \mu \left(D_{\sigma_{ij}}\right)} \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot \vec{\mathfrak{n}}_{ij} |\sigma_{ij}| \cdot V_L \Bigg] .
\end{aligned}
\tag{3.31}
$$

The electrical potential values $V_A,\ V_B,\ V_C,\ V_D$ come from the least square method (equation (3.21)). The least square weights are computed by equation (3.22).

3.1.4 Discretization of Intensity of Electric Field, Electron Drift Velocity and Diffusion Coefficient

The intensity of electric field is given as a negative gradient of the electrical potential $\vec{E}^n = -grad\,(V^n)$. The intensity of electric field is computed on cell faces by diamond scheme (see equation (3.17) and (3.20)).

The electron drift velocity is a function of the intensity of electric field $\vec{v}^n_{e_{ij}} = f\left(\vec{E}^n_{ij}\right)$ and depends on the ratio $\frac{\|\vec{E}^n_{ij}\|}{N}$ [V cm^2], where N is the neutral gas density ($N = 2.5 \cdot 10^{19}\text{cm}^{-3}$). The precise formula is expressed in (B.1).

Finally we can compute the diffusion coefficient D_e as a function of the intensity of electric field and the electron drift velocity

$$
D^n_{e_{ij}} = \left[0.3341 \cdot 10^9 \cdot \left(\frac{\|\vec{E}^n_{ij}\|}{N} \right)^{0.54069} \right] \cdot \frac{\|\vec{v}^n_{e_{ij}}\|}{\|\vec{E}^n_{ij}\|}. \tag{3.32}
$$

Values at the cell center of gravity are estimated as average value of the values on the cell faces.

3.1.5 Discretization of Source Terms

The source terms depends on the electron drift velocity and the electron density, so one can write for the cell T_i

$$
\begin{aligned}
S_{e_i}^n &= \frac{\alpha}{N} \cdot \|\vec{v}_{e_i}^n\| \cdot n_{e_i}^n \cdot N, \\
S_{i_i}^n &= S_{e_i}^n,
\end{aligned} \tag{3.33}
$$

where the ratio $\frac{\alpha}{N}$ [cm^2] is computed by the following formula

$$
\begin{aligned}
&\text{if } \frac{\|\vec{E}\|}{N} > 1.5 \cdot 10^{-15}, && \frac{\alpha}{N} = 2 \cdot 10^{-16} \cdot \exp\left(\frac{-7.248 \cdot 10^{-15}}{\|\vec{E}_i^n\|/N}\right), \\
&\text{else,} && \frac{\alpha}{N} = 6.619 \cdot 10^{-17} \cdot \exp\left(\frac{-5.593 \cdot 10^{-15}}{\|\vec{E}_i^n\|/N}\right).
\end{aligned} \tag{3.34}
$$

3.2 Some Results of Numerical Analysis in 2D

3.2.1 Convergence

The analysis is describe in more details and is taken from [20].
We suppose a case when elements of a mesh satisfied classic hypotheses of regularity:

There are constants $\alpha_1,\ \alpha_2,\ \beta_1,\ \beta_2 > 0$ such that for all $T_j \in \mathcal{T}_h$ and for all $e \in \partial T_j$

$$
\alpha_1 h_j^2 \leq \mu(T_j) \leq \alpha_2 h_j^2 \text{ and } \beta_1 h_j \leq mes(e) \leq \beta_2 h_j \tag{3.35}
$$

where h_j is a diameter of the triangle T_j and $\mu(T_j)$ is its volume.

We consider a elliptic problem

$$
\left\{
\begin{array}{rl}
-div\left(A\vec{\nabla}u\right) = f & \text{in } \ \Omega \subset \mathbb{R}^2 \\
u = g & \text{on } \ \Gamma = \partial\Omega
\end{array}
\right. \tag{3.36}
$$

where $A(x,y)$ is a symmetric positive definite matrix with coefficients a_{ij} in $C^1(\Omega)$, $f \in C^0(\Omega)$ and $g \in C^2(\Gamma)$.

We can construct co-volume $D_{\sigma_{ij}}$ around the face σ_{ij} (see Fig. 3.1). We can discretize $\frac{\partial u}{\partial x}|_{\sigma_{ij}}$ by following formula

$$
\frac{\partial u}{\partial x}|_{\sigma_{ij}} \approx \frac{1}{\mu(D_{\sigma_{ij}})} \sum_{\varepsilon \in D_{\sigma_{ij}}} \frac{1}{2}\left(u_{N_1} + u_{N_2}\right) \int_{\varepsilon} \mathfrak{n}_{x\varepsilon} d\sigma \tag{3.37}
$$

where ε is a face of the co-volume $D_{\sigma_{ij}}$ and $\mathfrak{n}_{x\varepsilon}$ is an axial component of its unit normal vector. N_1 and N_2 are starting and ending points of the face ε and $u_{N_1},\ u_{N_2}$ represent values of the state u in the points $N_1,\ N_2$.

For a node P of a mesh, we can use use linear approximation v of u in all cells which contain the node P. We can write

$$u_P = \sum_{K \in V(P)} \alpha_K(P) u_K, \tag{3.38}$$

where $V(P)$ are all triangles containing the node P, a_K is a state in the center of a triangle K and $\alpha_K(P)$ are weights of interpolation.

Let's label u_h an approximation of the exact solution u. u_j is an approximation of an average value in a triangle T_j.

We define exact and approached fluxes for a face e of a triangle K

$$\begin{aligned} \Phi^{Exact}(u) &= \frac{1}{mes(e)} \int_e A(s)p(s) \cdot \vec{\mathfrak{n}}_e ds \qquad \text{with } \ p = \vec{\nabla} u, \\ \Phi^{App}(u_h) &= (A_e p_e) \vec{\mathfrak{n}}_e, \end{aligned} \tag{3.39}$$

where A_e is an average value of A on the face e and it can be express in local base directions $\left(\vec{\mathfrak{n}}_e, \vec{\mathfrak{t}}_e\right)$ by following way

$$A = \begin{bmatrix} \lambda_e & \mu_e \\ \mu_e & \nu_e \end{bmatrix}. \tag{3.40}$$

Approximation of the gradient p_e on the face e can be express in the base $\left(\vec{\mathfrak{n}}_e, \vec{\mathfrak{t}}_e\right)$ like (see Fig. 3.1)

$$p_e = \left(\frac{u_R - u_L}{h_e} - \alpha_e \frac{u_B - u_A}{mes(e)} \right) \vec{\mathfrak{n}}_e + \frac{u_B - u_A}{mes(e)} \vec{\mathfrak{t}}_e \tag{3.41}$$

with $h_e = \overrightarrow{LR} \cdot \vec{\mathfrak{n}}_e$ and $\alpha_e = tan\left(\vec{\mathfrak{n}}_e, \overrightarrow{LR}\right) = \dfrac{\overrightarrow{LR} \cdot \vec{\mathfrak{t}}_e}{\overrightarrow{LR} \cdot \vec{\mathfrak{n}}_e}$. This gives the form of the approached diffusion flux

$$\Phi^{App}(u_h) = \lambda_e \frac{u_R - u_L}{h_e} + \beta_e \frac{u_B - u_A}{mes(e)}, \tag{3.42}$$

where $\beta_e = \mu_e - \alpha_e \lambda_e$.
Remark: if $A = Id$. the flux Φ^{App} is coherent with the formula (3.17).

Definition 3.2.1 *For $u \in H^2(\Omega)$, we define the consistency error as a difference between the exact and approached fluxes*

$$R_e(u) = \Phi^{Exact}(u) - \Phi^{App}(\pi_h u) \tag{3.43}$$

where $\pi_h u$ is a L^2 projection of u on the piecewise constant functions on cells $\mathcal{T}_h$.
The scheme is called weakly consistant if

$$\|R_h(u)\|_{L^2(\Omega)} \xrightarrow[h\to 0]{} 0 \tag{3.44}$$

where $R_{h|\mu(D_{\sigma_{ij}})}(u) = R_e(u)$.

The finite volume discretization of the equation (3.36) can be written in following discrete form, which depends on the parameter h and tends to the exact problem

$$\text{Let's find } u_h \text{ such that } \mathcal{A}_h u_h = f_h \text{ in } \Omega \text{ and } u_h = g \text{ on } \Gamma = \partial\Omega$$

where $f_{h,i} = \dfrac{1}{\mu(T_i)} \displaystyle\int_{T_i} f(x,y)dxdy$.
$\mathcal{A}_h$ is a discrete operator defined as

$$\int_{T_i} \mathcal{A}_h dxdy = \sum_{e\in\partial T_i} \epsilon_{T_i e} mes(e) \Phi_e^{App}(u_h) \tag{3.45}$$

where $\epsilon_{T_i e} = \pm 1$ if the normal vector of the face e is interior or exterior in the triangle T_i.

Let's denote

$$|u_h|_1 = \left(\sum_{e\in\mathcal{A}_h} mes(e) \frac{u_R - u_L}{d_e} \right)^{\frac{1}{2}} \tag{3.46}$$

where u_L and u_R are upstream and downstream values of u_h in the direction of the normal of the face e and $d_e = dist(X_R, X_L)$,

and let's denote

$$\langle \mathcal{A}_h u_h, u_h \rangle = \sum_{T_i\in\mathcal{T}_h} \left(\mathcal{A}_h u_h\right)_i u_{h,i}. \tag{3.47}$$

Definition 3.2.2 *Scheme is said coercive if* $\exists\alpha > 0$ *such as*

$$\langle \mathcal{A}_h u_h, u_h \rangle \geq \alpha |u_h|_1^2. \tag{3.48}$$

Definition 3.2.3 (Discrete Poincaré inequality)
We say that discrete Poincaré inequality is verified on a regular triangulation $\mathcal{T}_h$ *if* $\exists\beta > 0$ *independent of* h *such that*

$$\|u_h\|^2_{L^2(\Omega)} \leq \beta |u_h|_1^2. \tag{3.49}$$

Theorem 3.2.4 *If the scheme is weakly consistant, coercive and verifies discrete Poincaré inequality, then it is convergent*

$$\|u - u_h\|^2_{L^2(\Omega)} \xrightarrow[h\to 0]{} 0. \tag{3.50}$$

The following lemma gives a sufficient condition, based on the interpolation weights, for the weak consistency.

Lemma 3.2.5 *Whether the interpolation weights verify for all nodes P of a mesh*

(i) $\alpha_K(P) \geq 0$,

(ii) $\displaystyle\sum_{K \in V(P)} \alpha_K(P) = 1$,

(iii) $\displaystyle\sum_{K \in V(P)} \alpha_K(P) \overrightarrow{PK} = \vec{0}$,

then the scheme (3.37) is weakly consistant.

The least square approximation respects this conditions. More details about the analysis in [20].

3.2.2 Discrete Maximum Principle for Diamond Type scheme

The algorithm deals with discrete maximum principle based on repair technique for diamond type scheme for diffusion problem. We consider equation (3.36) and $\mathcal{T}$ is a set of all cells in a domain.

The maximum principle says if $f(x) \leq 0$ for all $x \in \Omega$ than the maximum of $u(x)$ is on the boundary

$$u(x) \leq \min_{x \in \partial\Omega} g(x), \quad \text{for all } x \in \Omega, \tag{3.51}$$

if $f(x) \geq 0$ for all $x \in \Omega$ than the minimum of $u(x)$ is on the boundary

$$u(x) \geq \max_{x \in \partial\Omega} g(x), \quad \text{for all } x \in \Omega. \tag{3.52}$$

The discrete maximum principle for all cells K says if $f_K \leq 0$ for all $K \in \mathcal{T}$ then

$$U_K \leq \min_{J \in \partial\Omega} g_J \quad \text{for all } K \in \mathcal{T}, \tag{3.53}$$

if $f_K \geq 0$ for all $K \in \mathcal{T}$ then

$$U_K \geq \max_{J \in \partial\Omega} g_J \quad \text{for all } K \in \mathcal{T}, \tag{3.54}$$

where J is boundary midpoint and the Dirichlet boundary condition is prescribed $g_J = g(J)$.

There are discussed two repair techniques in [21]. The first one is a local repair technique and the second one is a global repair technique.

The total energy of of the diffusion equation in a discrete form is defined as

$$E(U) = \sum_{K \in \mathcal{T}} U_K V_K, \tag{3.55}$$

where V_K is a volume of the cell K. It is important to keep the conservation of the total energy when the discrete solution is modified because of the satisfaction of the maximum principle. We define $U_{min} = \min_{J \in \partial\Omega} g_j$ as the minimum on the boundary. We require the repaired solution to have the same total energy $E(U^r) = E(U)$.

Local Repair Technique

The solution U is repaired cell by cell in this technique. If $U_K \leq U_{min}$ in the cell K, we have to add a needed energy $E_K = -(U_K - U_{min})V_K$ to the cell K. It leads to $U_K^r = U_{min}$. $\mathcal{S}(K)$ is a set of all cell which have a common edge with the cell K. The available energy which can be transfered from a cell L ($L \in \mathcal{S}(k)$) to the cell K is

$$E_L^a = \max(0, (U_L - U_{min})V_L). \tag{3.56}$$

The total available energy coming from all neighboring cells is

$$E^a = \sum_{L \in \mathcal{S}(K)} E_L^a. \tag{3.57}$$

There is enough available energy to correct the solution in the cell K if $E^a \geq E_K$. In such a case we set $U_K^r = U_{min}$ and the needed energy E_K from neighboring cells by following formula

$$U_L^r = U_L - \frac{E_L^a \frac{E_K}{E^a}}{V_L}. \tag{3.58}$$

In case that $E^a < E_K$, we enlarged the set $\mathcal{S}(K)$ by adding of all neighboring cells with cells in $\mathcal{S}(K)$.

Global Repair Technique

If we divide the set $\mathcal{T}$ into subsets $\mathcal{T}_p = \{K; U_K \geq U_{min}\}$, $\mathcal{T}_n = \{K; U_K < U_{min}\}$, we can define a total available energy in the whole computational domain end a total need energy in the whole computational domain. The total available energy is

$$E_p = \sum_{K \in \mathcal{T}_p} (U_K - U_{min})V_K \tag{3.59}$$

and the total needed energy is

$$E_n = \sum_{K \in \mathcal{T}_n} -(U_K - U_{min})V_K. \tag{3.60}$$

We can write for each $K \in \mathcal{T}$

$$E_K = \begin{cases} (U_K - U_{min})V_K & K \in \mathcal{T}_p \\ -(U_K - U_{min})V_K & K \in \mathcal{T}_n \end{cases}. \tag{3.61}$$

It is necessary to be satisfied $E_p \geq E_n$ in order to we can use the repair technique. In such a case the total energy is positive and we have enough energy which can be transport from $\mathcal{T}_p$ to $\mathcal{T}_n$. When the needed energy E_n is compensated, the total energy in $\mathcal{T}_n$ is

$$\begin{aligned} \sum_{K\in\mathcal{T}_n} U_K^r V_K &= \sum_{K\in\mathcal{T}_n} U_K V_K + E_n = \\ &= \sum_{K\in\mathcal{T}_n} U_K V_K + \left(- \sum_{K\in\mathcal{T}_n} (U_K - U_{min})V_K \right) = \\ &= \sum_{K\in\mathcal{T}_n} U_{min} V_K. \end{aligned} \tag{3.62}$$

Because the repaired solution has to satisfy $U_K^r \geq U_{min}$ we can see that

$$U_K^r = U_{min} \quad \text{for all } K \in \mathcal{T}_n. \tag{3.63}$$

We prescribe

$$U_K^r = U_K - \frac{E_k \frac{E_n}{E_p}}{V_K} \quad \text{for all } K \in \mathcal{T}_p. \tag{3.64}$$

Because $E_p \geq E_n$, we obtain

$$U_K^r \geq U_K - \frac{E_K}{V_K} = U_{min} \quad \text{for all } K \in \mathcal{T}_p. \tag{3.65}$$

More details in [21].

3.3 Dynamic Mesh Adaptation

Because of the increase of the efficiency of the presented scheme, the dynamic mesh adaptation is performed to construct a mesh which is able to capture the stiff gradient of variables in areas of streamer head propagation.

There are several possibilities how to adapt a mesh. One can distinguish mesh adaptation for steady cases and on the other hands for the unsteady cases.

A progressive refinement is often use in steady computations to catch significant phenomenons such as shock waves in fluid flows. It means that we refine (split cells) a mesh in the direction of a shock wave for a once time and let the computation run. If its necessary we can refine the mesh in the same area again. Another possibility is a deformation of cells - moving of vertices and if it necessary omission of some nodes.

When an unsteady simulation run, the dynamical mesh adaptation can increase the efficiency of the computation because we need less cells. It leads to decrease of CPU time and used memory. A mesh is therefore adapted in each time step if it necessary in such way that cells are reduced in the significant area and enlarge in areas where it isn't necessary. It means when we want to reduce a cell we move its vertices closer to each other, to enlarge a cells we move its vertices apart or we omit a vertex. We can also use different way how to reduce cells. We can divide cells into sub-cells which are similar to the original ones and unrefine them when the significant phenomenon moves from this area.

A quasi-stationary adaptation is also a possibility how to increase the efficiency of the code. We use a reference grid where we compute a criterion to know which cells should be refined. Then we refine the cells, satisfy the conformity and the computation run on the refined grid for given time steps. After that we create new mesh and interpolate values of unknowns from the last refined grid to the new one.

We choose the last type of mesh refinement because the refined area is sufficient large and the streamer moves in this area for some time. The mesh is again refined before the streamer head leaves the refined area.

3.3.1 Criterion

The algorithm starts by computing a criterion on a fixed coarse base mesh. The criterion consists of several parameters (e.g. gradient of source terms). The criterion is scaled to an interval $\langle 0,\ 1\rangle$. Then the criterion is smoothed by ten iterations of diffusion equation. After the smoothing the criterion once more scaled to the interval $\langle 0,\ 1\rangle$. Next the interval split on subintervals which say how many times each cell of the reference grid will be divided. The algorithm satisfies that the level of refinement among neighboring cells cannot be higher than one. The grid conformity is applied when each triangle is subdivided. Moreover in 2D if a cell has two neighbors with a common face which should be divided on four sub-triangles, the cell will be also divided on four triangles. Mores about the adaptation criterion is in the section 5.3. After a given number of iteration, the new criterion is computed on reference grid and we create new refined mesh. The values from old refined grid a interpolated to the new one.

3.3.2 Mesh refinement in 2D

The refinement in 2D is performed for meshes which consist only of triangular cells. First the reference triangle is subdivided in four new triangles. Then the conformity satisfies that two neighboring triangles have only one common face.

3.3.2.1 Cell dividing

The faces midpoints are used for the cell dividing. They are used together with the cell vertices to create four new triangles from the original one. The new triangles have a level of refinement decreased of one. The algorithm is repeated until all cells have level of refinement equaled to zero (see Fig. 3.3).

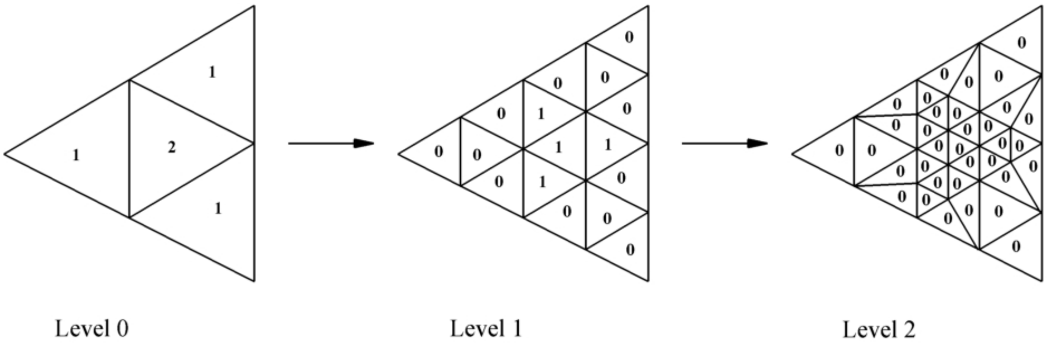

Figure 3.3: Mechanism of multi-level refinement in 2D

3.3.2.2 Conformity

If the levels of refinement among neighboring cells are different, we have to satisfy the mesh conformity. For each cell, we check the level of refinement of neighboring cells with common face. In case that at least two neighboring cells has higher level of refinement, we increase the level of refinement for given cell and it's divided by standard algorithm. In case that only one neighboring cell has higher level of refinement, we split given cell on two triangles (see Fig. 3.3, part Level 2). This algorithm slightly enlarges the refined area.

3.3.3 Mesh refinement in 3D

The mesh refinement algorithm in 3D is very similar to the algorithm in 2D. The conformity step is a little bit different and will be explained later. The presented mesh refinement algorithm works only with tetrahedrons.

3.3.3.1 Cell dividing

The level of refinement for each cell is set according to a criterion as in 2D case. In one step of refinement a given cell is subdivided into eight new tetrahedrons using edges midpoints and vertices from the original cell. New tetrahedron is created at each vertex. When we imaginary cut these tetrahedrons out of the original cell, a double-pyramid remains. This double-pyramid is subsequently split into four tetrahedrons (see Fig. 3.4).

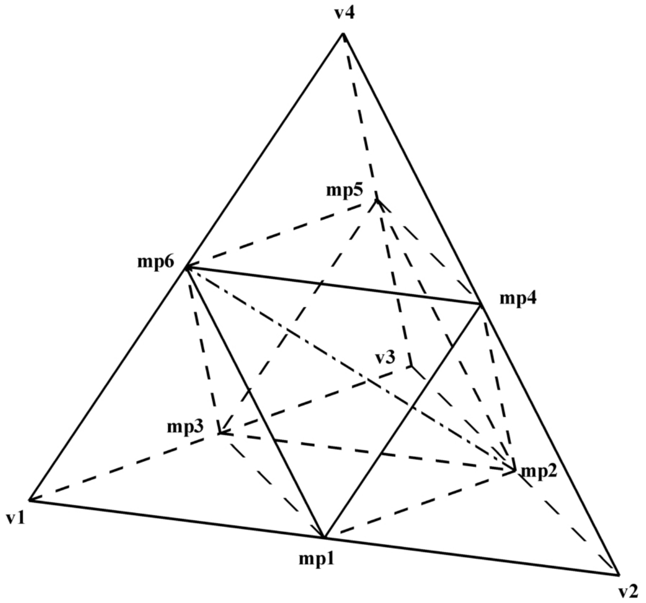

Figure 3.4: Mechanism of refinement in 3D

If the original tetrahedron has vertices labeled $v1,\ v2,\ v3,\ v4$, the new eight tetrahedrons are labeled

- $v1,\ mp1,\ mp3,\ mp6$,
- $mp1,\ v2,\ mp2,\ mp4$,
- $mp6,\ mp4,\ mp5,\ v4$,
- $mp3,\ mp2,\ v3,\ mp5$,
- $mp1,\ mp2,\ mp3,\ mp6$.
- $mp1,\ mp4,\ mp6,\ mp2$,
- $mp3,\ mp5,\ mp6,\ mp2$,
- $mp2,\ mp4,\ mp5,\ mp6$.

3.3.3.2 Conformity

If a given cell shouldn't be divided, we have to satisfy conformity of mesh. We check all neighboring cells common at least on edge if there are cells which has been divided. There are several cases how to split the cell according number of edges which contains divided neighboring cells. We split the given cell into two new tetrahedrons if only one edge contains divided neighbors (see Fig. 3.5).

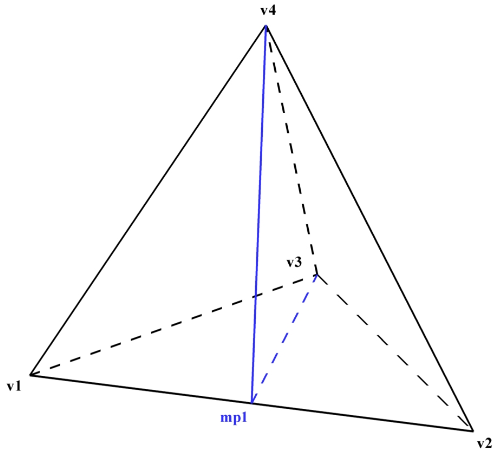

Figure 3.5: Conformity when one edge should be split

According to the splitting, we obtain tetrahedrons

- $v1,\ mp1,\ v3,\ v4,$
- $mp1,\ v2,\ v3,\ v4.$

If each of six edges of a given cell should be divided, the cell is split as it is regular cell described in section 3.3.3.1.

In case that more than one but less then six edges should be split, we firstly divide the given cell into four tetrahedrons using the cell center of gravity C (see Fir. 3.6). It leads to set of tetrahedrons

- $v1,\ v2,\ C,\ v4,$
- $v1,\ v2,\ v3,\ C,$
- $v2,\ v3,\ C,\ v4,$
- $v3,\ v1,\ C,\ v4.$

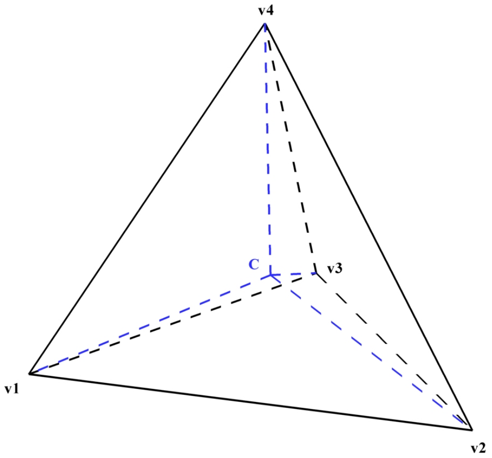

Figure 3.6: Splitting of tetrahedron using the center of gravity

Next we deal with each new tetrahedron separately. We check how many edges should be split. This gives three possibilities. We show the possibilities on the tetrahedron $v1,\ v2,\ C,\ v4$. If only one edge should be split, we obtain two tetrahedrons (see Fig. 3.7)

- $v1,\ mp1,\ C,\ v4$,
- $mp1,\ v2,\ C,\ v4$.

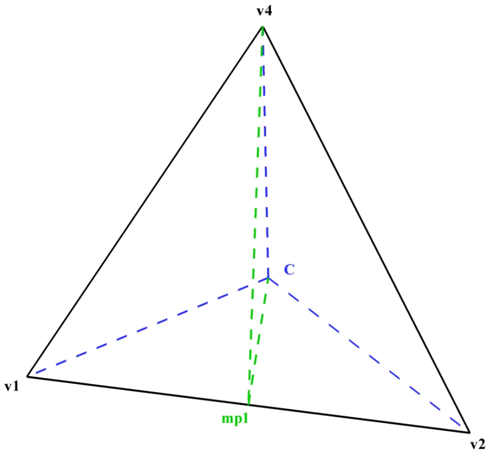

Figure 3.7: One edge to be split

In case that two edges should be split, it leads into three new tetrahedrons (see Fig. 3.8)

- $v1,\ mp1,\ C,\ mp2,$

- $mp2,\ mp1,\ C,\ v4,$

- $mp1,\ v2,\ C,\ v4.$

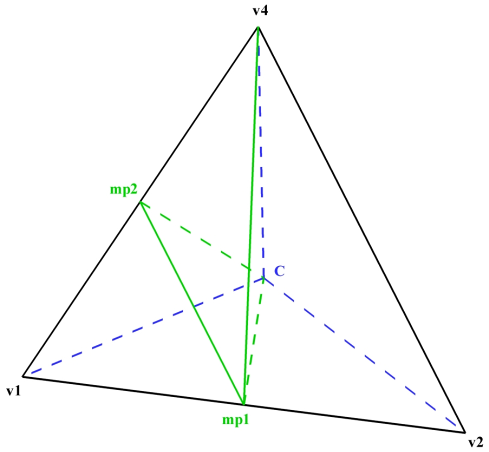

Figure 3.8: Two edge to be split

Last there is possibility that three edges of the tetrahedron $v1,\ v2,\ C,\ v4$ should be split. This gives four new tetrahedrons (see Fig. 3.9)

- $v1,\ m1,\ C,\ mp2,$

- $mp2,\ mp1,\ C,\ mp3,$

- $mp1,\ v2,\ C,\ mp3,$

- $mp2,\ mp3,\ C,\ v4.$

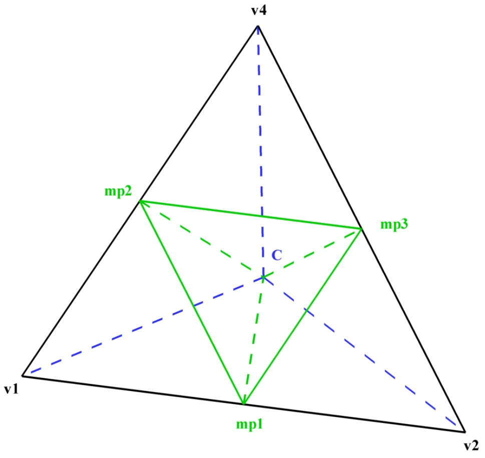

Figure 3.9: Three edge to be split

This way of satisfying of conformity doesn't expand the area of refinement. On the other hand, we obtain worse shape of cells. The conformity could be improved by the proposal in [18] and [19].

Chapter 4

Simple Tests of Grid Adaptation Algorithm

4.1 Linear Advection Equation, Grid Adaptation

These section deals with testing of the advection together with the mesh adaptation on a simple scalar equation

$$u_t + \vec{a} \cdot \nabla u = 0, \tag{4.1}$$

in one, two and three dimensions. The velocity vector $\vec{a}$ is $\vec{a} = 1$ in an one dimensional case, $\vec{a} = (1,\ 0)$ in a two dimensional case and $\vec{a} = (1,\ 0,\ 0)$ in a three dimensional case.
The numerical test are as for 1^{st} order of accuracy as for 2^{nd} order of accuracy scheme. The simple upwind scheme (see eq. (3.6)) is used in the 1^{st} order scheme. The extension of the upwind scheme on the second order in space is made by Van Leer's type MUSCL algorithm together with Barth-Jespersen limiter (see eq. (3.7)), more in [17]. The limiter is also used e.g. in [14]. The second order of accuracy in time is due to three steps Runge-Kutta method.

The testing domains are

- an interval $x \in \langle 0,\ 1\rangle$ for 1D case,
- a rectangle $\langle 0,\ 1\rangle \times \langle 0,\ 0.5\rangle$ for 2D case,
- a brick $\langle 0,\ 1\rangle \times \langle 0,\ 0.5\rangle \times \langle 0,\ 0.5\rangle$ for 3D case.

Initial condition
The initial condition is represented by Gaussian pulse with maximum magnitude of 10

$$u(x,y,z,0) = 10 \cdot e^{-\frac{(x-0.2)^2+(y-0.25)^2+(z-0.25)^2}{0.05^2}}. \tag{4.2}$$

The initial condition 4.2 is written for 3D case where the initial Gaussian pulse is a sphere. The initial condition for 2D and 1D cases are derived from 3D initial condition by omitting z variable in 2D ($y,\ z$ variables in 1D). The 3D case is also tested for cylindrical initial Gaussian pulse and in such case the initial condition is same as for 2D case.

Boundary conditions
For all test cases, the homogeneous Neumann's conditions are used at all boundaries.

4.1.1 Test Results

All presented results are at physical time $t = 0.6$ s. We can see results obtained by 1D computation in Figs. 4.1-4.2. Fig. 4.1 shows results computed by 1^{st} order scheme and 2^{nd} order scheme with CFL number equaled to 0.4. There is more numerical diffusion in the 1^{st} order scheme which leads to decrease of maximum value of the initial Gaussian pulse and its spreading during transportation. When the 2^{nd} order scheme is used, the maximum value almost doesn't decrease and the initial pulse doesn't spread.

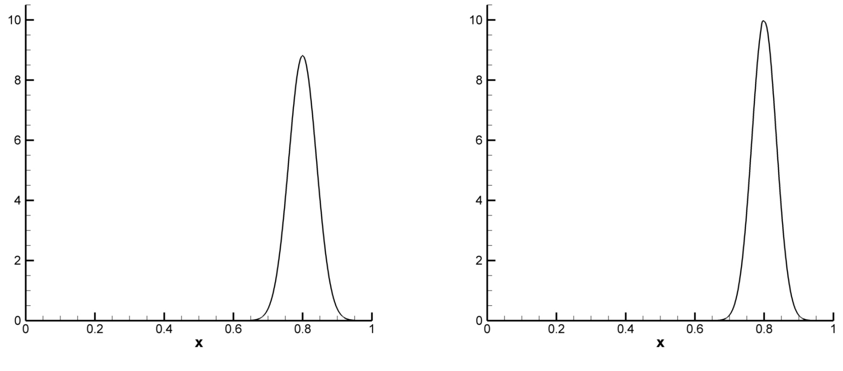

Figure 4.1: 1D computation: 1^{st} order scheme (left), 2^{nd} order scheme (right), $CFL = 0.4$

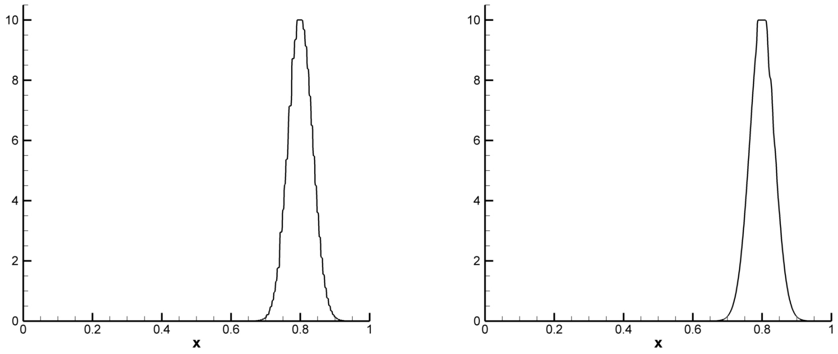

Figure 4.2: 1D computation: 1^{st} order in time, 2^{nd} order in space, $CFL = 0.4$ (left), $CFL = 0.1$ (right)

Fig. 4.2 gives results obtained by scheme which is 1^{st} order of accuracy in time and 2^{nd} order of accuracy in space. The left figure shows results gained with $CFL = 0.4$. We can see that the maximum value almost doesn't decrease but the quality of the pulse is very bad. Decreasing of time step ($CFL = 0.1$, right figure) gives better quality of the transported Gaussian pulse but not as good as 2^{nd} order scheme with bigger time step.

Results coming from 2D computation on structured mesh with approximately 300000 cells are in Figs. 4.3-4.5. The results represent transportation of a circle of the Gaussian pulse. Fig. 4.3 shows comparison of 1^{st} order scheme and 2^{nd} order scheme along the axis of computational domain $y = 0.25$. We can again see that the first order scheme is more dissipative than the second order scheme. Fig. 4.4 presents isolines of the same results.

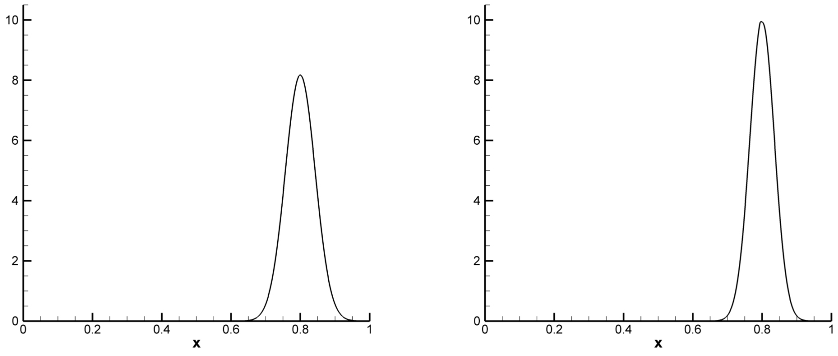

Figure 4.3: 2D computation: 1^{st} order scheme (left), 2^{nd} order scheme (right), $CFL = 0.4$

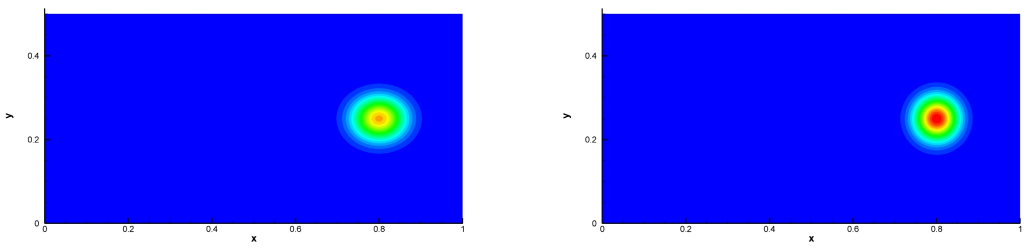

Figure 4.4: 2D computation: 1^{st} order scheme (left), 2^{nd} order scheme (right) isolines, $CFL = 0.4$

Fig. 4.5 shows results when the first order of accuracy in time and the second order of accuracy space is used. Together with smaller time step it gives relatively good results but not good as fully 2^{nd} order scheme.

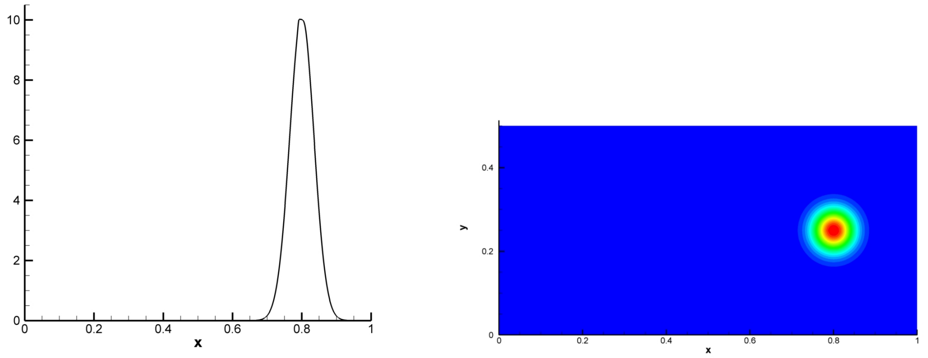

Figure 4.5: 2D computation: 1^{st} order in time, 2^{nd} order in space, $CFL = 0.1$

Fig. 4.6 presents comparison of results form 3D computation obtained on fine cartesian grid with approximately 1.2 million cells and on unstructured grid with approximately 60 thousands cells. The results (transportation of cylinder) shows magnitude of variable along the line $y = 0.25,\ z = 0.25$ computed by 2^{nd} order scheme. We can see that the dense of a grid is also very important.

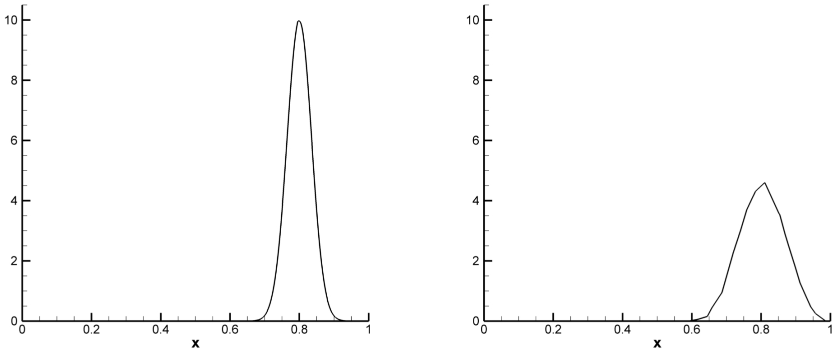

Figure 4.6: 3D computation: 2^{nd} order scheme $CFL = 0.4$, structured grid 1200000 cells (left), unstructured grid 60000 cells (right)

The Figs. 4.7-4.8 shows results (transportation of sphere) obtained on unstructured grid with 3 levels of mesh adaptation. It leads to mesh with approximately 240000 cells. If the important area is refined, we obtained good results with smaller grid and shorter CPU time. Fig. 4.7 presents isolines and mesh at slice $y = 0.25$. The magnitude of variable along the axis $y = 0.25,\ z = 0.25$, we can see in Fig. 4.8.

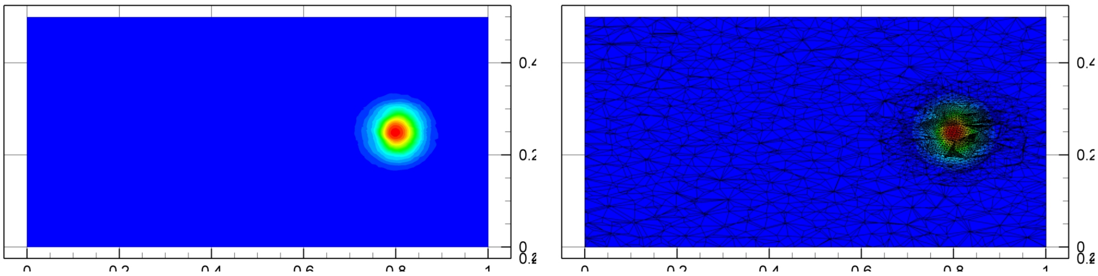

Figure 4.7: 3D computation: 2^{nd} order scheme $CFL = 0.4$, dynamical mesh adaptation, transportation of a sphere, isolines (left), mesh (right)

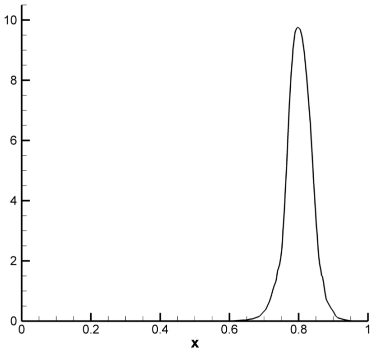

Figure 4.8: 3D computation: 2^{nd} order scheme $CFL = 0.4$, dynamical mesh adaptation, transportation of a sphere

	max	min
1^{st} order	8.8106	0
2^{nd} order	9.9664	0
1/2 order $CFL = 0.1$	9.9968	0
1/2 order $CFL = 0.4$	9.999	0

Table 4.1: 1D computation, 1000 cells

	max	min
1^{st} order	8.176	0
2^{nd} order	9.9433	$-3e-8$
1/2 order $CFL = 0.1$	10.021	$-4e-8$

Table 4.2: 2D computation, 775x387 cells

	max	min
2^{nd} order, grid 775x387x4	9.970	$-1e-5$
2^{nd} order, unstructured grid 60000 cells,	5.1348	-0.19
2^{nd} order, unstructured grid 60000 cells, sphere 3 levels	9.7774	-0.002

Table 4.3: 3D computation

Tables 4.1-4.3 present maximum and minimum values of the results at time $t = 0.6$ s. 1/2 order in tables means 1^{st} order in time and 2^{nd} order in space.

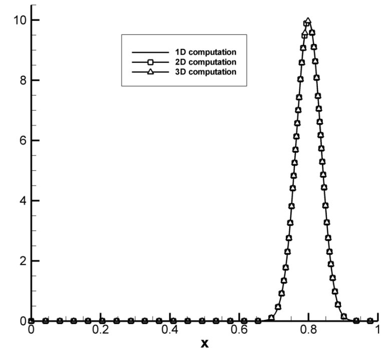

Figure 4.9: Comparison of results from different dimensions

The comparison of results obtained from computation in different dimensions is in Fig. 4.9. The results from 3D computation in the figure are from transportation of a cylinder on cartesian grid.

4.1.2 Comparison of limiters

This section compare results obtained by 2D computations where Barth-Jespersen and Van Albada limiters are used. The initial condition (4.2) is slightly modified

$$u(x,y,z,0) = 10^{16} \cdot e^{-\frac{(x-0.2)^2+(y-0.25)^2+(z-0.25)^2}{0.05^2}} \tag{4.3}$$

to verify that we can use such orders of magnitude.

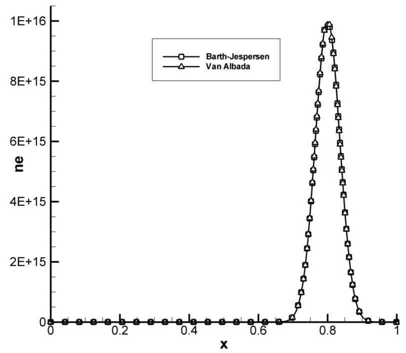

Figure 4.10: Comparison of limiters

Fig. 4.10 shows comparison of results of 2D computation on a structured grid with approximately 300000 cells. We can see that both Barth-Jespersen and Van Albada limiters gives almost the same results. The Barth-Jespersen limiter is unlike the Van Albada limiter independent on chosen parameters. The Bart-Jespersen limiter is therefore more suitable to use because we don't have to “tune” any parameter if a computation condition is changed.

Chapter 5

Numerical Results for 2D Planar Discharge

5.1 Description of Test Cases

We consider the propagation of 2D problem of discharge in homogeneous electric field described by equations (2.1)-(2.2). The first domain is a simple rectangle 1 × 0.5cm, the second domain has a left boundary partly replaced by a parabola. The parabola goes through points $[0, 0.15]$, $[0.2, 0.25]$ and $[0, 0.35]$ (see Fig. 5.1).

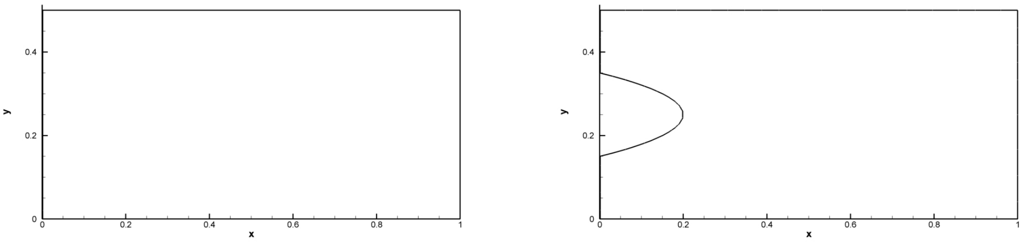

Figure 5.1: 2D computational domains

Initial Conditions
We consider the following initial conditions for the rectangular domain

$$\begin{aligned} n_e\left(x, y, 0\right) &= 10^{16} \cdot e^{-\frac{(x-0.2)^2+(y-0.25)^2}{\sigma^2}} + 10^9 \text{ [cm}^{-3}\text{]}, \quad \sigma = 0.01, \\ n_i\left(x, y, 0\right) &= n_e\left(x, y, 0\right). \end{aligned} \tag{5.1}$$

The initial Gaussian pulse for the electron and ion densities creates disturbance in the electric field which is necessary for initialization of ionization process in the wave propagation. The background electron and ion densities 10^9 cm^{-3} substitute the photoionization effect which is neglected in this simple streamer model. The computation is stopped at prescribed time when the head of electron avalanche is still inside the computational domain. In further computation, the discharge goes out through the right boundaries and the source terms increase to infinity but this effect doesn't represents physical behavior of the electric discharge.

The second domain (domain with parabolic left boundary) has different initial conditions

$$\begin{aligned} n_e\left(x,y,0\right) &= 10^9\ [\mathrm{cm}^{-3}], \\ n_i\left(x,y,0\right) &= n_e\left(x,y,0\right). \end{aligned} \tag{5.2}$$

The disturbance in the electric field is created by the shape of the left boundary so we don't need any initial Gaussian pulse.

Boundary Conditions
The left boundaries of the domains are anode, the right boundaries are plane cathodes. There are prescribed homogeneous Neumann conditions for all unknowns for upper and lower boundaries. Following boundary conditions are prescribed for the anode and the cathode

$$\begin{array}{ll} \dfrac{\partial n_e}{\partial \vec{n}} = 0, & \text{for anode and cathode,} \\ V = 25000\ [\mathrm{V}], & \text{for anode,} \\ V = 0\ [\mathrm{V}], & \text{for cathode.} \end{array} \tag{5.3}$$

5.2 Discharge in Rectangular Domain

Figs. 5.3 and 5.2 present results obtained with the second upwind scheme on dynamically adaptive grid in rectangular domain. The shape and magnitude of velocity vectors agrees with physical mechanism of the electric discharge propagation. The results in front of the scale in Fig. 5.3 are rotated about 90° in positive sense.

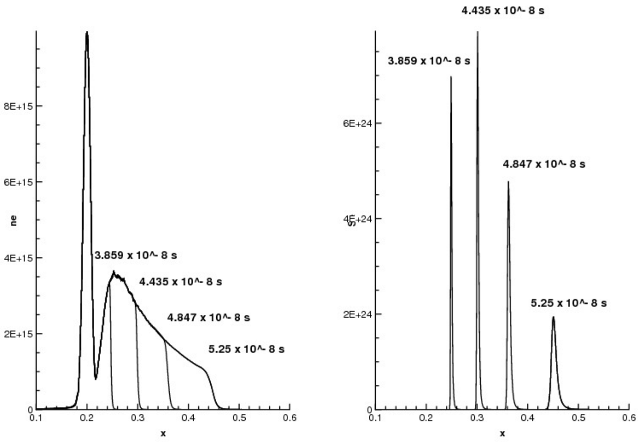

Figure 5.2: 2^{nd} order computation: Evolution of electron density (left) and source terms (right) magnitudes, 2^{nd} order scheme, adapted grid

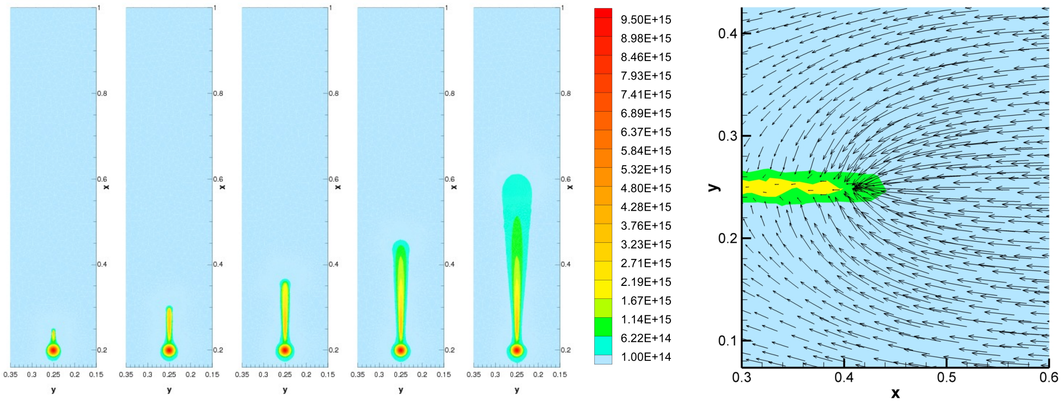

Figure 5.3: 2^{nd} order computation: Electron density (contour levels) at times $t = 3.859 \cdot 10^{-8}s$, $t = 4.435 \cdot 10^{-8}s$, $t = 4.847 \cdot 10^{-8}s$, $t = 5.25 \cdot 10^{-8}s$ and $t = 5.659 \cdot 10^{-8}s$ (end of computation), 2^{nd} order scheme, adapted grid; electron velocity vectors at a head of a discharge (right)

5.3 Grid Adaptation Test

The properties of the grid adaptation algorithm are the subject of the tests. It is important to have sufficiently high density grid in critical areas of the computational domain. These areas are the streamer (planar wave in 2D) path and the head of the discharge. The grid adaptation up to level 4 is therefore used. We need on the other hand to avoid too large extension of region with high level of adaptation because of almost explosive increase of number of cells.

Test Description
All computations are done on the same reference grid with 4700 cells (see Fig. 5.4). The first order scheme is used in the tests. As the initial condition we use the temporal result achieved on adapted grid at physical time $t_{in} = 3.14 \cdot 10^{-8}$ s (see Table 5.1) calculated from initial conditions (5.3).

We can see (Fig.5.4), that in this time the locally modified electric field is developed. It further initiated the growth and motion of electron avalanche. The computation starts with grid adaptation step and next the time evaluation is calculated. It means, that the new grid is created according to prescribed specific version of grid adaptation algorithm and initial solution is conservatively interpolated into finite volumes of new grid. After that we start to compute the time evaluation of solution. We use the same criterion for calculation of time step, the same CFL number, fixed number of computed time steps between grid adaptation. The computed cases differs only in grid adaptation algorithm or its parameters. We compare results achieved at the same physical time $T = 5.25 \cdot 10^{-8}$ s. The number of triangles and nodes in Table 5.2 means the numbers on adapted grid at time $t = T$. The CPU time is the relative CPU time necessary for computation of solution in time interval $\langle t_{in}, T \rangle$.

	cells	nodes
reference mesh	4700	2424
mesh at $t_{in} = 3.14 \cdot 10^{-8}$	9154	4651

Table 5.1: Number of cells, nodes of reference mesh and mesh at $t_{in} = 3.14 \cdot 10^{-8}$

adaptation criterion	cells	nodes	CPU time
source terms	11760	5954	1
electron density gradient	16714	8431	0.447
both, algorithm V1	18520	9334	0.716
both, algorithm V2	22422	11285	1.522
new distribution, alg. V2	46304	23226	2.551

Table 5.2: Number of cells, nodes, time steps and CPU time at $t = 5.25 \cdot 10^{-8}s$

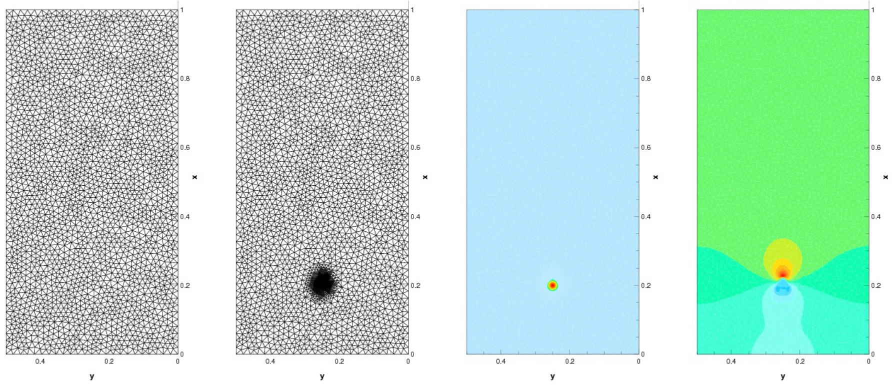

Figure 5.4: Reference mesh, mesh, electron density, electric field at $t_{in} = 3.14 \cdot 10^{-8}$

Results Presentation

Each result is presented in a one row. The computational domain is rotated 90° anti-clockwise. The horizontal coordinate $\langle 0, 0.5\rangle$ is then oriented from right to left, the vertical coordinate $\langle 0, 1\rangle$ is oriented from bottom to up. The discharge propagates from bottom.

The first and second column shows the isolines of the electron density n_e and the source term S_e at time T in zoom $\langle 0.15, 0.35\rangle$ horizontally and $\langle 0.16, 1\rangle$ vertically. The adapted grid at time T is plotted in third column. The relation of adapted grid density and isolines of the electron density n_e is shown in fourth (zoom $\langle 0.18, 0.32\rangle$ x $\langle 0.15, 0.4\rangle$) and fifth columns (zoom $\langle 0.18, 0.32\rangle$ x $\langle 0.4, 0.68\rangle$). The zoom of adapted grid around planar wave head is plotted together with isolines of the sorce term S_e in the last sixth column.

Criterion Variables

The first row in Fig. 5.5 shows results, when the magnitude of the source term is chosen as criterion function $crit = S_e$ (case 1). The second row shows results of case 2, when the criterion is based on gradient of the electron density n_e, $crit = |\text{grad} n_e|$. Finally we present results of case 3 , when both previous criterion are used simultaneously $crit = \max\,(|\text{grad}\ n_e|, S_e)$. We use this expression in two different algorithms V1 and V2.

In the version V1 (row 3 in Fig. 5.5), both criteria are scaled to one (dividing by the maximum value respectively). Then for each cell is chosen maximum of these two scaled criteria. The new criterion is smoothed using diffusion equation and scaled once more to one and it gives the final new criterion.

In the version V2 (see Fig. 5.6), each criterion is smoothed and scaled to one separately and the new criterion is maximum of these two values.

Discussion of Results

We can observe a significant influence of choice of criterion variable on position of head of the planar wave in fixed time T and consequently on discharge propagation velocity. The

lower level of grid adaptation in region of high values of source term S_e brings smearing of source term and higher propagation speed - typically case 2, partly case 3 version V1. On the other hand the criterion based only on the magnitude of source term - case 1 - produces too low values of electron density n_e in discharge path (initial part). The results of case 1 and 2 confirm, that the critical regions are well and efficiently (without significant enlargement) covered by highly refined (level 4 or 3) triangles.
We can also observe, that only simultaneous use of both variables (case 3) in grid adaptation criterion produce acceptable results. The position of planar wave head and shape of source term as in case 1 is similar only in version V2.

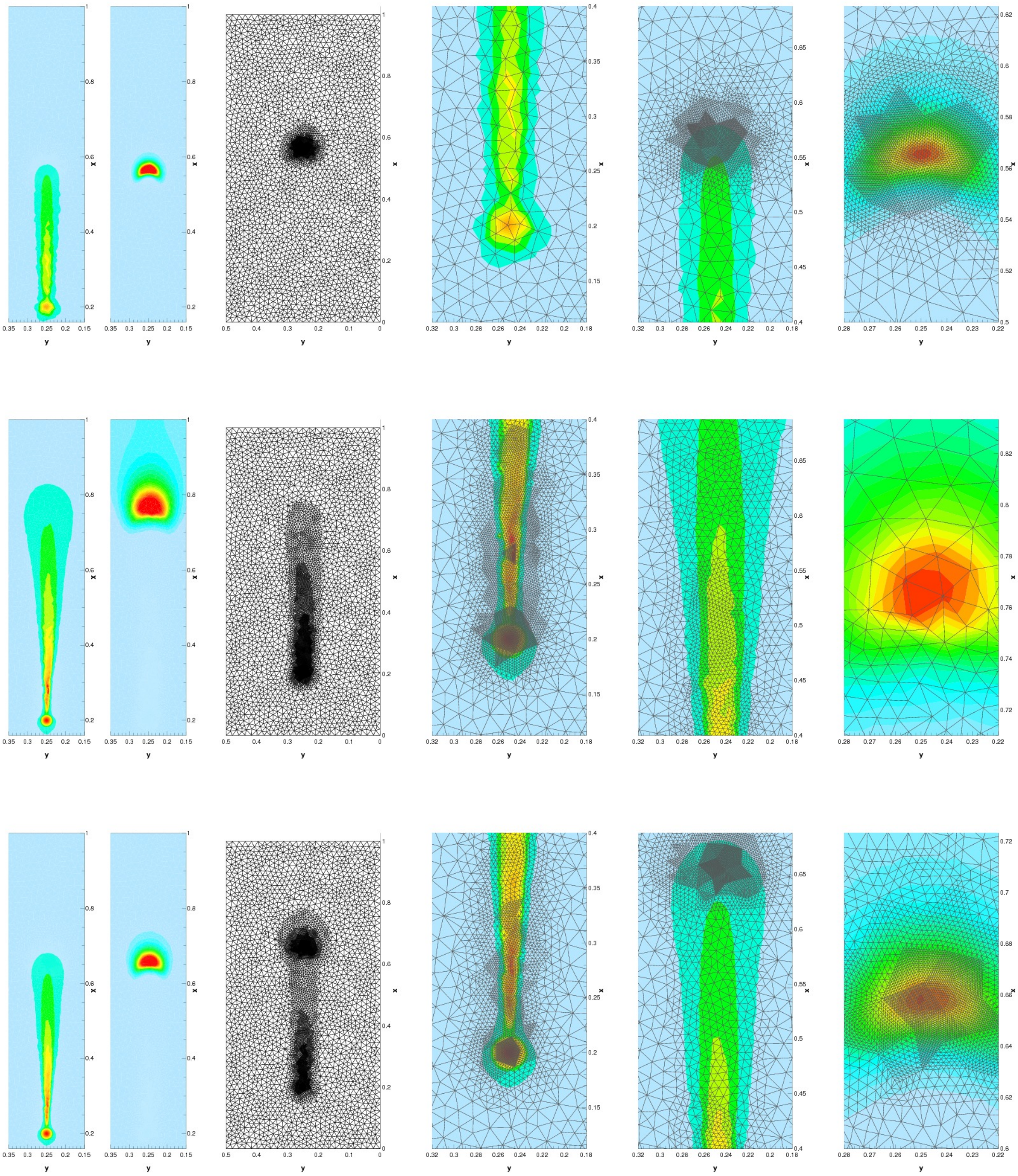

Figure 5.5: Criterion based on source terms (first row), criterion based on gradient of electron density (second row), criterion based on source terms and gradient of electron density - algorithm V1 (third row)

The table 5.2 shows, that the highest number of grid cells is also in case 3, version V2.

Comparing to case 1 it uses twice more cells, the number of time steps is almost the same and CPU time is 50% increasing. The maximum of grid points (about 22 000) confirms high efficiency of dynamic grid adaptation algorithm for solved problem. When all triangles of reference grid will be adapted up to fourth level, the number of volumes overcomes 1 200 000.

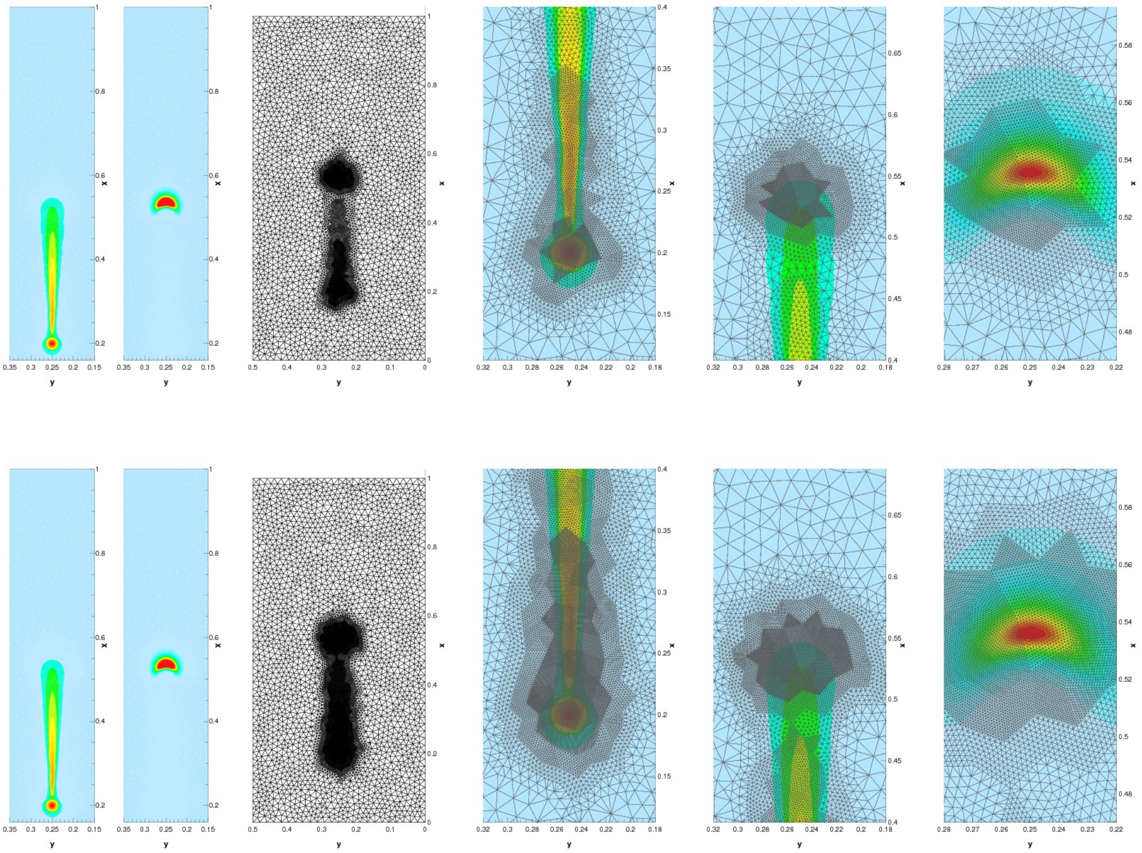

Figure 5.6: Criterion based on source terms and gradient of electron density - algorithm V2, basic distribution of level of adaptation(first row), enlarged interval of higher level of adaptation (second row)

Distribution of Subintervals of Adaptation Levels

The results in first row in Fig. 5.6 as well as all results presented in Fig. 5.5 were computed by the same basic definition of subintervals of $\langle 0, 1\rangle$ of different levels of adaptation - level 4 for $crit \in \langle 0.64, 1\rangle$, level 3 for $crit \in \langle 0.36, 0.64)$, level 2 for $crit \in \langle 0.16, 0.36)$, level 1 for $crit \in \langle 0.04, 0.16)$. The results presented in the second row in Fig. 5.6 were achieved by the same algorithm, but with enlarged interval of highest level of adaptation - level 4 for $crit \in \langle 0.36, 1\rangle$, level 3 for $crit \in \langle 0.16, 0.36)$, level 2 for $crit \in \langle 0.08, 0.16)$, level 1 for $crit \in \langle 0.04, 0.08)$.

Discussion of Results
We can observe higher density of grid in plotted regions, but also very similar distribution of n_e as well as position and shape of source term. The number of grid volumes is two times higher as for basic distribution, number of time steps is almost the same and CPU is 1.7 time higher.We can conclude, that the quality of achieved result is not too sensitive on choice of subinterval of highest level of adaptation.

5.4 Grid Density and Numerical Convergence Test

The influence of the grid density is studied for both 1^{st} and 2^{nd} order of approximation in space variables. The test cases are constructed with the same way as in previous subsection.

Test Description
The solution at time $t_{in} = 3.14 \cdot 10^{-8}$ s (see Fig. 5.4) is once more chosen as initial approximation. The computational grid is obtained as uniformly refined reference grid, i.e. all triangles of reference grid are split into the same number of 4^{lr} triangles in dependence on level of refinement $lr = 2, \cdots, 5$. The computation starts with generation of grid and conservative interpolation of unknowns at starting time into volumes of grid. Then the time evolution is calculated up to the prescribed time. We use the same CFL number for all cases.

Uniformly Refined Grid
Fig. 5.7 shows the isolines of net charge density $\rho = e\,(n_i - n_e)$ obtained by first order scheme. The isolines of net charge density computed by second order scheme are plotted in Fig. 5.8. 1D plots of the electron density n_e and the net charge density ρ along the axis of discharge ($y = 0.25$) document the influence of the grid density (see Fig. 5.9 and Fig. 5.10). Presented results approve that the first order scheme over predicts the electric discharge propagation speed. This property is in accordance with the conclusion in [15]. We can observe in both Fig. 5.7, and Fig. 5.9 that the position of the discharge head as well as the magnitude of the electron density decreases with higher grid density. The magnitude of net charge density is increasing. The distribution of net charge also shows, that the difference between results on neighboring coarse and fine grid slowly decreases with increasing of gird density (position and magnitude of maximum).

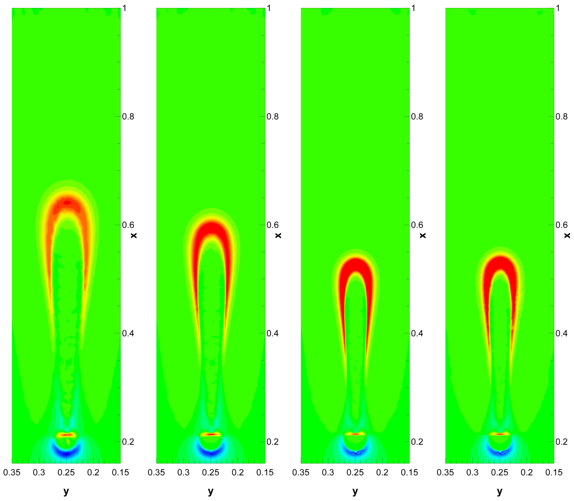

Figure 5.7: $1^{st} order$, comparison of net charge density computed on different uniformly refined meshes at time $t = 5.25 \cdot 10^{-8} s$. From left to right: 2 levels, 3 levels, 4 levels, adaptation (4 levels)

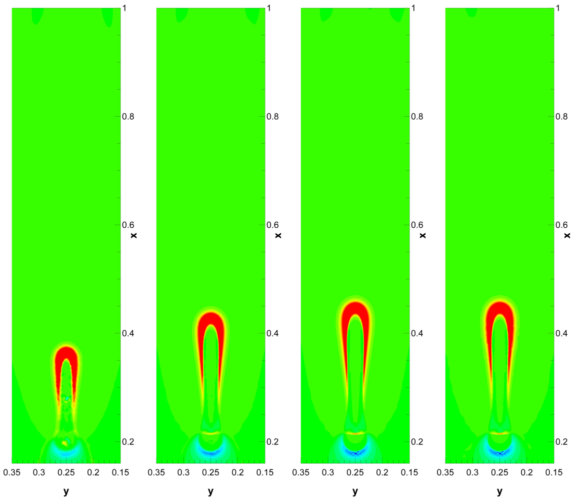

Figure 5.8: $2^{nd} order$, comparison of net charge density computed on different uniformly refined meshes at time $t = 5.25 \cdot 10^{-8} s$. From left to right: 2 levels, 3 levels, 4 levels, adaptation (4 levels)

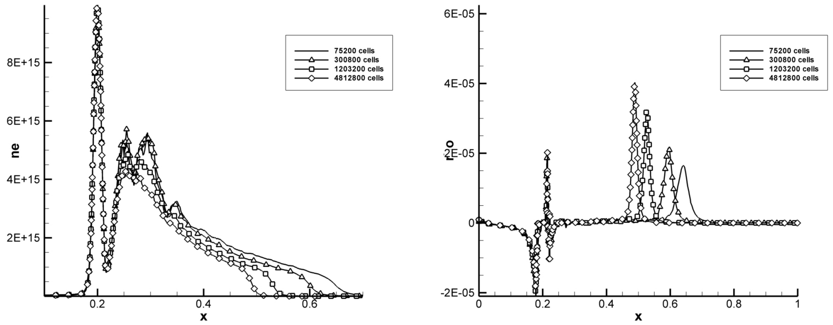

Figure 5.9: $1^{st}order$, comparison of electron density (left) and net charge density (right) computed on different coarse meshes at time $t = 5.25 \cdot 10^{-8}s$t: 2 levels, 3 levels, 4 levels, 5 levels

The method with second order scheme in space variables has different behavior. The position of the discharge head and magnitude of electron density increases and magnitude of net charge decreases when the grid density is increasing. We can also observe much smaller difference between results (net charge distribution in Fig.5.10) on neighboring coarse and fine grid as in case of first order scheme. This property also in some sense document higher accuracy and convergence of second order scheme.
We can see that the first order scheme is still too dissipative and inaccurate even if very fine mesh is used. The Fig. 5.10 shows that 4 levels of mesh refinement is sufficient to obtain satisfactory accurate results with second order scheme.

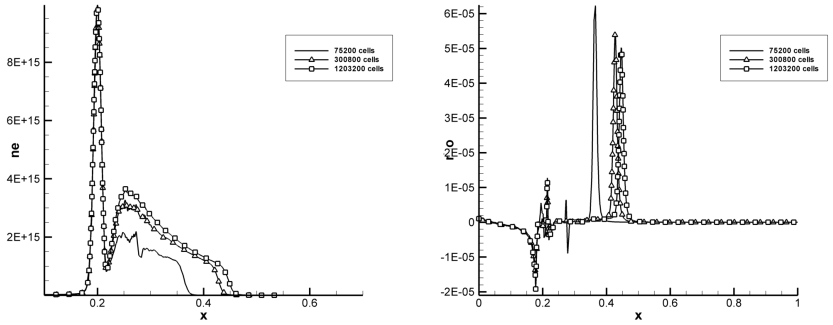

Figure 5.10: $2^{nd}order$, comparison of electron density (left) and net charge density (right) computed on different coarse meshes at time $t = 5.25 \cdot 10^{-8}s$.t: 2 levels, 3 levels, 4 levels

Although the difference between the first and the second order schemes decrease with increasing mesh density, it is still important. We can see the difference plotted in Fig. 5.11 for grid with 4 levels of refinement.

	cells	nodes	CPU time
2 levels uniformly refined grid	75200	37893	1
3 levels uniformly refined grid	300800	150985	8.158
4 levels uniformly refined grid	1203200	602769	58.663
Adapted grid, algorithm V2	18636	9392	1.084

Table 5.3: $2^{nd} order$: number of cells, nodes and CPU time at time $t = 5.25 \cdot 10^{-8} s$

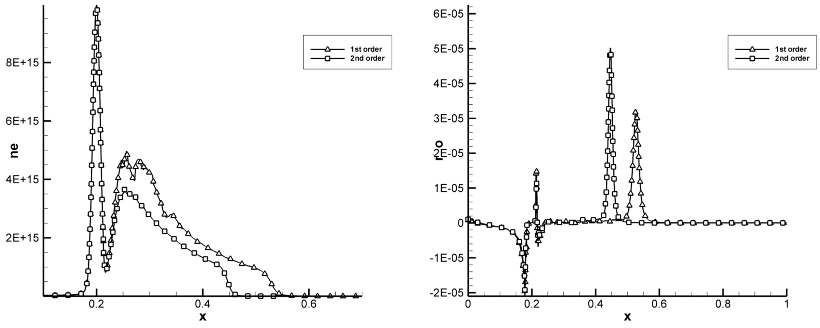

Figure 5.11: Comparison of electron density (left) and net charge density (right) for first and second order schemes, grid with 4 levels of refinement

Comparison of Uniformly and Dynamically Adapted Grid

Finally the accuracy and efficiency of proposed dynamic mesh adaptation algorithm is tested. The results are compared with the numerical results obtained on uniformly refined grid with 4 levels of refinement. These comparisons are plotted for both the fist and the second order scheme in form of the electron density n_e and the net charge density ρ along the axis (Fig. 5.12 and Fig. 5.13). The comparison is also shown in form of isolines for the net charge density in Fig. 5.7 and Fig. 5.8 where the third column is uniformly refined grid and the fourth column is dynamically refined grid. We can see that the results are almost identical. It confirms that the accuracy of uniformly refined grid is conserved. The table 5.3 shows the high efficiency of the dynamic mesh adaptation algorithm.

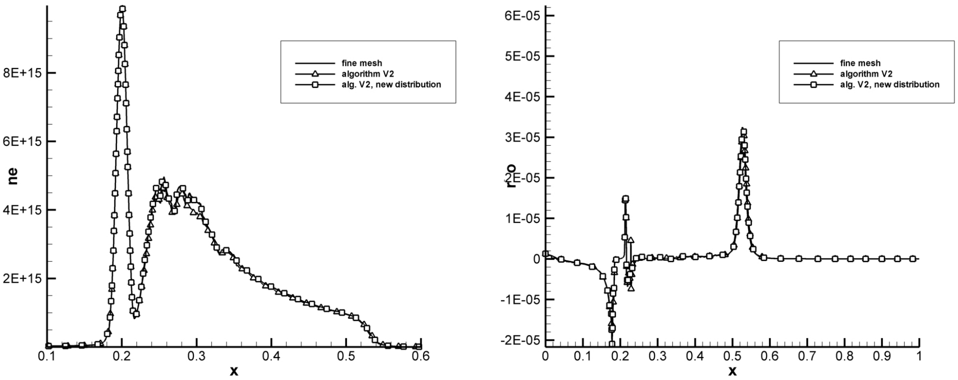

Figure 5.12: $1^{st} order$, comparison of electron density (left) and net charge density (right) computed on fine mesh and mesh with dynamic adaptation

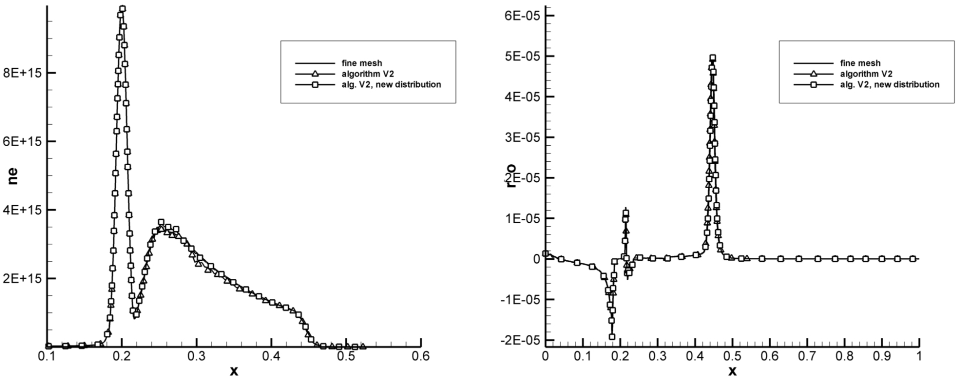

Figure 5.13: $2^{nd} order$, comparison of electron density (left) and net charge density (right) computed on fine mesh and mesh with dynamic adaptation

5.5 Planar Wave in Domain with Parabolic Anode

Figs. 5.14-5.18 present time evolution planar wave in 2D domain where the anode has a parabolic shape. The disturbance in the electric field is given by the shape of the anode and there is no initial Gaussian pulse. We can see in the figures evolution of the electron density, the net charge density and the source terms.

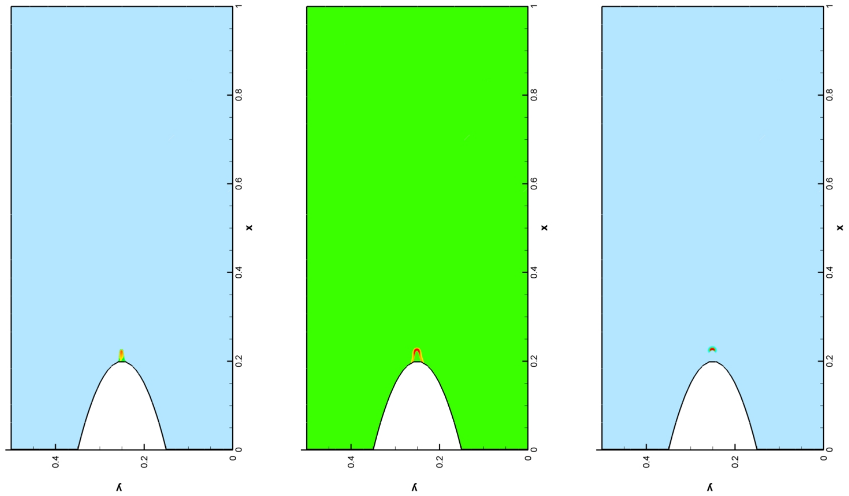

Figure 5.14: Electron density (left), net charge density (middle) and source term (right) at time $t = 1.155 \cdot 10^{-8}$ s.

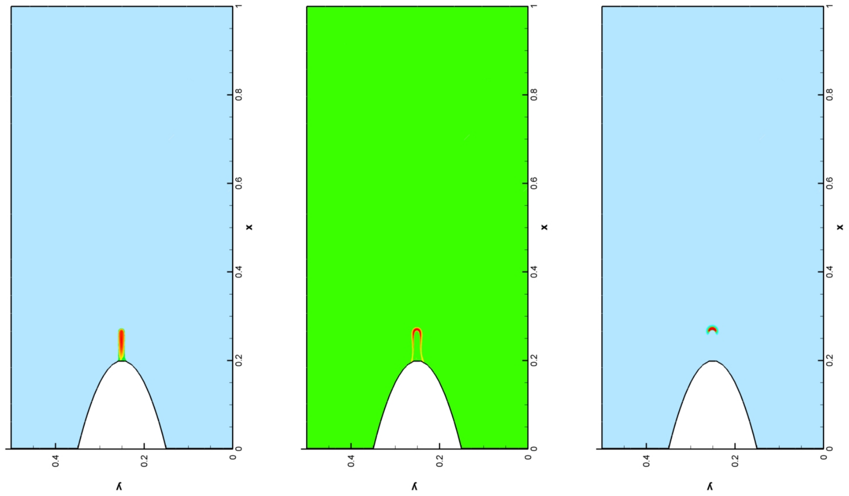

Figure 5.15: Electron density (left), net charge density (middle) and source term (right) at time $t = 1.47 \cdot 10^{-8}$ s.

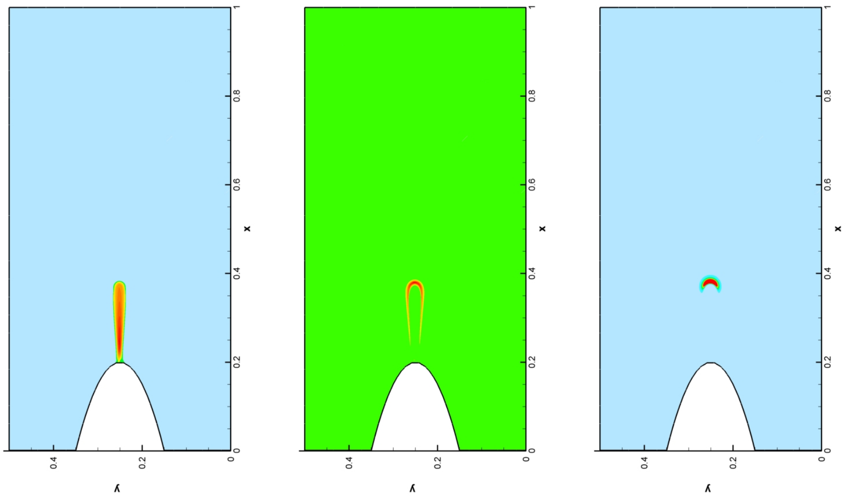

Figure 5.16: Electron density (left), net charge density (middle) and source term (right) at time $t = 1.89 \cdot 10^{-8}$ s.

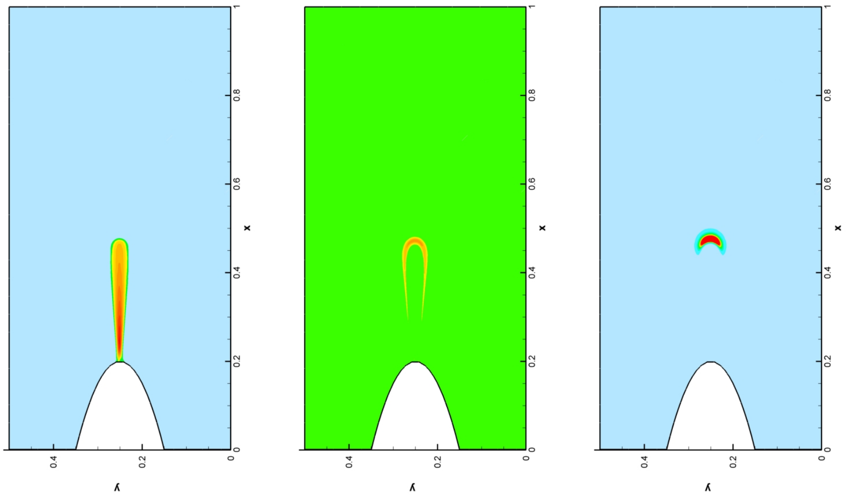

Figure 5.17: Electron density (left), net charge density (middle) and source term (right) at time $t = 2.1 \cdot 10^{-8}$ s.

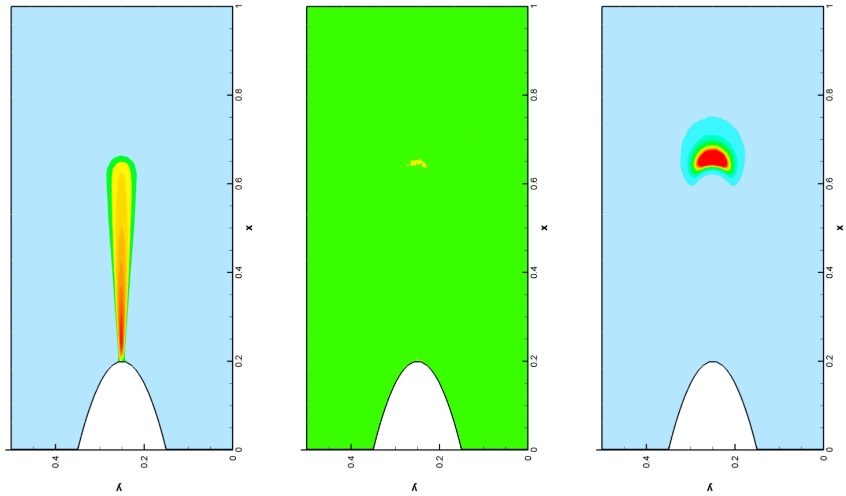

Figure 5.18: Electron density (left), net charge density (middle) and source term (right) at time $t = 2.31 \cdot 10^{-8}$ s.

Figs. 5.19-5.20 show the time evolution of the electron density, the net charge density and the source terms along the axis of the computational domain ($y = 0.25$).

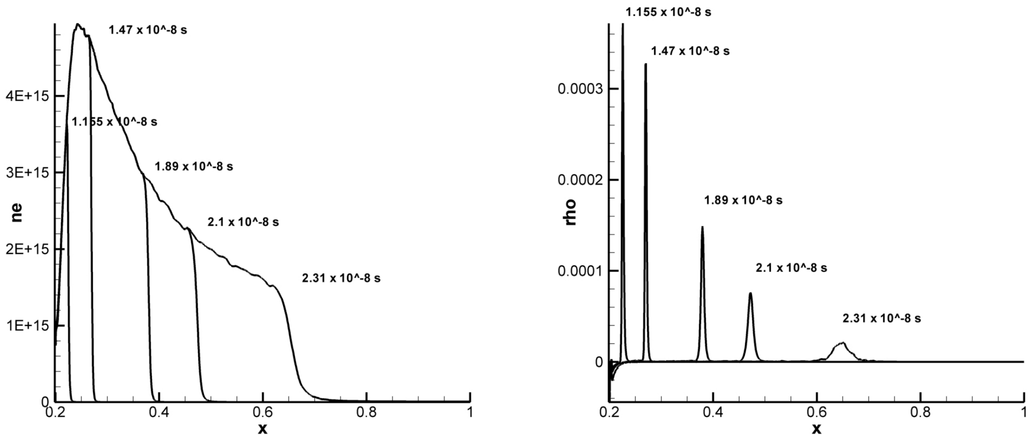

Figure 5.19: Evolution of electron density (left) and net charge density (left).

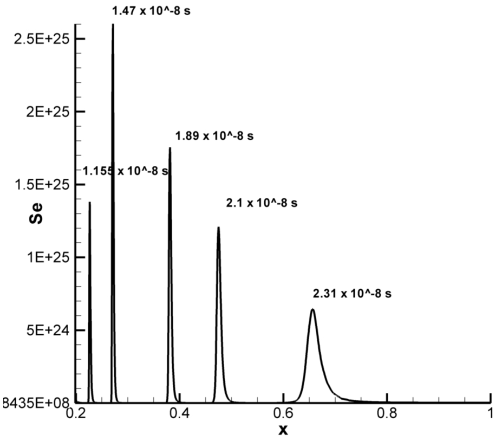

Figure 5.20: Evolution of source term.

Chapter 6

Numerical Results for 3D Streamer

6.1 Description of Test Cases

The best method from 2D simulations is adopted to the 3D experiments. It means treatment with the choice of parameters for the adaptation criterion, smoothing of criterion, using of the second order scheme. The algorithm V2 (see section 5.3) is used in adaptation criterion with 3 levels of refinement - level 3 fro $crit \in \langle 0.5625, 1\rangle$, level 2 for $crit \in \langle 0.25, 0.5625)$, level 1 for $\langle 0.0625, 0.25)$. The lower level of refinement is because of the CPU time and also the computer memory. The 3D simulations take almost 3 weeks to obtain results even with the dynamcal mesh adaptation. There are two different domains for experiments in3D. The first of them is a simple brick $1 \times 0.5 \times 0.5$ cm (see Fig. 6.1).

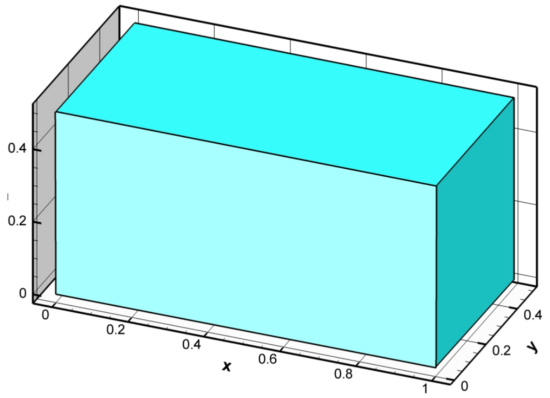

Figure 6.1: 3D computational domain

The second domain is a cube with edge's length 0.5 cm where the right boundary is deformed in such way that we obtain a pyramid with angle $\alpha = 161.82°$ at the top of the pyramid (see

Fig. 6.2).

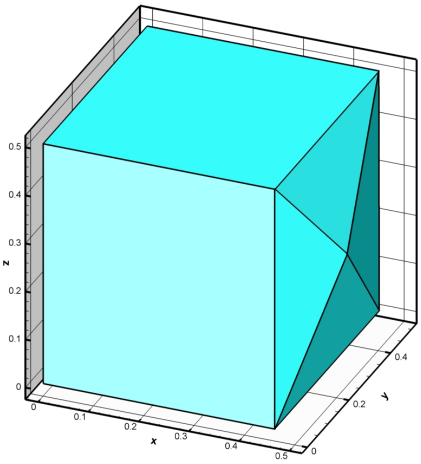

Figure 6.2: 3D computational domain - pyramid

Initial Conditions
We use the same initial conditions for all three computational domains

$$\begin{aligned} &n_e\left(x,y,z,0\right) = 10^{12} \cdot e^{-\frac{(x-0.2)^2+(y-0.25)^2}{\sigma^2}} + 10^8 \ [\mathrm{cm}^{-3}], \quad \sigma = 0.01, \\ &n_i\left(x,y,z,0\right) = n_e\left(x,y,0\right). \end{aligned} \tag{6.1}$$

The meaning of initial Gaussian pulse and the background electron and ion densities is same as in 2D. The increase of background electron and ion densities comes from axisymmetrical simulations [22] where a low magnitude of the densities have made blow up of the solution.

Boundary Conditions
The left boundary is an anode with high voltage, right boundary is a cathode. The homogeneous boundary conditions are prescribed for all unknowns on upper, lower front and back boundaries. Boundary conditions for the anode and the cathode are given by following formulas

$$\begin{aligned} &\frac{\partial n_e}{\partial \vec{n}} = 0, && \text{for anode and cathode,} \\ &V = 25000 \ [\mathrm{V}], && \text{for anode,} \\ &V = 0 \ [\mathrm{V}], && \text{for cathode.} \end{aligned} \tag{6.2}$$

The second domain ("pyramid") has half voltage on the anode

$$V = 12500 \ [\mathrm{V}], \ \text{for the anode.} \tag{6.3}$$

6.2 Streamer Propagation

Figures in this section shows results coming from computation in the domain where the initial disturbance of the electric field is due to the initial Gaussian pulse. The results are similar to the 2D computations. It demonstrate that we can trust to the method also in 3D simulations.

The mesh has been generated by GMSH mesh generator. The generator produced a mesh where the ratio between the smallest and the biggest cells is 1 : 10. The total number of cells in used mesh is around 150000. During the tuning of the code, it has been found out that the streamer simulation is sensitive on a bad quality of an used mesh. The old version of the GMSH generator produces grids where the ratio between the smallest and the biggest cells is $1 : 10^4$. When such grid is used, the method diverges.

The computation takes three weeks of CPU time. The number of iterations needed to obtain the results at the time $t = 2.52 \cdot 10^{-8}$ s is around 20500. The dynamical mesh adaptation leads to a mesh with around 450000 cells at the final time $t = 2.52 \cdot 10^{-8}$ s.

We can see a time evolution of the electron density in Figs. 6.3-6.5. The figures present isolines of the electron density in a cut plane $y = 0.25$ (left part of the figures) and full 3D view (right part of the figures).

A time evolution of the net charge density is shown in Figs. 6.6-6.8. As in previous set of figures, the left part of the figures presents isolines in the cut plane $y = 0.5$ and the right part of the figures shows full 3D view.

The results seem to be a little bit more oscillatory in the electron density n_e. 2D test presents that the numerical method has such behavior on coarse grids. The oscillatory can be also caused by a slightly different refinement algorithm in 3D and there is also different conformity step in the 3D. The cell dividing is different in 2D and 3D from the nature of the geometrical shapes of cells. A triangle is divided on the four same sub-triangles similar to the original one. The tetrahedron is divided on eight sub-tetrahedrons where four of them (tetrahedrons at vertices) are similar to the original one but the rest of the sub-tetrahedrons have different shape. This property together with the conformity step can deform the mesh. These properties will be tested in the future.

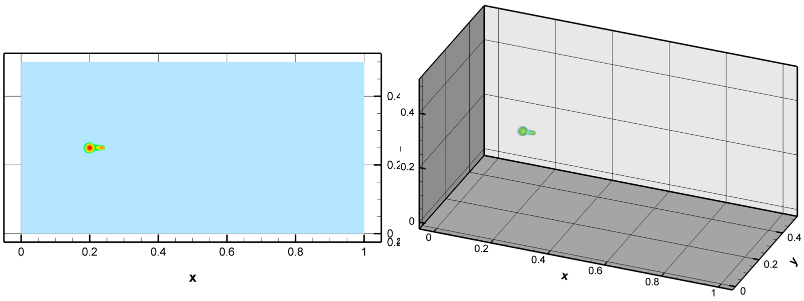

Figure 6.3: Electron density at time $t = 1.155 \cdot 10^{-8}$ s

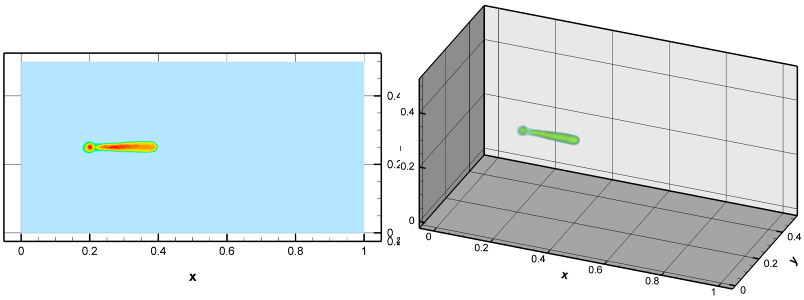

Figure 6.4: Electron density at time $t = 1.995 \cdot 10^{-8}$ s

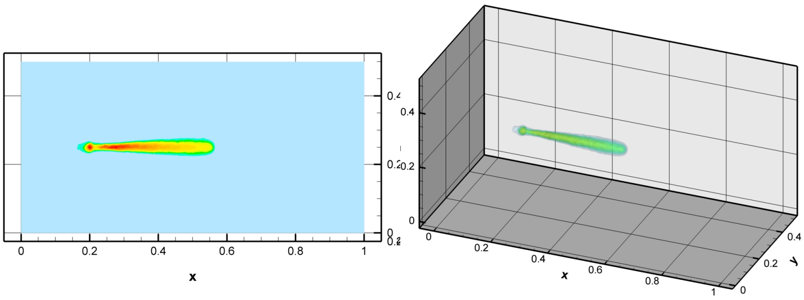

Figure 6.5: Electron density at time $t = 2.52 \cdot 10^{-8}$ s

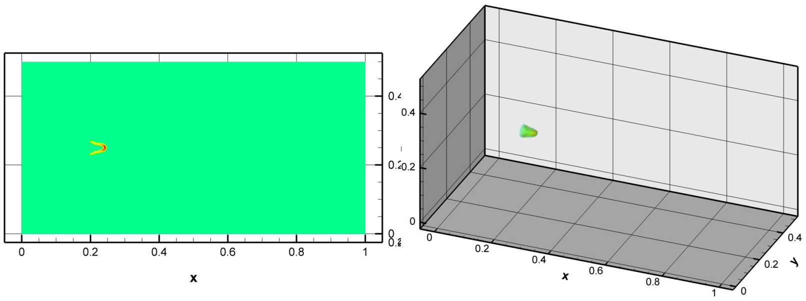

Figure 6.6: Net charge density at time $t = 1.155 \cdot 10^{-8}$ s

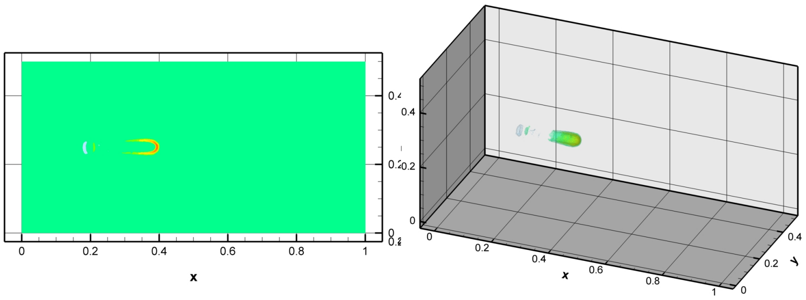

Figure 6.7: Net charge density at time $t = 1.995 \cdot 10^{-8}$ s

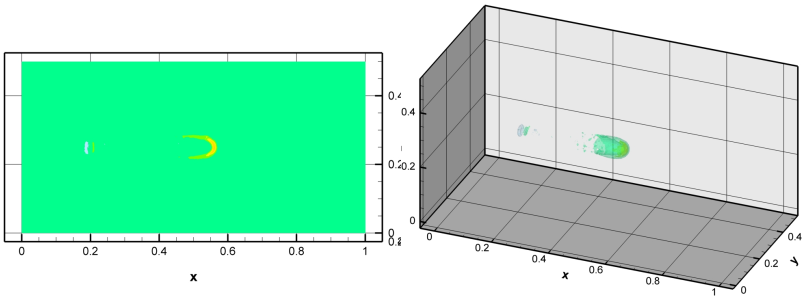

Figure 6.8: Net charge density at time $t = 2.52 \cdot 10^{-8}$ s

6.3 Streamer Propagation in the "Pyramid" Domain

Figures in this section are obtained by a computation in the "pyramid" domain (the cathode has a shape of a pyramid). Figs. 6.9-6.11 present a time evolution of the electron density and Figs. 6.12-6.14 show a time evolution of the net charge density. Results obtained by 2D computations in a domain with cathode which has a shape of "V" where the motivation for the test. This cathode caused ramification of discharge in 2D. Unfortunately the cathode shape doesn't have any influence on the ramification of the streamer in 3D simulations. The streamer discharge propagates without any signs of ramification toward the cathode. The domain with the "pyramid" cathode is next used for simulations of the streamer branching where a plasma spot is added to disturb the electric field (see section 6.4). The 2D results aren't included to the thesis.

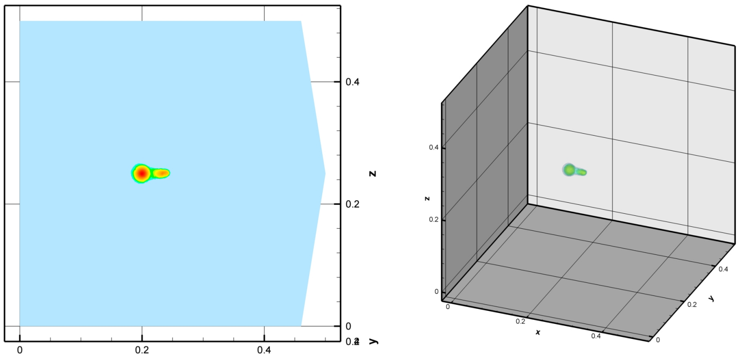

Figure 6.9: Electron density at time $t = 0.945 \cdot 10^{-8}$ s

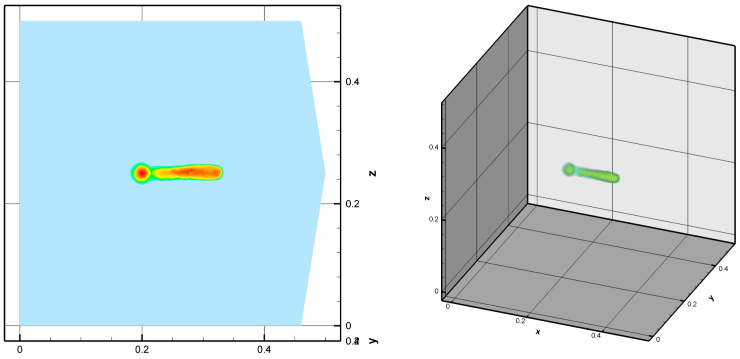

Figure 6.10: Electron density at time $t = 1.47 \cdot 10^{-8}$ s

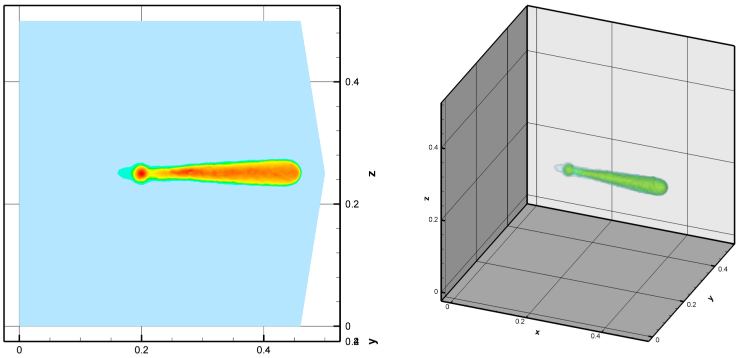

Figure 6.11: Electron density at time $t = 1.89 \cdot 10^{-8}$ s

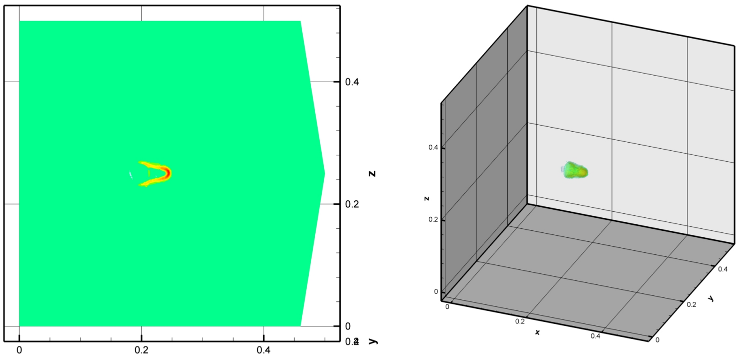

Figure 6.12: Net charge density at time $t = 0.945 \cdot 10^{-8}$ s

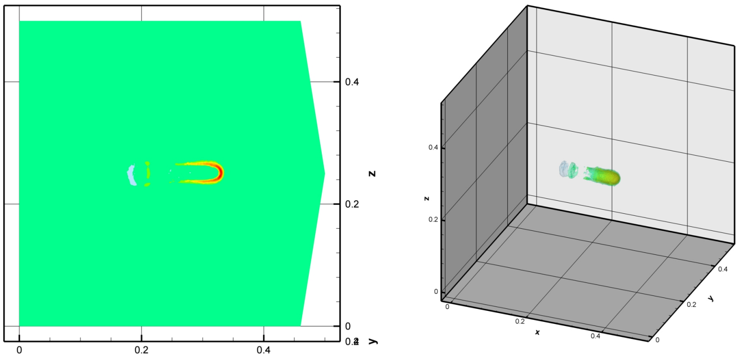

Figure 6.13: Net charge density at time $t = 1.47 \cdot 10^{-8}$ s

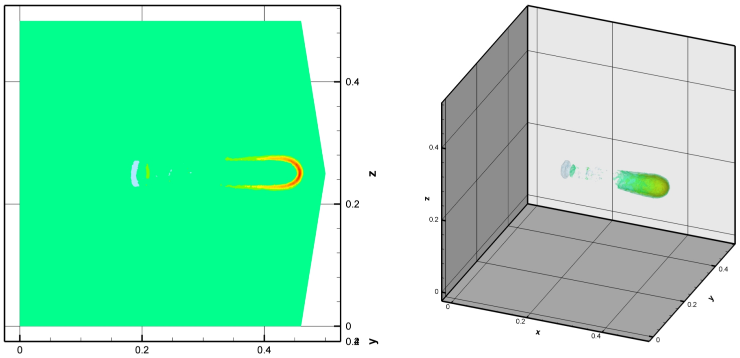

Figure 6.14: Net charge density at time $t = 1.89 \cdot 10^{-8}$ s

There is a comparison of the streamer motion in different domains in Fig. 6.15. The figure shows the electron density in different domains. We can see that the shape of a domain doesn't have significant influence on the physical phenomenon. The shape of the cathode has an influence on the velocity of the motion. The results aren't compared with laboratory experiments.

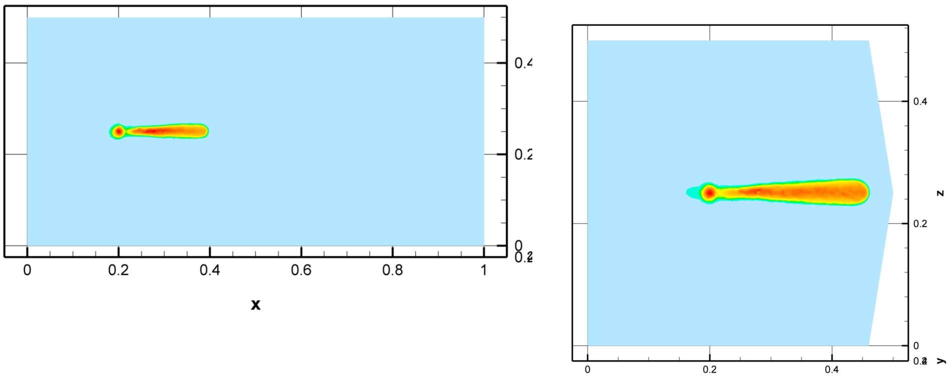

Figure 6.15: Electron density at time $t = 1.995 \cdot 10^{-8}$ s (left), electron density at time $t = 1.89 \cdot 10^{-8}$ s in "pyramid" domain(right)

6.4 Streamer Branching

The tested streamer branching is caused by an instability in the electric field. The effect of plasma spot [16] is applied near the head of the streamer in this work. We add neutral plasma spot which is introduced into the model as another source term into the continuity equations for the electron and the positive ions densities. The plasma spot is canceled after short time (less than nanosecond).

Figs. 6.16-6.107 show results of four test versions when one or two plasma spots were added. Tests are computed in the "pyramid" domain. The figures are zoomed around the streamer paths.

The plasma spot is added at time $t = 1.26 \cdot 10^{-8}$ s and canceled after the "duration time". In *Version 3* and *Version 4*, the second plasma spot is added when the first one is canceled.

Version 1
Plasma spot: $S_e = 10^{21} \cdot e^{\frac{(x-0.3)^2+(y-0.25)^2+(z-0.265)^2}{0.005^2}}$
Position: $X_0 = [0.3,\ 0.25,\ 0.265]$
Duration: $t = 0.5 \cdot 10^{-9}$ s

Figs. 6.16-6.35 present an evolution of the electron density and the net charge density where the neutral plasma spot is introduced too close the streamer head. Figs. 6.16-6.20 and 6.26-6.30 show isolines in a cut plane $y = 0.25$, Figs. 6.21-6.25 and 6.31-6.35 show full 3D view. Figs. 6.16, 6.21, 6.26 and 6.31 show the state when the plasma spot is added to the electric field. Figs. 6.17, 6.22, 6.27 and 6.32 captures a moment when the plasma spot is canceled. We can observe that the electron from the streamer head are attached to the area of the plasma spot. Figs. 6.18, 6.23, 6.28 and 6.33 show that the streamer motion is deflected to the area of the plasma spot. Figs. 6.19, 6.20, 6.24, 6.25, 6.29, 6.30, 6.34 and 6.35 display that the streamer avalanche next moves through the the area of the plasma spot toward the cathode. We can see that introducing of the plasma spot too close the streamer head causes only a deflection of the streamer motion. Fig. 6.36 shows an adapted mesh around the streamer path in the cut plane $y = 0.25$ at the time $t = 1.89 \cdot 10^{-8}$ s. The mesh has around 440000 cells at the final time and it's been needed 28500 time steps to obtain the results.

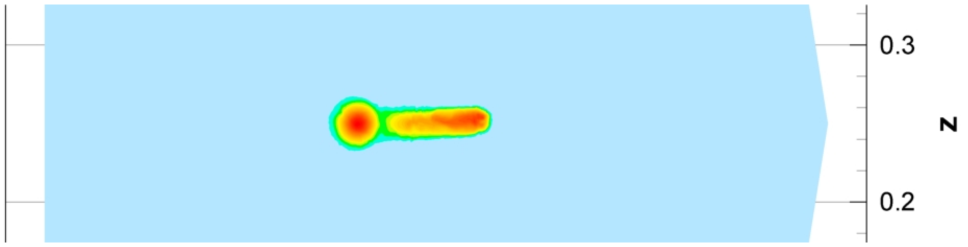

Figure 6.16: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

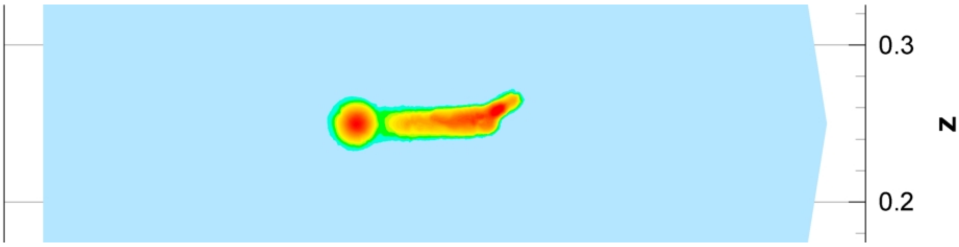

Figure 6.17: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

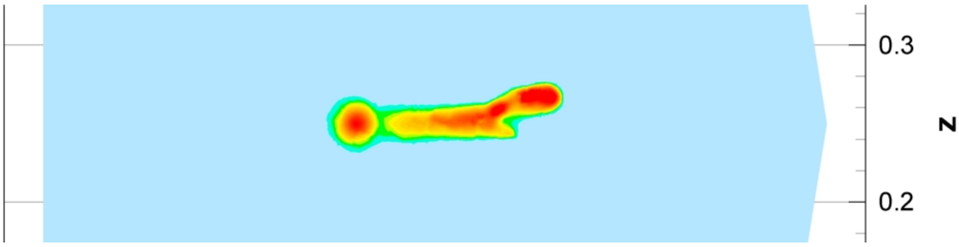

Figure 6.18: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

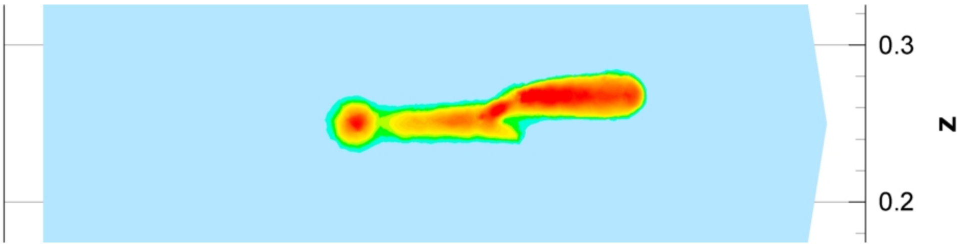

Figure 6.19: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

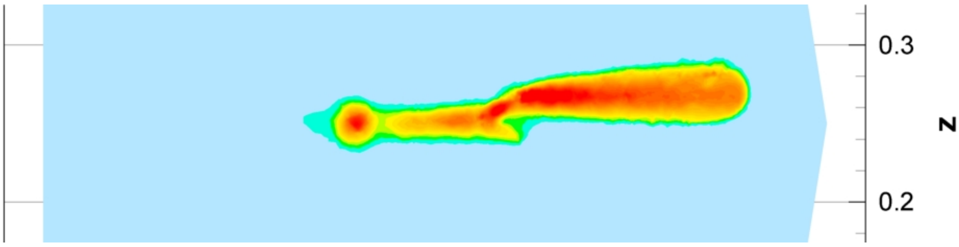

Figure 6.20: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

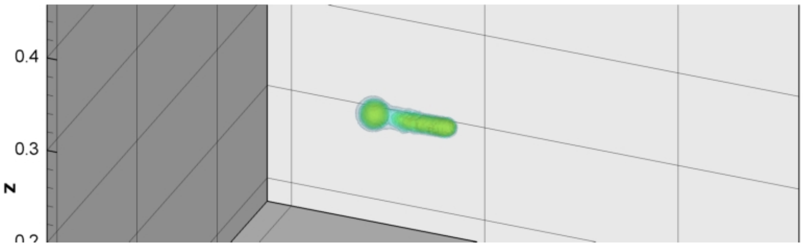

Figure 6.21: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

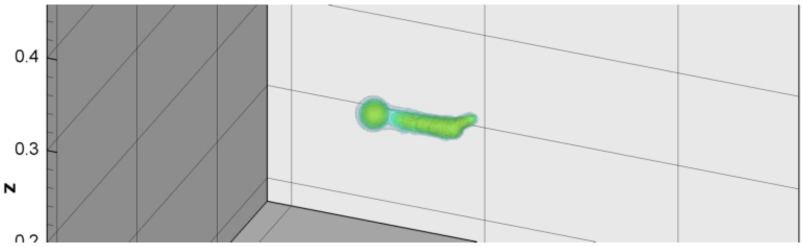

Figure 6.22: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

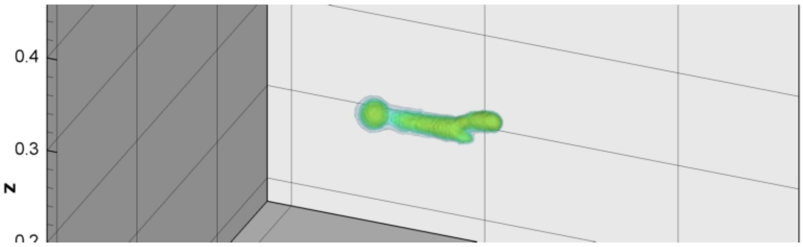

Figure 6.23: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

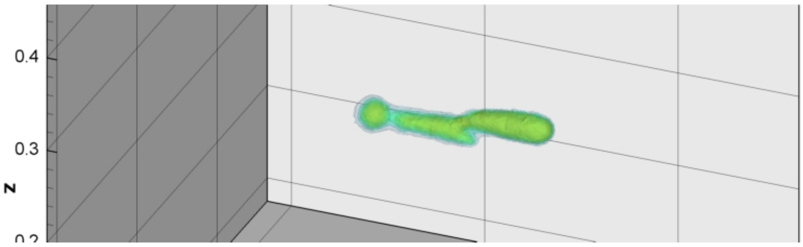

Figure 6.24: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

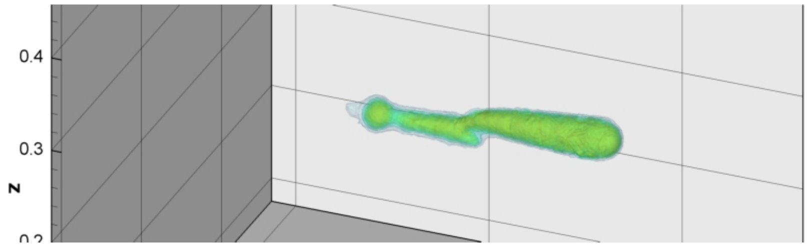

Figure 6.25: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

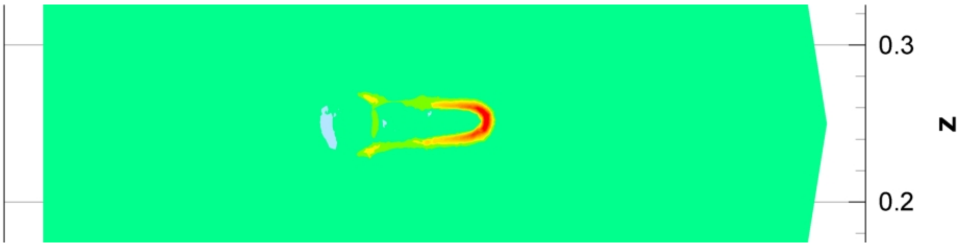

Figure 6.26: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

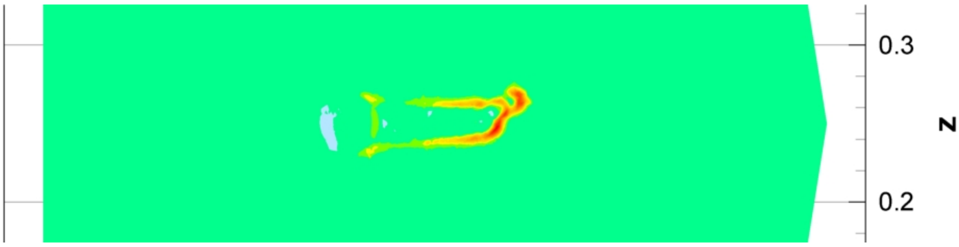

Figure 6.27: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

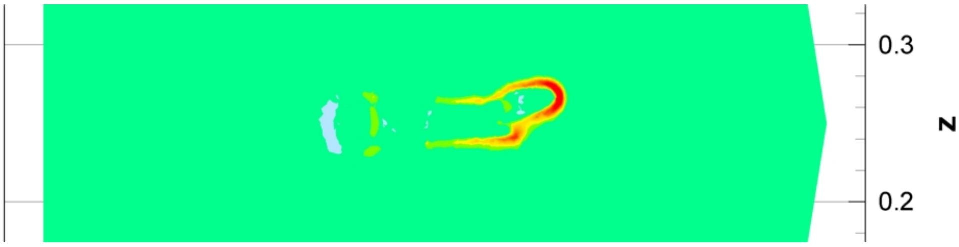

Figure 6.28: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

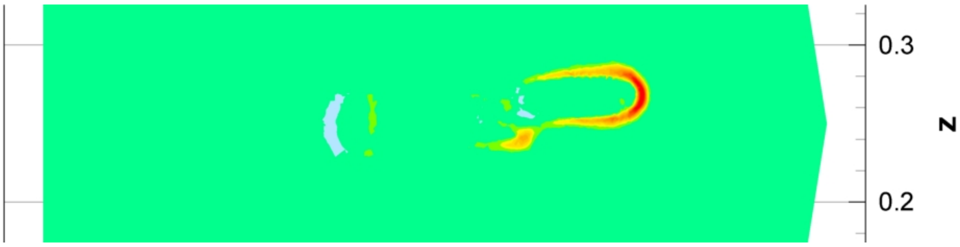

Figure 6.29: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

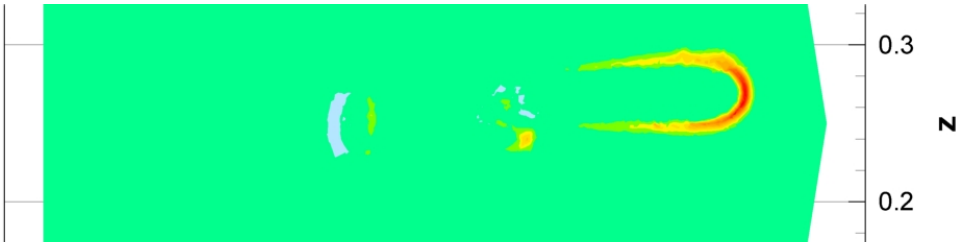

Figure 6.30: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

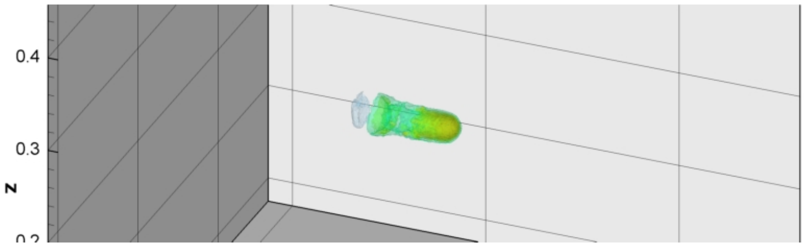

Figure 6.31: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

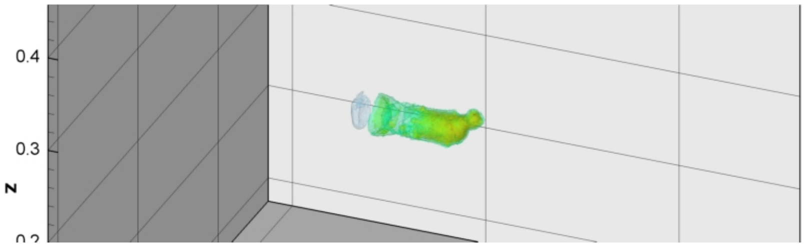

Figure 6.32: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

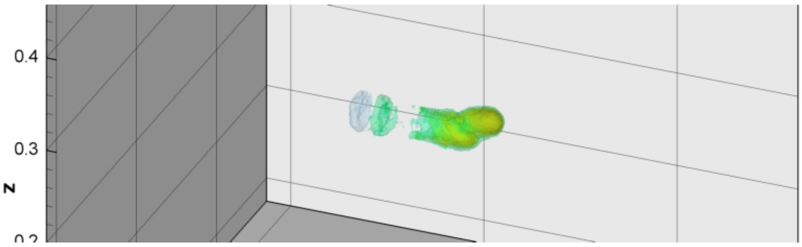

Figure 6.33: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

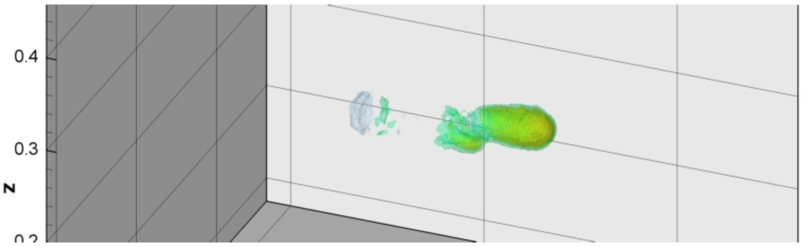

Figure 6.34: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

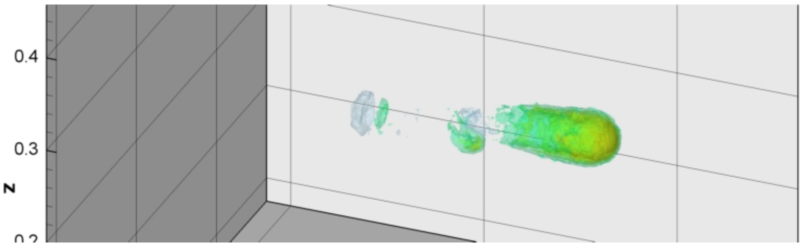

Figure 6.35: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

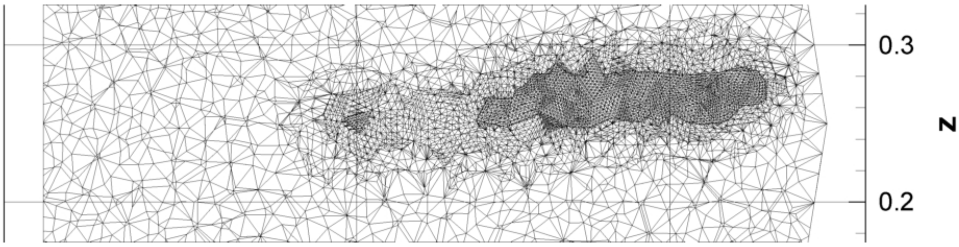

Figure 6.36: Mesh at time $t = 1.89 \cdot 10^{-8}$ s in a cut plane $y = 0.25$

Version 2

Plasma spot: $S_e = 10^{21} \cdot e^{\frac{(x-0.3)^2+(y-0.25)^2+(z-0.28)^2}{0.005^2}}$
Position: $X_0 = [0.3,\ 0.25,\ 0.28]$
Duration: $t = 0.5 \cdot 10^{-9}$ s

An evolution of the streamer branching, where the plasma spot is introduced in a double distance from the streamer filament axis, is presented in Figs. 6.37 and 6.56. The evolution of the electron avalanche is in Figs. 6.37-6.46. Figs. 6.47-6.56 show the evolution of the net charge density. As in the previous case, the first row in the figures show the streamer just in the moment when the plasma spot is applied to the electric field. The moment when the plasma spot is abolished is presented in the second row. The electrons from the streamer filament are attached to the plasma spot area and we can see that the electrons also continue in the straight direction. We can observe a branch coming from the main filament in the third row. The fourth and fifth rows show that the main streamer discharge still propagates toward the cathode while the propagation in the branch almost stops. An adapted mesh in the planar cut at the time $t = 1.89 \cdot 10^{-8}$ s is plotted in Fig. 6.57. The mesh size is 660000 cells and the computation has made 31000 time steps.

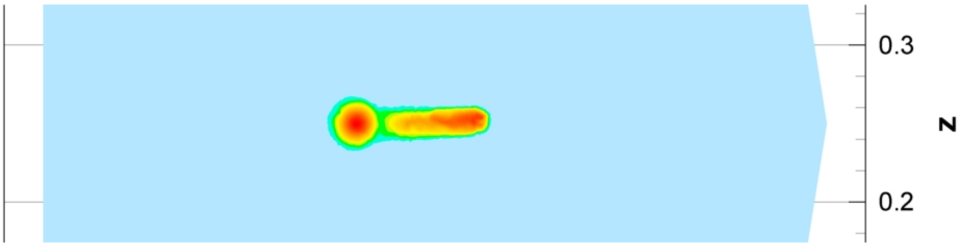

Figure 6.37: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

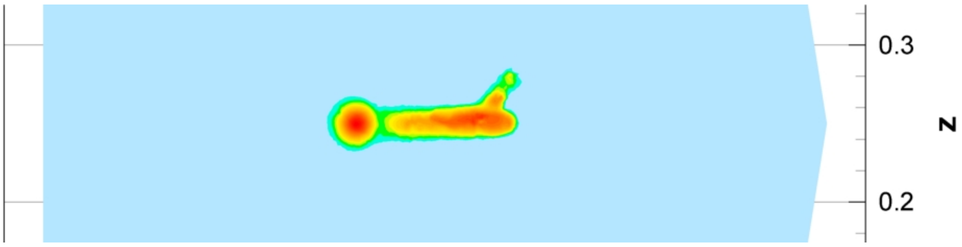

Figure 6.38: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

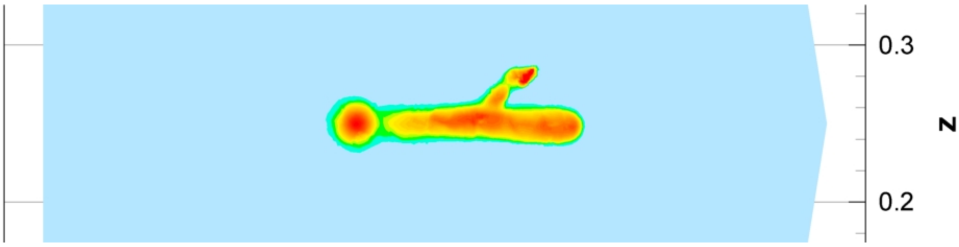

Figure 6.39: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

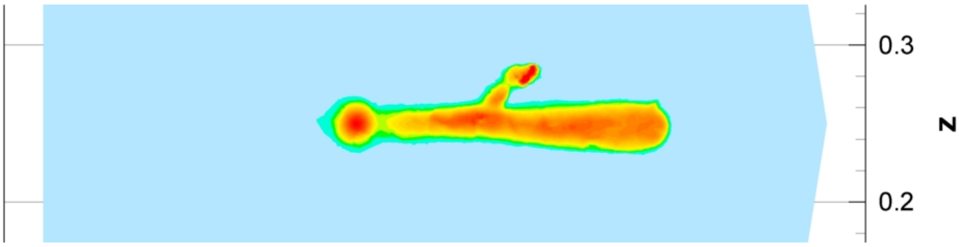

Figure 6.40: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

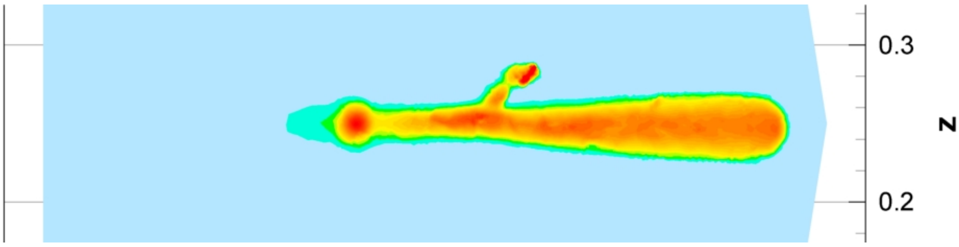

Figure 6.41: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

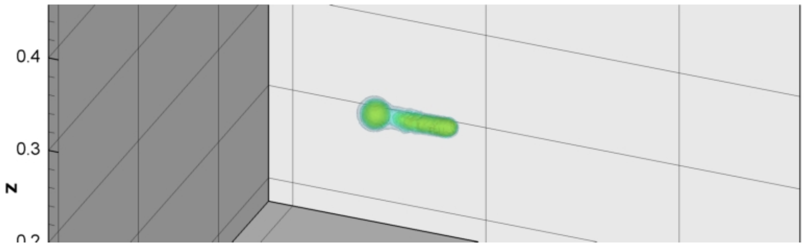

Figure 6.42: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

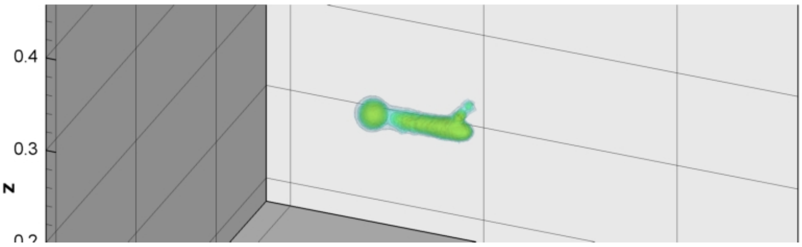

Figure 6.43: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

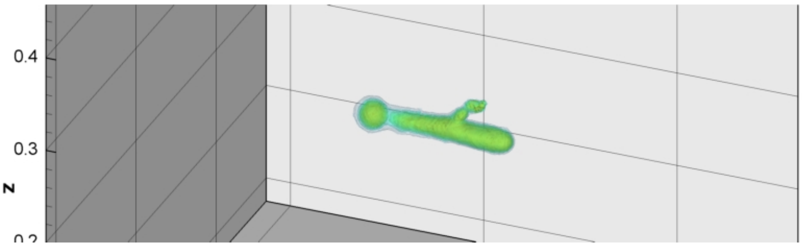

Figure 6.44: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

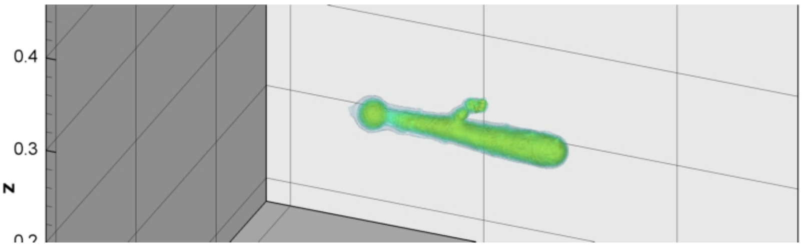

Figure 6.45: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

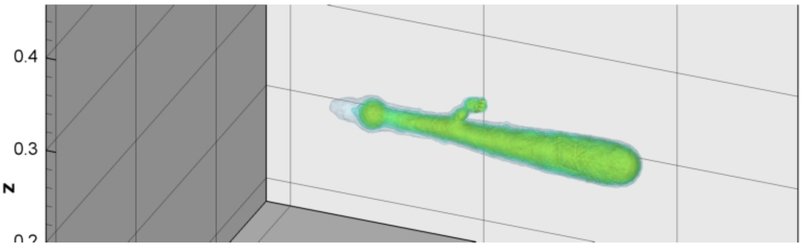

Figure 6.46: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

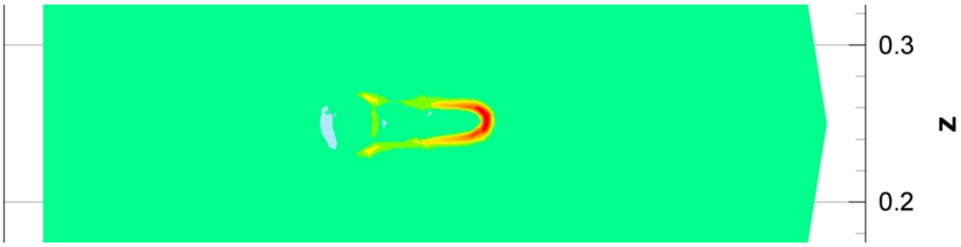

Figure 6.47: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

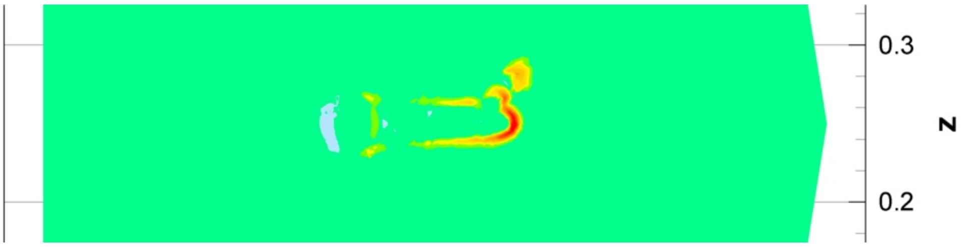

Figure 6.48: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

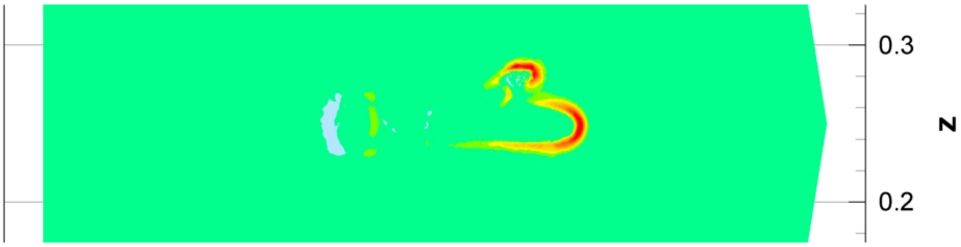

Figure 6.49: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

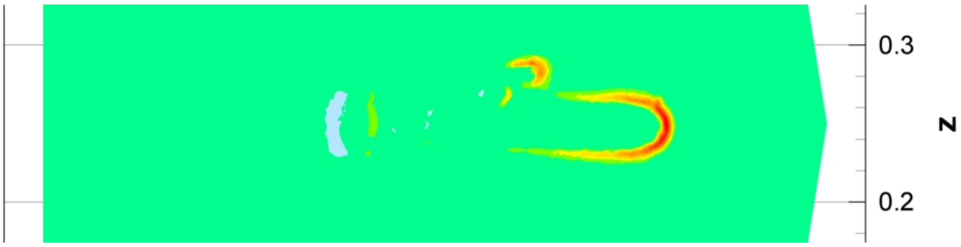

Figure 6.50: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

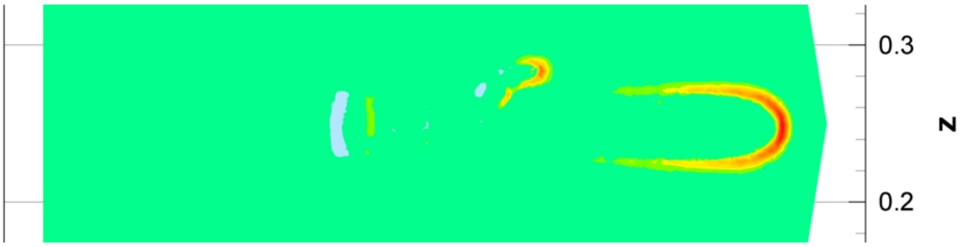

Figure 6.51: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

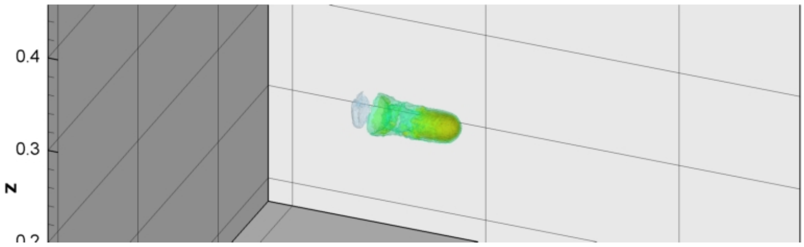

Figure 6.52: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

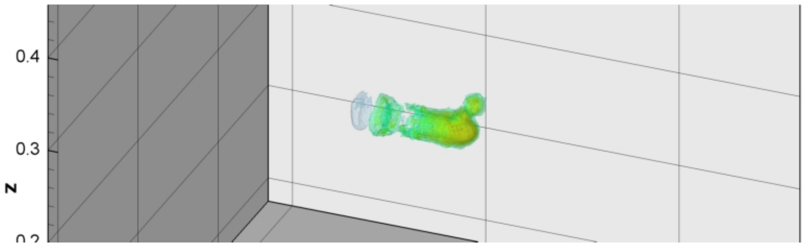

Figure 6.53: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

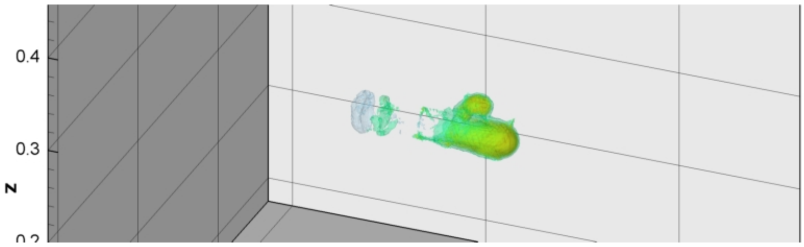

Figure 6.54: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

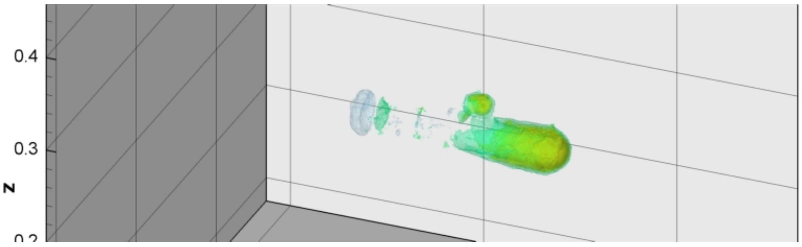

Figure 6.55: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

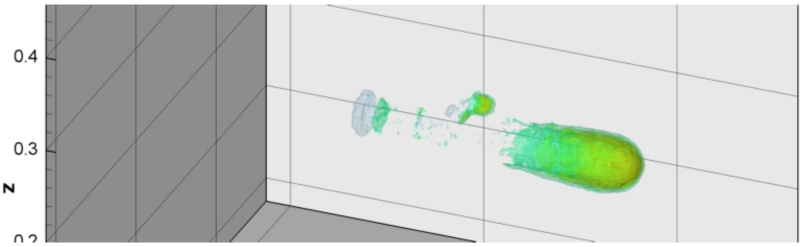

Figure 6.56: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

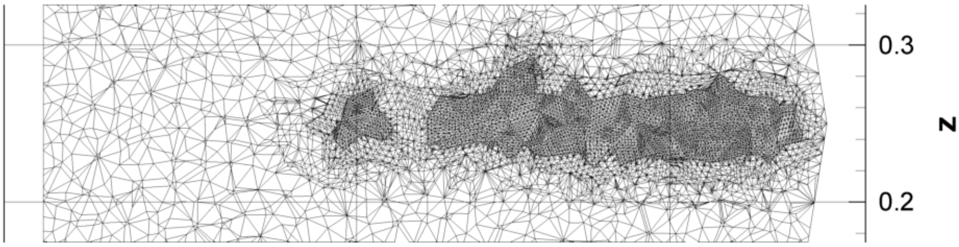

Figure 6.57: Mesh at time $t = 1.89 \cdot 10^{-8}$ s in a cut plane $y = 0.25$

Version 3

1^{st} *plasma spot:* $S_e = 10^{21} \cdot e^{\frac{(x-0.3)^2+(y-0.25)^2+(z-0.28)^2}{0.005^2}}$
Position: $X_{01} = [0.3,\ 0.25,\ 0.28]$
Duration: $t_1 = 0.5 \cdot 10^{-9}$ s
2^{nd} *plasma spot:* $S_e = 10^{21} \cdot e^{\frac{(x-0.315)^2+(y-0.25)^2+(z-0.295)^2}{0.005^2}}$
Position: $X_{02} = [0.315,\ 0.25,\ 0.295]$
Duration: $t_2 = 0.5 \cdot 10^{-9}$ s

The results, where two plasma spots are sequentially added, are plotted in Fig. 6.58- 6.81. Both plasma spots lie above the streamer filament. The electron avalanche evolution is presented in Figs. 6.58-6.69 and the net charge evolution is in Figs. 6.70-6.81. The introducing of the first plasma spot is shown in Figs. 6.58, 6.64, 6.70 and 6.76. Figs. 6.59, 6.65, 6.71 and 6.77 present a moment when the first plasma spot is voided and the second plasma spot is added to the electric field. Figs. 6.60, 6.66, 6.72 and 6.78 display a moment when the second plasma spot is abolished. We can see the electron motion toward to the first plasma spot place. We observe the progression toward the second spot and the main filament slightly deflects from the straight direction. The last figures in the sets show following the streamer evolution. Two plasma spots on the same side from the main streamer filament produce a longer branch than the case where only one plasma spot is added. On the other hand the propagation velocity of the main filament is slower than in the case with one plasma spot. The adapted mesh around the branched streamer at the time $t = 1.89 \cdot 10^{-8}$ s is plotted in Fig. 6.82. There is 570000 cells in the mesh. There has been needed 28900 time steps to obtain the results at the time $t = 1.89 \cdot 10^{-8}$ s.

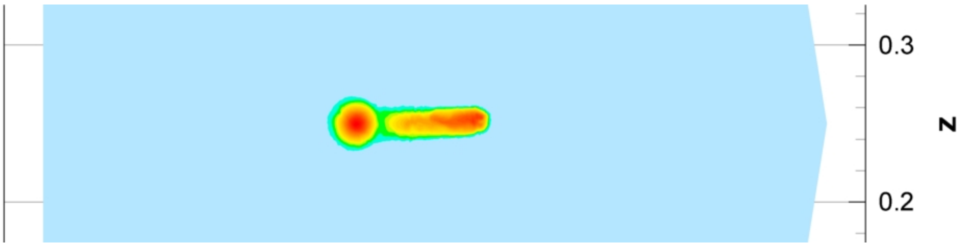

Figure 6.58: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

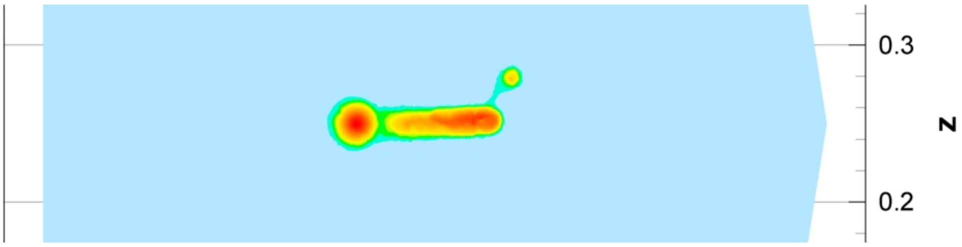

Figure 6.59: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

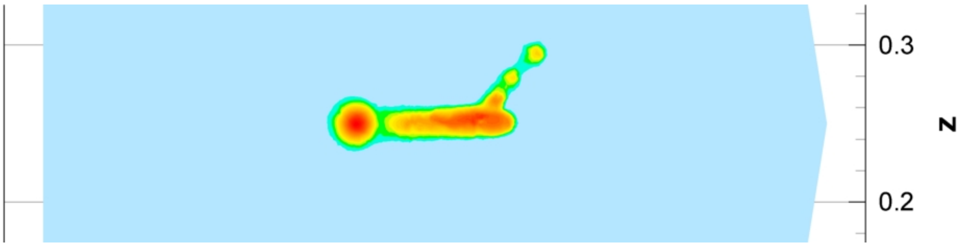

Figure 6.60: Electron density at times $t_1 = 1.36 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

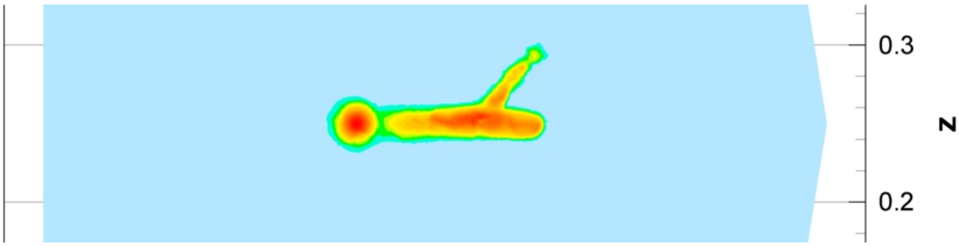

Figure 6.61: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

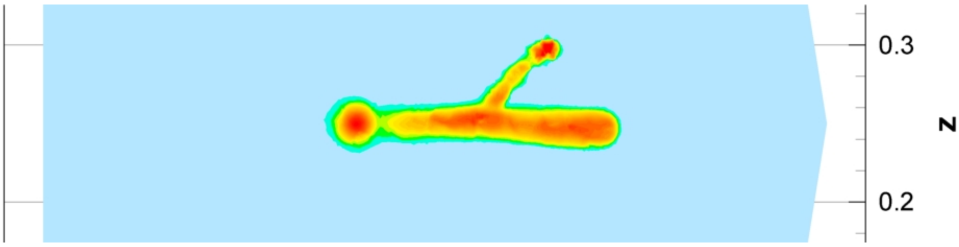

Figure 6.62: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

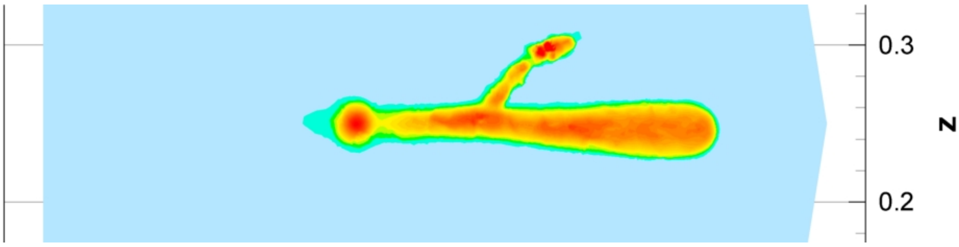

Figure 6.63: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

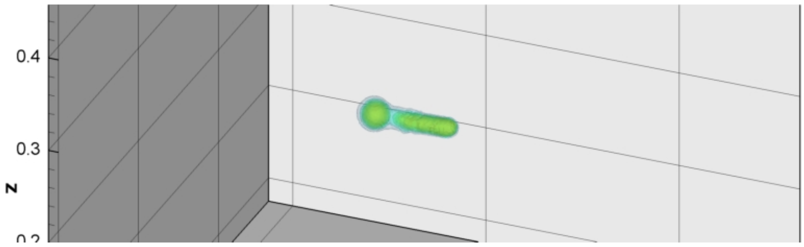

Figure 6.64: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

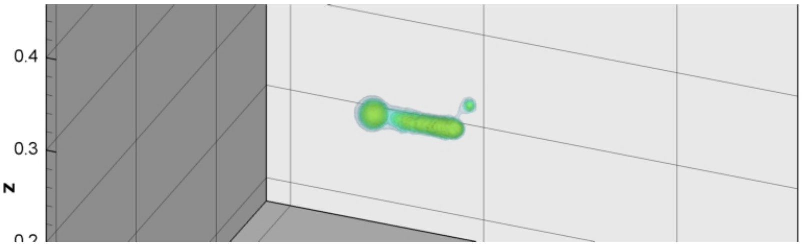

Figure 6.65: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

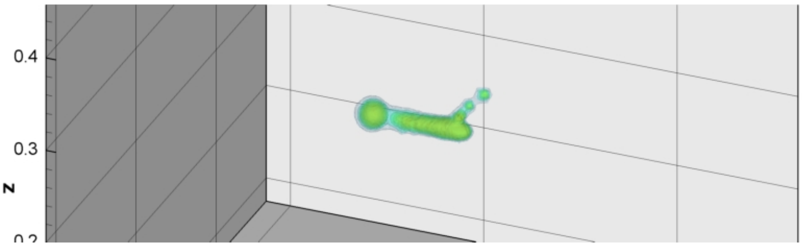

Figure 6.66: Electron density at times $t_1 = 1.36 \cdot 10^{-8}$ s, 3D view

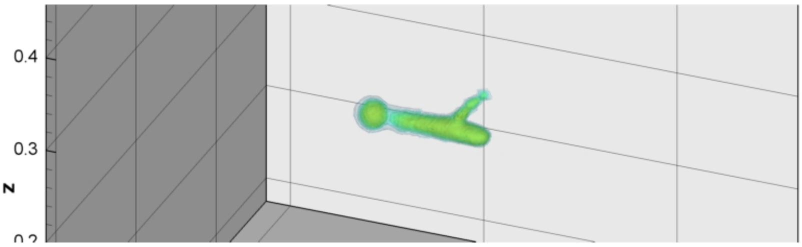

Figure 6.67: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

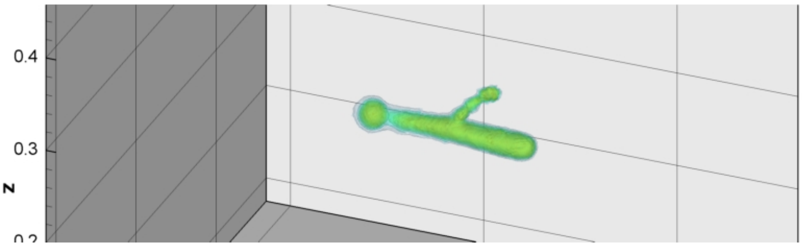

Figure 6.68: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

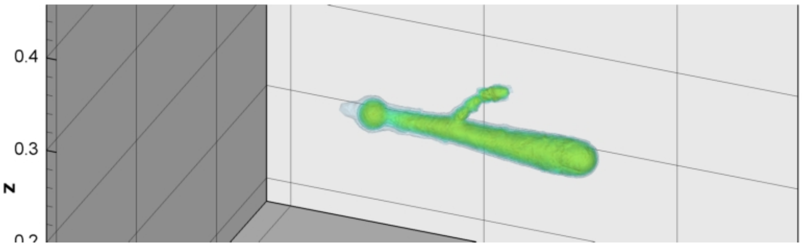

Figure 6.69: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

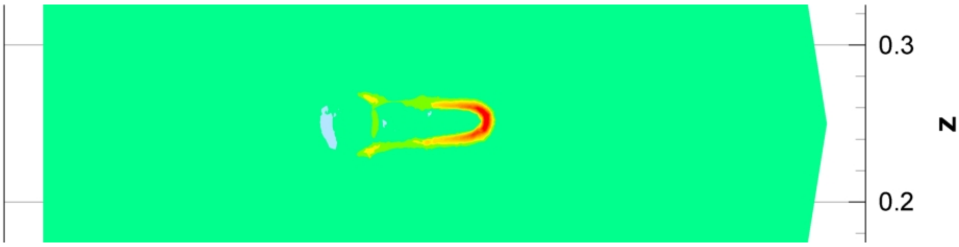

Figure 6.70: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

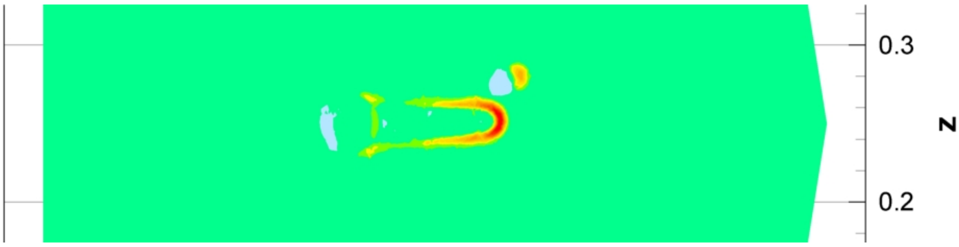

Figure 6.71: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

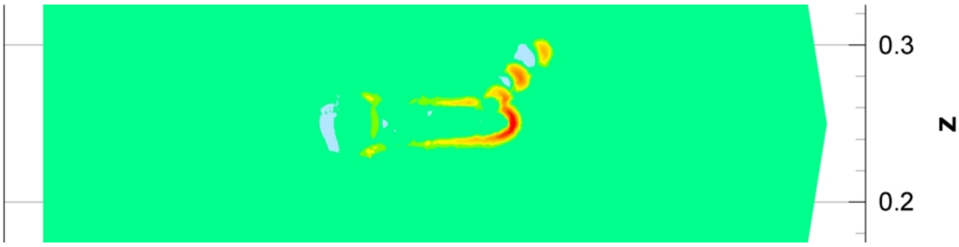

Figure 6.72: Net charge density at times $t_1 = 1.36 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

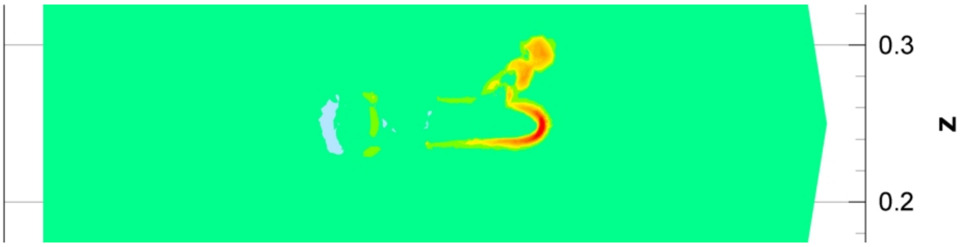

Figure 6.73: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

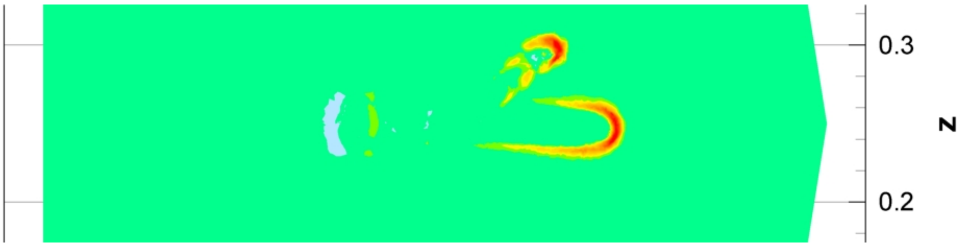

Figure 6.74: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

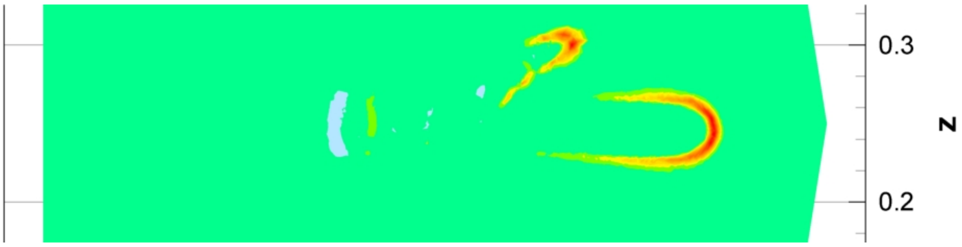

Figure 6.75: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

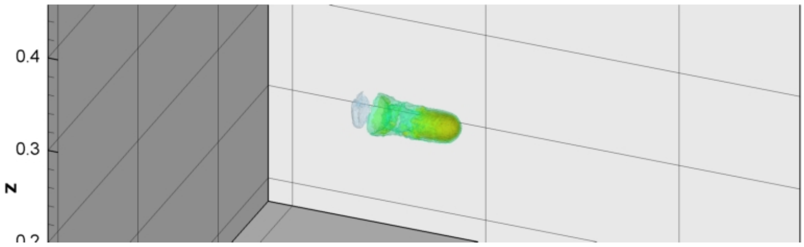

Figure 6.76: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

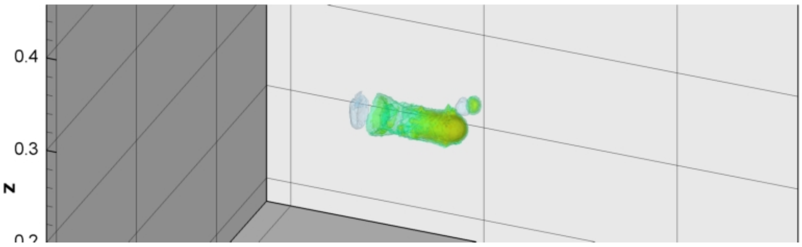

Figure 6.77: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

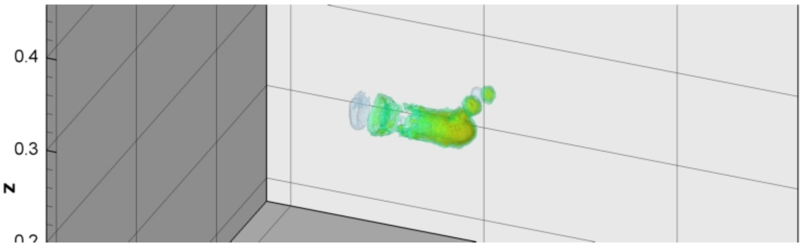

Figure 6.78: Net charge density at times $t_1 = 1.36 \cdot 10^{-8}$ s, 3D view

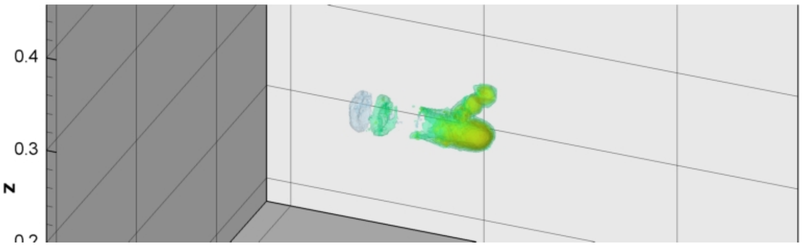

Figure 6.79: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

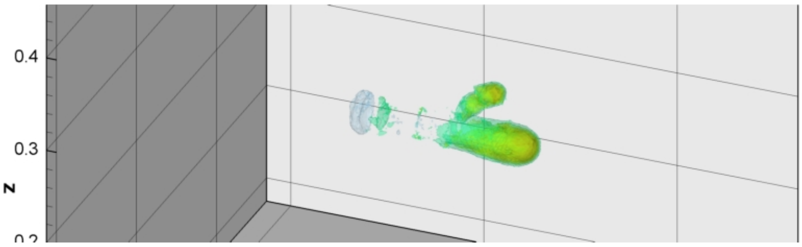

Figure 6.80: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

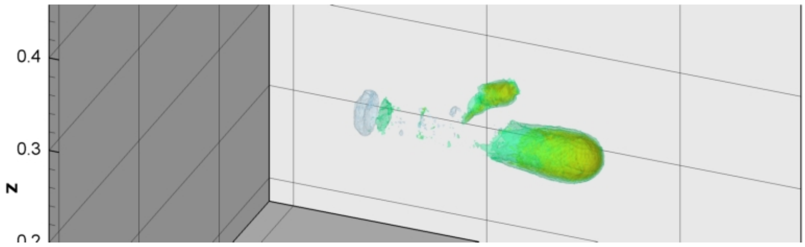

Figure 6.81: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

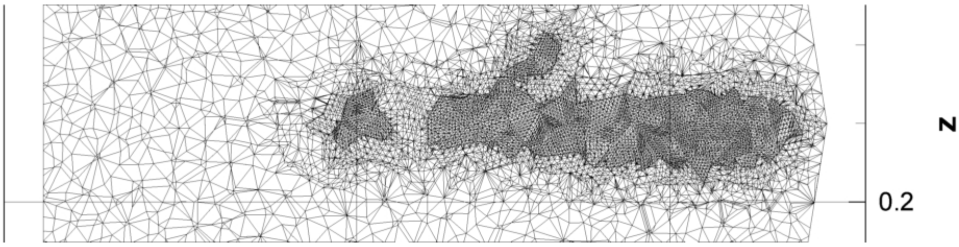

Figure 6.82: Mesh at time $t = 1.89 \cdot 10^{-8}$ s in a cut plane $y = 0.25$

Version 4

1^{st} *plasma spot:* $S_e = 10^{21} \cdot e^{\frac{(x-0.3)^2+(y-0.25)^2+(z-0.28)^2}{0.005^2}}$
Position: $X_{01} = [0.3,\ 0.25,\ 0.28]$
Duration: $t_1 = 0.5 \cdot 10^{-9}$ s
2^{nd} *plasma spot:* $S_e = 10^{21} \cdot e^{\frac{(x-0.31)^2+(y-0.25)^2+(z-0.22)^2}{0.005^2}}$
Position: $X_{02} = [0.31,\ 0.25,\ 0.22]$
Duration: $t_2 = 0.5 \cdot 10^{-9}$ s

The last test simulate conditions when two independent plasma spots are added to the electric field in two different times. The first spot is placed above the main streamer filament and the second plasma spot is placed under the streamer filament. Results of the test are plotted in Figs. 6.83- 6.106. Figs. 6.83, 6.89, 6.95 and 6.101 have the same meaning as in previous cases. Figs. 6.84, 6.90, 6.96 and 6.102 show the streamer when the first spot is canceled and the second spot is introduced. There is a state when the second plasma spot is abolished in Figs. 6.85, 6.91, 6.97 and 6.103. We see that the first branch begins here. The beginning of the second branch is plotted in Figs. 6.86, 6.92, 6.98 and 6.104. The last figures in the sets present the progression of the main streamer filament and the branches. Fig. 6.107 shows the adapted mesh at the time $t = 1.89 \cdot 10^{-8}$ s. The mesh contains around 540000 cells. We have needed 28300 time steps to obtain the results at the final time.

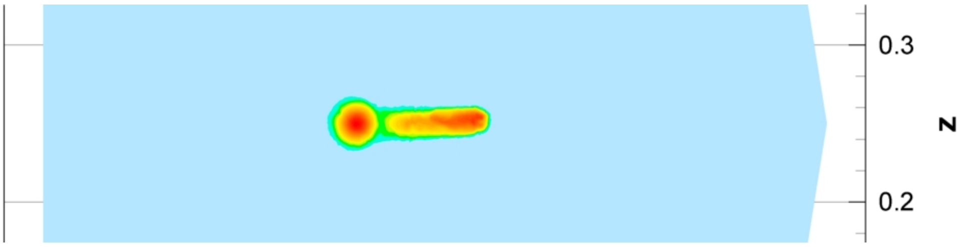

Figure 6.83: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

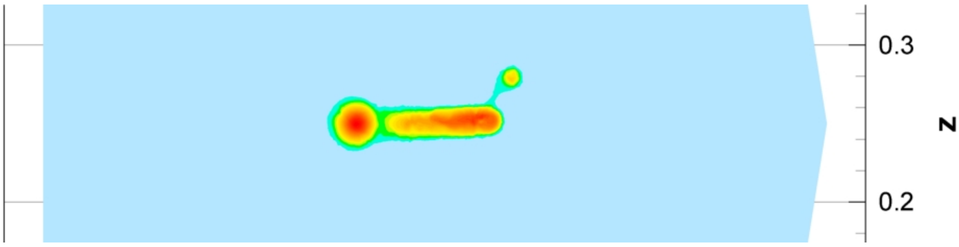

Figure 6.84: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

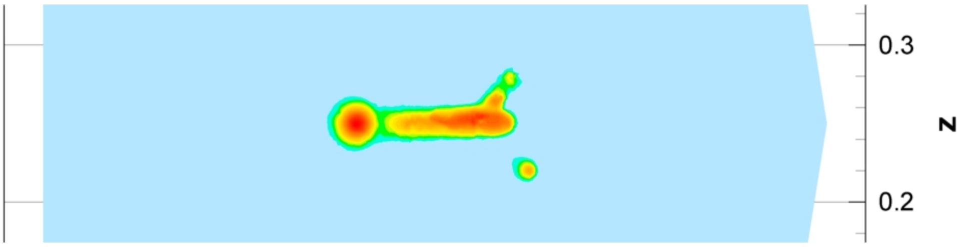

Figure 6.85: Electron density at times $t_1 = 1.36 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

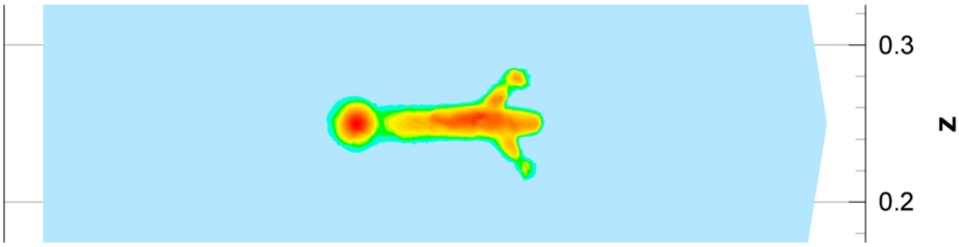

Figure 6.86: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

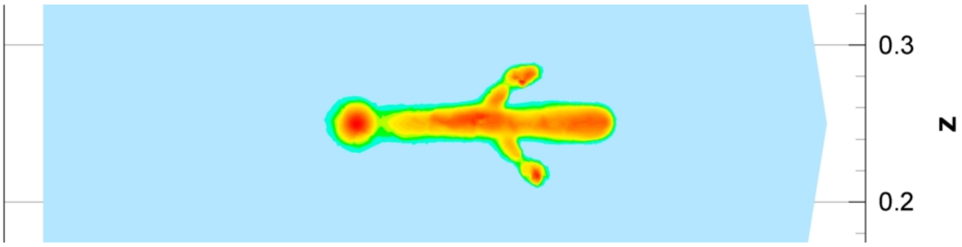

Figure 6.87: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

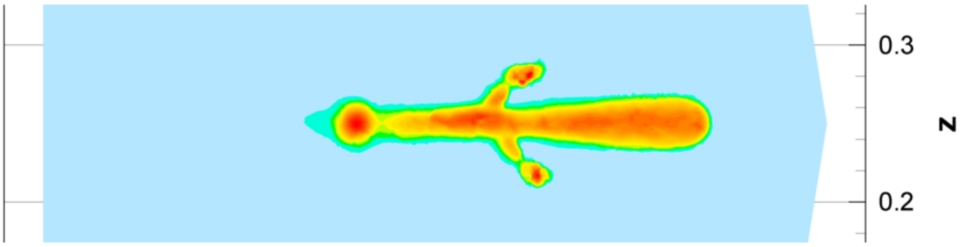

Figure 6.88: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

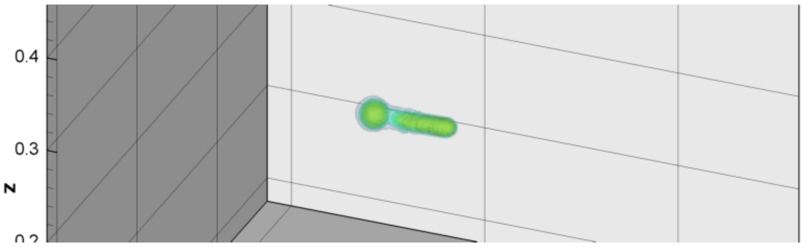

Figure 6.89: Electron density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

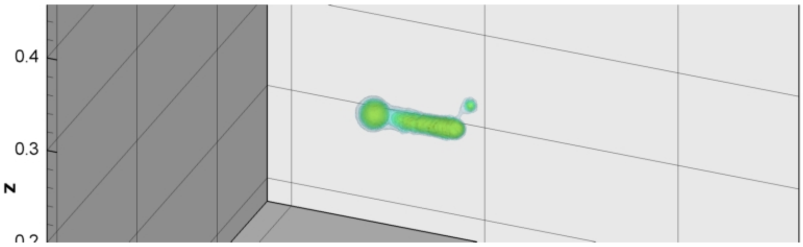

Figure 6.90: Electron density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

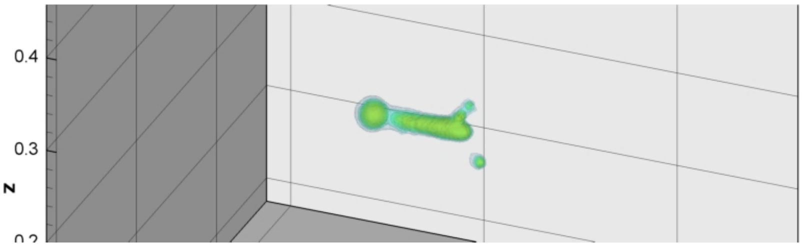

Figure 6.91: Electron density at times $t_1 = 1.36 \cdot 10^{-8}$ s, 3D view

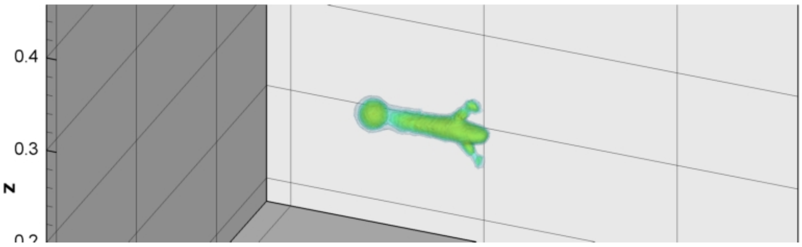

Figure 6.92: Electron density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

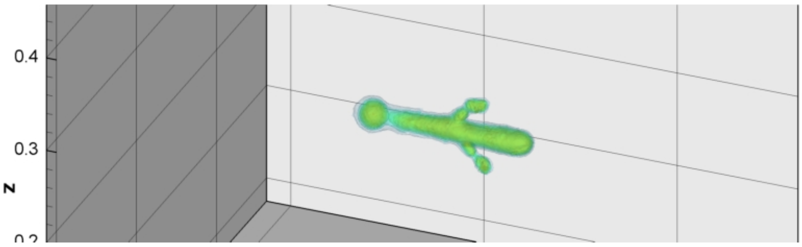

Figure 6.93: Electron density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

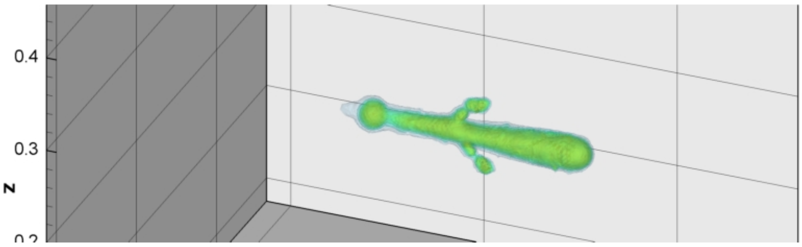

Figure 6.94: Electron density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

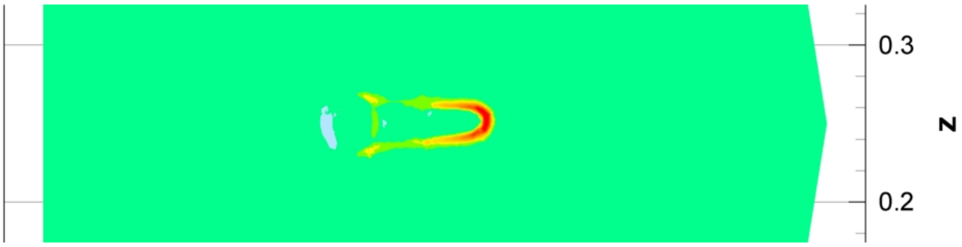

Figure 6.95: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

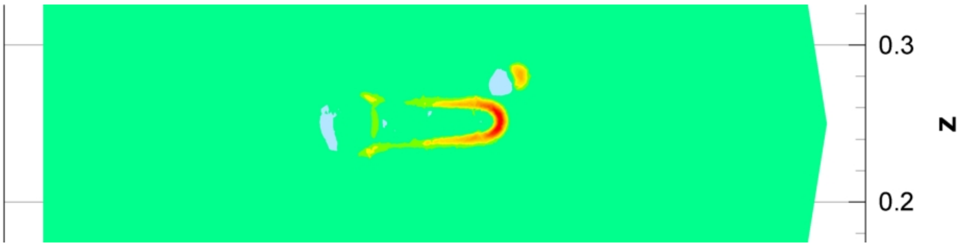

Figure 6.96: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

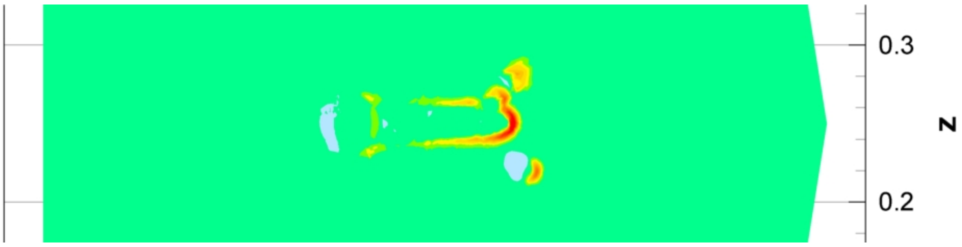

Figure 6.97: Net charge density at times $t_1 = 1.36 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

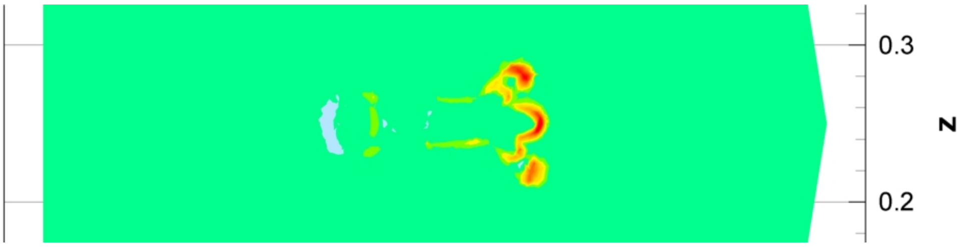

Figure 6.98: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

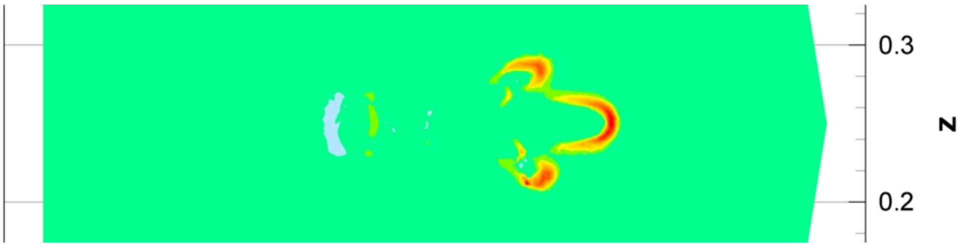

Figure 6.99: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

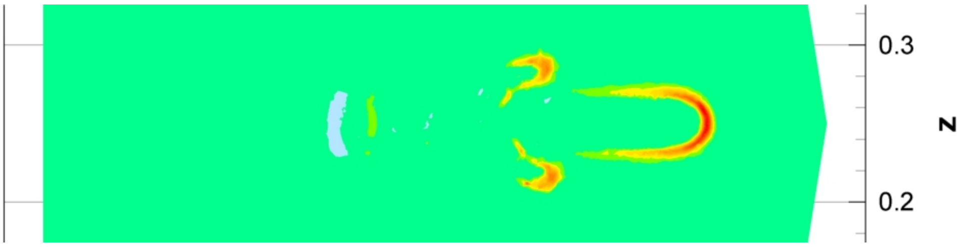

Figure 6.100: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, isolines in a cut plane $y = 0.25$

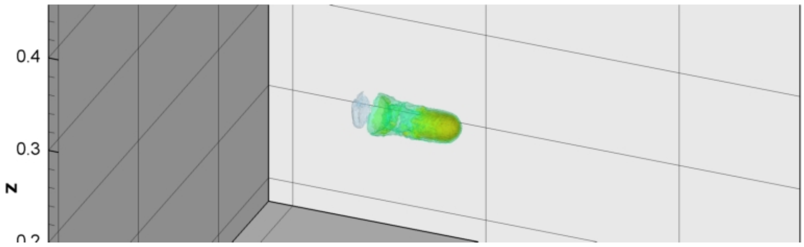

Figure 6.101: Net charge density at times $t_1 = 1.26 \cdot 10^{-8}$ s, 3D view

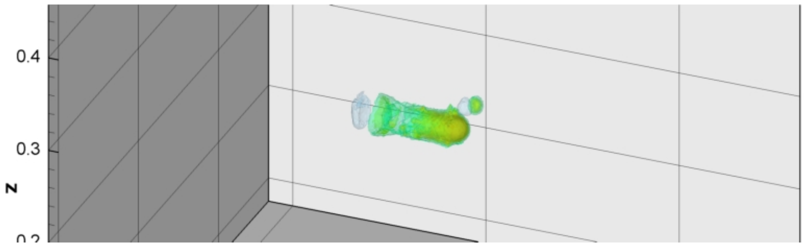

Figure 6.102: Net charge density at times $t_1 = 1.31 \cdot 10^{-8}$ s, 3D view

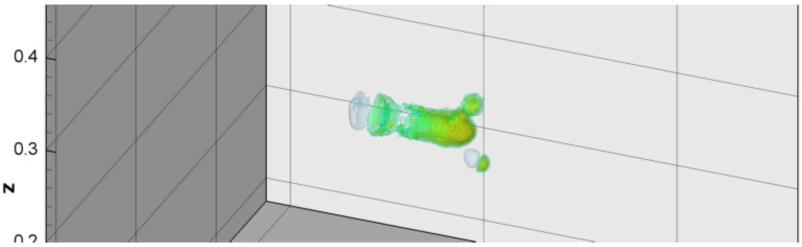

Figure 6.103: Net charge density at times $t_1 = 1.36 \cdot 10^{-8}$ s, 3D view

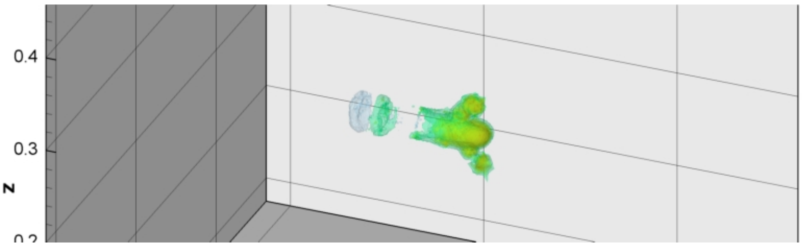

Figure 6.104: Net charge density at times $t_1 = 1.47 \cdot 10^{-8}$ s, 3D view

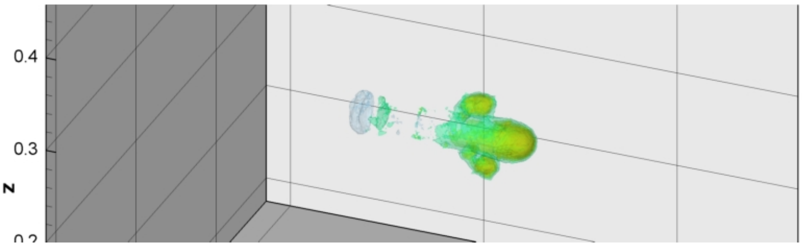

Figure 6.105: Net charge density at times $t_1 = 1.68 \cdot 10^{-8}$ s, 3D view

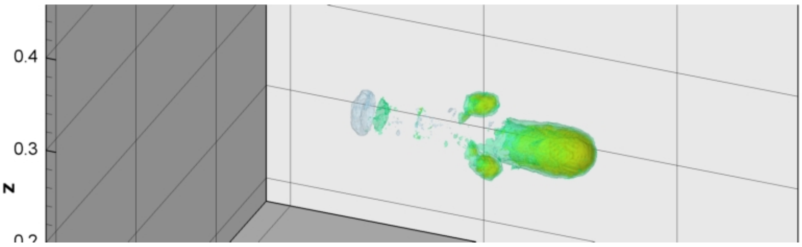

Figure 6.106: Net charge density at times $t_1 = 1.89 \cdot 10^{-8}$ s, 3D view

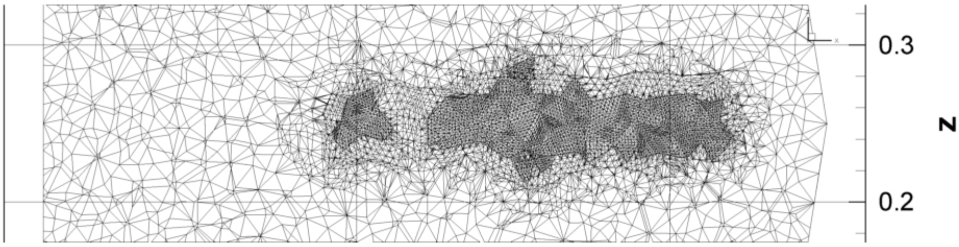

Figure 6.107: Mesh at time $t = 1.89 \cdot 10^{-8}$ s in a cut plane $y = 0.25$

Chapter 7

Conclusions

The goals of the thesis have been fulfilled. The work deals with the streamer propagation which is an electric discharge in a high voltage electric field. The streamer branching has been successfully simulated by introducing of temporal disturbances in the electric field. We have simulated a change of streamer direction, a singular branching and a multiple branching, see the section 6.4. The shape and position of streamer branches depend on position of plasma seeds (local disturbances) and also on the time in which the disturbances influence the electric field. We have referred about the streamer branching in [3] and we haven't seen successful simulation of the streamer branching in full 3D coordinates yet.

Results in chapter 6 demonstrate the applicability of the developed method. The results present a streamer propagation in a simple 3D domain. Computation of a one 3D test takes around three weeks and the computer memory costs are also very high even if the dynamical mesh adaptation is used. That's the reason why only three levels of refinement are taken into account in the calculations although 2D tests show that at least four levels of the mesh refinement should be considered.

The properties of the method have been tested on 2D problems (see chapter 5). The tests of the adaptation criterion in the section 5.3 reveal that the grid density around the streamer (planar discharge) head and the streamer path has influence on the velocity of the propagation. Therefor we need a fine mesh in the computational domain at these places. We also found an optimal adaptation level (as compromise of the grid size and quality of results) and confirm, that results achieved on the adapted grid are the same as results of a static fine grid in the whole domain (see Figs. 5.12, 5.13). The dynamical mesh adaptation decreases the CPU time almost 60 times. We have released findings about the influence of the dynamical mesh adaptation and the proper choice of the refinement criterion in [1], [2] and [4]. We don't deem that the development of the method is finished. The parallelization should be improved and maybe more effective algorithm for solving of a liner systems should be included.

A general platform based on C++ programing have been used and successfully tested. It can solve different physical problems. The streamer motion has been solved by the platform in this thesis. It's been used for a shallow water system at University Paris 13 in thesis [23]. The platform is also utilized at Czech Technical University for fluid mechanics computation where an implicit method has been included.

Future Work

Despite the achieved results, there is still lots of work to do. It's necessary to improve the dynamical mesh adaptation algorithm in three dimension. Especially the mesh conformity step needs changes to produce better quality of a mesh. One of the possibilities is described e.g. in papers [18], [19].
It is also important to implement massive parallelization to the developed code because the 3D computation is very time consuming.
The simple minimal model of streamer motion needs to be extended to more complex model (adding equation for other charged species, more realistic time depending boundary conditions, chemical reactions etc.).

Appendix A

A.1 Relations Used in Reconstruction

Relations coming from the least square method which are used in 2D reconstruction

$$\begin{aligned} I_{xx} &= \sum_{j=1}^{n} (x_j - x_i)^2, & I_{yy} &= \sum_{j=1}^{n} (y_j - y_i)^2, \\ I_{xy} &= \sum_{j=1}^{n} (x_j - x_i)(y_j - y_i), & J_x &= \sum_{j=1}^{n} (n_{ej} - n_{ei})(x_j - x_i), \\ J_y &= \sum_{j=1}^{n} (n_{ej} - n_{ei})(y_j - y_i), & D &= I_{xx} \cdot I_{yy} - I_{xy}^2. \end{aligned} \tag{A.1}$$

Relations coming for the least square method which are used in 3D reconstruction

$$\begin{aligned} I_{xx} &= \sum_{j=1}^{n} (x_j - x_i)^2, & I_{xy} &= \sum_{j=1}^{n} (x_j - x_i)(y_j - y_i), \\ I_{yy} &= \sum_{j=1}^{n} (y_j - y_i)^2, & I_{xz} &= \sum_{j=1}^{n} (x_j - x_i)(z_j - z_i), \\ I_{zz} &= \sum_{j=1}^{n} (z_j - z_i)^2, & I_{yz} &= \sum_{j=1}^{n} (y_j - y_i)(z_j - z_i), \end{aligned} \tag{A.2}$$

$$D = I_{xx}I_{yy}I_{zz} + 2I_{xy}I_{xz}I_{yz} - I_{xx}I_{yz}^2 - I_{yy}I_{xz}^2 - I_{zz}I_{xy}^2, \tag{A.3}$$

$$\begin{aligned} J_x &= \sum_{j=1}^{n} (n_{ej} - n_{ei})(x_j - x_i), & J_y &= \sum_{j=1}^{n} (n_{ej} - n_{ei})(y_j - y_i), \\ J_z &= \sum_{j=1}^{n} (n_{ej} - n_{ei})(z_j - z_i), \end{aligned} \tag{A.4}$$

A.2 Relations for Dissipative Terms

Weights parameters for the 2D gradient computation in diffusion terms:

$$\begin{aligned}
&R_x = \sum_{p=1}^{N(A)} (x_p - x_A)\,, \quad R_y = \sum_{p=1}^{N(A)} (y_p - y_A)\,, \\
&I_{xx} = \sum_{p=1}^{N(A)} (x_p - x_A)^2\,, \quad I_{yy} = \sum_{p=1}^{N(A)} (y_p - y_A)^2\,, \quad I_{xy} = \sum_{p=1}^{N(A)} (x_p - x_A)(y_p - y_A)\,, \\
&\lambda_x = \frac{I_{xy}R_y - I_{yy}R_x}{D}, \quad \lambda_y = \frac{I_{xy}R_x - I_{xx}R_y}{D}, \quad D = I_{xx}I_{yy} - I_{xy}^2.
\end{aligned} \tag{A.5}$$

3D computation of the gradient in the diffusion terms is given by following relation

$$\begin{aligned}
\vec{\nabla} n_{eij} = \frac{1}{\mu\left(D_{\sigma_{ij}}\right)} & \left[\frac{1}{3}\left(n_e(A) + n_e(B) + n_e(R)\right)\vec{\mathfrak{n}}_{ABR}|\sigma_{ABR}| + \right. \\
&+ \frac{1}{3}\left(n_e(B) + n_e(C) + n_e(R)\right)\vec{\mathfrak{n}}_{BCR}|\sigma_{BCR}| + \\
&+ \frac{1}{3}\left(n_e(C) + n_e(D) + n_e(R)\right)\vec{\mathfrak{n}}_{CDR}|\sigma_{CDR}| + \\
&+ \frac{1}{3}\left(n_e(D) + n_e(A) + n_e(R)\right)\vec{\mathfrak{n}}_{DAR}|\sigma_{DAR}| + \\
&+ \frac{1}{3}\left(n_e(A) + n_e(B) + n_e(L)\right)\vec{\mathfrak{n}}_{BAL}|\sigma_{BAL}| + \\
&+ \frac{1}{3}\left(n_e(B) + n_e(C) + n_e(L)\right)\vec{\mathfrak{n}}_{CBL}|\sigma_{CBL}| + \\
&+ \frac{1}{3}\left(n_e(C) + n_e(D) + n_e(L)\right)\vec{\mathfrak{n}}_{DCL}|\sigma_{DCL}| + \\
&\left. + \frac{1}{3}\left(n_e(D) + n_e(A) + n_e(L)\right)\vec{\mathfrak{n}}_{ADL}|\sigma_{ADL}|\right],
\end{aligned} \tag{A.6}$$

where $n_e(A)$ ($n_e(B),\ \cdots$) is the electron density value in the point A ($B,\ \cdots$), $\vec{\mathfrak{n}}_{ABR}$ ($\vec{\mathfrak{n}}_{BCR},\ \cdots$) is an unit outward vector of the face σ_{ABR} ($\sigma_{BCR},\ \cdots$) and $|\sigma_{ABR}|$ ($|\sigma_{BCR}|,\ \cdots$) is its measure. When we factor one third out of the square bracket and modify expressions inside the bracket, we obtain

$$
\begin{aligned}
\vec{\nabla} n_{eij} = \frac{1}{3\mu\left(D_{\sigma_{ij}}\right)} \Big[& \left(\vec{\mathfrak{n}}_{ABR}|\sigma_{ABR}| + \vec{\mathfrak{n}}_{DAR}|\sigma_{DAR}| + \vec{\mathfrak{n}}_{BAL}|\sigma_{BAL}| + \vec{\mathfrak{n}}_{ADL}|\sigma_{ADL}|\right) n_e(A) + \\
& + \left(\vec{\mathfrak{n}}_{ABR}|\sigma_{ABR}| + \vec{\mathfrak{n}}_{BCR}|\sigma_{BCR}| + \vec{\mathfrak{n}}_{BAL}|\sigma_{BAL}| + \vec{\mathfrak{n}}_{CBL}|\sigma_{CBL}|\right) n_e(B) + \\
& + \left(\vec{\mathfrak{n}}_{BCR}|\sigma_{BCR}| + \vec{\mathfrak{n}}_{CDR}|\sigma_{CDR}| + \vec{\mathfrak{n}}_{CBL}|\sigma_{CBL}| + \vec{\mathfrak{n}}_{DCL}|\sigma_{DCL}|\right) n_e(C) + \\
& + \left(\vec{\mathfrak{n}}_{CDR}|\sigma_{CDR}| + \vec{\mathfrak{n}}_{DAR}|\sigma_{DAR}| + \vec{\mathfrak{n}}_{DCL}|\sigma_{DCL}| + \vec{\mathfrak{n}}_{ADL}|\sigma_{ADL}|\right) n_e(D) + \\
& + \left(\vec{\mathfrak{n}}_{ABR}|\sigma_{ABR}| + \vec{\mathfrak{n}}_{BCR}|\sigma_{BCR}| + \vec{\mathfrak{n}}_{CDR}|\sigma_{CDR}| + \vec{\mathfrak{n}}_{DAR}|\sigma_{DAR}|\right) n_e(R) + \\
& + \left(\vec{\mathfrak{n}}_{BAL}|\sigma_{BAL}| + \vec{\mathfrak{n}}_{CBL}|\sigma_{CBL}| + \vec{\mathfrak{n}}_{DCL}|\sigma_{DCL}| + \vec{\mathfrak{n}}_{ADL}|\sigma_{ADL}|\right) n_e(L) \Big].
\end{aligned}
\tag{A.7}
$$

The vector equilibrium implies

$$
\begin{aligned}
\vec{\mathfrak{n}}_{ABR}|\sigma_{ABR}| + \vec{\mathfrak{n}}_{DAR}|\sigma_{DAR}| + \vec{\mathfrak{n}}_{BAL}|\sigma_{BAL}| + \vec{\mathfrak{n}}_{ADL}|\sigma_{ADL}| &= \vec{\mathfrak{n}}_{BRDL}|\sigma_{BRDL}|, \\
\vec{\mathfrak{n}}_{ABR}|\sigma_{ABR}| + \vec{\mathfrak{n}}_{BCR}|\sigma_{BCR}| + \vec{\mathfrak{n}}_{BAL}|\sigma_{BAL}| + \vec{\mathfrak{n}}_{CBL}|\sigma_{CBL}| &= \vec{\mathfrak{n}}_{ALCR}|\sigma_{ALCR}|, \\
\vec{\mathfrak{n}}_{BCR}|\sigma_{BCR}| + \vec{\mathfrak{n}}_{CDR}|\sigma_{CDR}| + \vec{\mathfrak{n}}_{CBL}|\sigma_{CBL}| + \vec{\mathfrak{n}}_{DCL}|\sigma_{DCL}| &= -\vec{\mathfrak{n}}_{RBDL}|\sigma_{RBDL}|, \\
\vec{\mathfrak{n}}_{CDR}|\sigma_{CDR}| + \vec{\mathfrak{n}}_{DAR}|\sigma_{DAR}| + \vec{\mathfrak{n}}_{DCL}|\sigma_{DCL}| + \vec{\mathfrak{n}}_{ADL}|\sigma_{ADL}| &= -\vec{\mathfrak{n}}_{ALCR}|\sigma_{ALCR}|, \\
\vec{\mathfrak{n}}_{ABR}|\sigma_{ABR}| + \vec{\mathfrak{n}}_{BCR}|\sigma_{BCR}| + \vec{\mathfrak{n}}_{CDR}|\sigma_{CDR}| + \vec{\mathfrak{n}}_{DAR}|\sigma_{DAR}| &= \vec{\mathfrak{n}}_{ij}|\sigma_{ij}|, \\
\vec{\mathfrak{n}}_{BAL}|\sigma_{BAL}| + \vec{\mathfrak{n}}_{CBL}|\sigma_{CBL}| + \vec{\mathfrak{n}}_{DCL}|\sigma_{DCL}| + \vec{\mathfrak{n}}_{ADL}|\sigma_{ADL}| &= -\vec{\mathfrak{n}}_{ij}|\sigma_{ij}|.
\end{aligned}
\tag{A.8}
$$

When we substitute the equations (A.8) in the equation (A.7), we can write for the electron density gradient $\vec{\nabla} n_{eij}$

$$
\begin{aligned}
\vec{\nabla} n_{eij} = \frac{1}{3\mu\left(D_{\sigma_{ij}}\right)} \Big[& \left(n_e(A) - n_e(C)\right) \vec{\mathfrak{n}}_{BRDL}|\sigma_{BRDL}| + \\
& + \left(n_e(B) - n_e(D)\right) \vec{\mathfrak{n}}_{ALCR}|\sigma_{ALCR}| + \left(n_e(R) - n_e(L)\right) \vec{\mathfrak{n}}_{ij}|\sigma_{ij}| \Big].
\end{aligned}
\tag{A.9}
$$

Parameters in the weights for gradient computation are given by following relations

$$
\begin{aligned}
I_{xx} &= \sum_{p=1}^{N(A)} \left(x_p - x_A\right)^2, & I_{xy} &= \sum_{p=1}^{N(A)} \left(x_p - x_A\right)\left(y_p - y_A\right), & R_x &= \sum_{p=1}^{N(A)} \left(x_p - x_A\right), \\
I_{yy} &= \sum_{p=1}^{N(A)} \left(y_p - y_A\right)^2, & I_{xz} &= \sum_{p=1}^{N(A)} \left(x_p - x_A\right)\left(z_p - z_A\right), & R_y &= \sum_{p=1}^{N(A)} \left(y_p - y_A\right), \\
I_{zz} &= \sum_{p=1}^{n(A)} \left(z_p - z_A\right)^2, & I_{yz} &= \sum_{p=1}^{N(A)} \left(y_p - y_A\right)\left(z_j - z_v\right), & R_z &= \sum_{p=1}^{N(A)} \left(z_p - z_A\right),
\end{aligned}
\tag{A.10}
$$

$$
D = I_{xx}I_{yy}I_{zz} + 2I_{xy}I_{xz}I_{yz} - I_{xx}I_{yz}^2 - I_{yy}I_{xz}^2 - I_{zz}I_{xy}^2,
\tag{A.11}
$$

$$
\begin{aligned}
\lambda_x &= \frac{\left(I_{yz}^2 - I_{yy}I_{zz}\right)R_x + \left(I_{xy}I_{zz} - I_{xz}I_{yz}\right)R_y + \left(I_{xz}I_{yy} - I_{xy}I_{yz}\right)R_z}{D},\\
\lambda_y &= \frac{\left(I_{xy}I_{zz} - I_{xz}I_{yz}\right)R_x + \left(I_{xz}^2 - I_{xx}I_{zz}\right)R_y + \left(I_{yz}I_{xx} - I_{xz}I_{xy}\right)R_z}{D},\\
\lambda_z &= \frac{\left(I_{xz}I_{yy} - I_{xy}I_{yz}\right)R_x + \left(I_{yz}I_{xx} - I_{xz}I_{xy}\right)R_y + \left(I_{xy}^2 - I_{xx}I_{yy}\right)R_z}{D}.
\end{aligned}
\tag{A.12}
$$

Appendix B

B.1 Material Functions for the Air

The relations for the material functions are taken from [5].

Electron drift velocity $\vec{v}_e$:

$$\begin{aligned}
&\text{for } \frac{\|\vec{E}\|}{N} > 2\cdot 10^{-15}, && \vec{v}_e = -\left[7.4\cdot 10^{21}\cdot\frac{\|\vec{E}\|}{N} + 7.1\cdot 10^{6}\right]\cdot\frac{\vec{E}}{\|\vec{E}\|} \quad [\text{cm}\cdot\ \text{s}^{-1}]\\
&\text{for } 10^{-16} < \frac{\|\vec{E}\|}{N} \le 2\cdot 10^{-15}, && \vec{v}_e = -\left[1.03\cdot 10^{22}\cdot\frac{\|\vec{E}\|}{N} + 1.3\cdot 10^{6}\right]\cdot\frac{\vec{E}}{\|\vec{E}\|} \quad [\text{cm}\cdot\ \text{s}^{-1}]\\
&\text{for } 2.6\cdot 10^{-17} < \frac{\|\vec{E}\|}{N} \le 10^{-16}, && \vec{v}_e = -\left[7.2973\cdot 10^{21}\cdot\frac{\|\vec{E}\|}{N} + 1.63\cdot 10^{6}\right]\cdot\frac{\vec{E}}{\|\vec{E}\|} \quad [\text{cm}\cdot\ \text{s}^{-1}]\\
&\text{for } \frac{\|\vec{E}\|}{N} \le 2.6\cdot 10^{-17}, && \vec{v}_e = -\left[6.87\cdot 10^{22}\cdot\frac{\|\vec{E}\|}{N} + 3.38\cdot 10^{4}\right]\cdot\frac{\vec{E}}{\|\vec{E}\|} \quad [\text{cm}\cdot\ \text{s}^{-1}]
\end{aligned} \tag{B.1}$$

Positive ion drift velocity $\vec{v}_p$:

$$\vec{v}_p = 2.34\cdot\vec{E}\cdot\frac{p_0}{p} \quad [\text{cm}\cdot\ \text{s}^{-1}] \tag{B.2}$$

Negative ion drift velocity $\vec{v}_n$:

$$\begin{aligned}
&\text{for } \frac{\|\vec{E}\|}{N} > 5\cdot 10^{-16}, && \vec{v}_n = -2.7\cdot\vec{E}\cdot\frac{p_0}{p} \quad [\text{cm}\cdot\ \text{s}^{-1}]\\
&\text{for } \frac{\|\vec{E}\|}{N} < 5\cdot 10^{-15}, && \vec{v}_n = -1.86\cdot\vec{E}\cdot\frac{p_0}{p} \quad [\text{cm}\cdot\ \text{s}^{-1}]
\end{aligned} \tag{B.3}$$

$\frac{p_0}{p}$ is a ratio of the atmospheric pressure to a gas pressure.

Diffusion coefficient D_e:

$$D_e = \left[0.3341\cdot 10^{9}\cdot\left(\frac{\|\vec{E}\|}{N}\right)^{0.54069}\right]\cdot\frac{\|\vec{v}_e\|}{\|\vec{E}\|} \quad [\text{cm}^2\cdot\ \text{s}^{-1}] \tag{B.4}$$

Ionization coefficient α:

$$\begin{aligned} &\text{if } \frac{\|\vec{E}\|}{N} > 1.5\cdot 10^{-15}, \quad \frac{\alpha}{N} = 2\cdot 10^{-16}\cdot \exp\left(\frac{-7.248\cdot 10^{-15}}{\|\vec{E}\|/N}\right) \quad [\mathrm{cm}^2] \\ &\text{else,} \qquad \frac{\alpha}{N} = 6.619\cdot 10^{-17}\cdot \exp\left(\frac{-5.593\cdot 10^{-15}}{\|\vec{E}\|/N}\right) \quad [\mathrm{cm}^2] \end{aligned} \tag{B.5}$$

Attachment coefficient β_{att}, (labeled η in [5]):

$$\begin{aligned} &\text{if } \frac{\|\vec{E}\|}{N} > 1.05\cdot 10^{-15}, \quad \frac{\eta_2}{N} = 8.889\cdot 10^{-5}\cdot \frac{\|\vec{E}\|}{N} + 2.567\cdot 10^{-19} \quad [\mathrm{cm}^2] \\ &\text{else,} \qquad \frac{\eta_2}{N} = 6.089\cdot 10^{-4}\cdot \frac{\|\vec{E}\|}{N} - 2.893\cdot 10^{-19} \quad [\mathrm{cm}^2] \end{aligned} \tag{B.6}$$

$$\frac{\eta_3}{N^2} = 4.7778\cdot 10^{-59}\cdot \left(\frac{\|\vec{E}\|}{N}\right)^{-1.2749} \quad [\mathrm{cm}^5] \tag{B.7}$$

η_2 is two-body attachment coefficient, η_3 is three-body attachment coefficient.

Recombination coefficients $\beta_{ep},\ \beta_{np}$ (there is only β in [5]):

$$\beta = 2\cdot 10^{-7} \quad [\mathrm{cm}^3\cdot\ \mathrm{s}^{-1}] \tag{B.8}$$

The neutral gas density $N = 2.5\cdot 10^{19} \quad [\mathrm{cm}^{-3}]$ is use in relations (B.1), (B.3) - (B.7).

Bibliography

[1] F. Benkhaldoun, J. Fořt, K. Hassouni, J. Karel: Dynamical grid adaptation method for model plasma problem, *Topical Problems of Fluid Mechanics* , 2012, pp. 1-4

[2] F. Benkhaldoun , J. Fort, K. Hassouni, J. Karel: Simulation of Planar ionization wave Front Propagation on Unstructured Adaptive Grid, *Journal of Computational and Applied Mathematics*, vol. 236, Issue 18, 2012, pp. 46234634

[3] F. Benkhaldoun, J. Fořt, K. Hassouni, J. Karel, G. Scarella, D.Trdlička: A Full 3D dynamically adaptive unstructured grid mesh finite volume approach to simulate multiple branching in streamer propagation, *IEEE Transaction on Plasma Science*, accepted for publishing

[4] F. Benkhaldoun, J. Fořt, K. Hassouni, J. Karel, D.Trdlička: Numerical Study of Planar Discharge Motion, *Applied and Computational Mechanics*, accepted for publishing

[5] R. Morrow, J. J. Lowke: Streamer propagation in air, *J, Phys. O: Appl. Phys.*, 30 (1997) 614-627.

[6] P. Crispel, P. Degond, M. H. Vignal: Quasi-neutral fluid models for current-carrying plasmas, *Journal of Computational Physics*, **205** (2005) 408438.

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

[8] D.S. Nikandrov, R.R. Arslanbekov, V.I. Kolobov: Streamer simulations with dynamically adaptive cartesian mesh, *IEEE Trans. Plasma Sci.* 36 (4) (2008) 932-933.

[9] S. Pancheshnyi, P. Segur, J. Capeillere, A. Bourdon: Numerical simulation of filamentary discharges with parallel adaptive mesh refinement, *J. Comput. Phys.* 227 (13) (2008) 6574-6590.

[10] Min, W., H. Kim, S. Lee, and S. Hahn (2001): A study on the streamer simulation using adaptive mesh generation and FEM-FCT, *IEEE Transactions on Magnetics*, 37 (5, Part 1), 3141-3144

[11] Papadakis, A. P., G. E. Georghiou, and A. C. Metaxas (2008): New high quality adaptive mesh generator utilized in modelling plasma streamer propagation at atmospheric pressures, *Journal of Physics D: Applied Physics*, 41(23), 234,019.

[12] O. Ducasse, L. Papageorghiou, O. Eichwald, N. Spyrou, M. Yousfi: Critical analysis on two-dimenional point-to-plane streamer simulation using finite element and finite volume methods. *IEEE Transactions of Plasma Science* 35 No.5, 2007, pp. 1287 -1300

[13] A. Bourdon, D. Bessieres, J. Paillol, A. Michau, K. Hassouni, E. Marode, P. Segur: Influence of numerical schemes on positive streamer propagation. 27 ICPIG, July 18-22, 2005, Eindhoven, The Nederlands.

[14] J. Dobes, J. Fort, J. Halama:Numerical solution of single and multi-phase internal transonic flow problems *Int. Journal for Numerical Methods in Fluid*, Vol. 48, Iss. 1, pp. 91-97, 2005

[15] S. Célestin: Etude de la dynamique des streamers dans l'air a la pression atmospherique. Thesis, E.M2.C CNRS et ecole Centrale Paris, 2008

[16] L. Papageorgiou, A. C. Metaxas, G. E. Georghiou: Three-dimensional numerical modelling of gas discharges at atmospheric pressure incorporating photoionization phenomena *Journal of Physics D: Applied Physics* 44, (2011) 045203

[17] J. Blazek: *Computational Fluid Dynamics: Principles and Applications*, Elsevier 2001, ISBN 0 08 043009 0

[18] J. Bey: Tetrahedral Grid Refinement. *Computing*, 55, 355-378 (1995)

[19] A. Liu, B. Joe: Quality Local Refinement of Tetrahedral Meshes Based on 8-Subtetrahedron Subdivision. *Mathematics of Computation*, Vol. 65, no 215, 1996, pp. 1183-1200

[20] B. Martin: *Élaboration de solveurs volumes finis 2D/3D pour résoudre le problème de l'élasticité linéaire*, thesis, ENS Cachan, 2012

[21] S. Wang, G. Yuan, Y. Li, Z. Sheng: Discrete Maximum Principle based on Repair Technique for Diamond Type Scheme of Diffusion Problems. *International Journal for Numerical Methods in Fluids*, Vol. 70, 2012, pp. 1188-1205

[22] D. Trdlička: *Application of Finite Volume Method for Numerical Simulation of Streamer*, thesis, CTU in Prague, 2012

[23] S. Sari: *Modélisation mathématique et numérique de systèmes multicouches avec transport de sédimets pour les problèmes à surface libre*, thesis, Université Paris 13, 2013