Why we need a fast and accurate solution of Poisson’s equation for low-temperature plasmas?

François Pechereau\textsuperscript{1} and Anne Bourdon\textsuperscript{2}

\textsuperscript{1}Laboratory CERFACS, Toulouse  
\textsuperscript{2}Laboratory LPP, Ecole Polytechnique  
on leave from laboratory EM2C, Ecole Centrale Paris

Journée problème de Poisson, January 26, 2015, IHP, France
1. Introduction on non-thermal discharges at atmospheric pressure
2. Rapid overview of the characteristics of streamer discharges
3. On the modeling of streamer discharges
4. Exemple of code improvements: test case
5. Conclusions
1. Introduction on non-thermal discharges at atmospheric pressure
2. Rapid overview of the characteristics of streamer discharges
3. On the modeling of streamer discharges
4. Exemple of code improvements: test case
5. Conclusions
Non-thermal discharges at atmospheric pressure

Applications of non-thermal discharges at \( P_{atm} \)?

- Since a few years, many studies on non-thermal discharges at atmospheric ground pressure
- Wide range of applications at low pressure → possible at ground pressure to reduce costs (no need for pumping systems)?
- New applications as biomedical applications, plasma assisted combustion

Plasma assisted combustion
\[ \Phi = 0.8, \text{ Air flow rate } = 15 \text{ m}^3/\text{h} \]
Lean premixed burner
How to generate non-thermal discharges at atmospheric pressure?

Between two metallic electrodes

- Interelectrode gaps of a few mm to a few cm at $P_{atm}$

- Risk: If the voltage pulse is too long → transition to spark

4 cm

Briels, PhD (2007)
How to generate non-thermal discharges at atmospheric pressure?

Dielectric Barrier Discharge (DBD)

- (A) Ignition of a discharge between electrodes
- (B) Transition to spark $\rightarrow$ high current, $T_g > 300K$
- (C) To prevent spark transition: dielectric layers between the electrodes

- $T_i = T_g = 300K$, $T_e > 10000K$ $\rightarrow$ Cold plasma
- O, OH, radicals and UV radiation
How to generate non-thermal discharges at atmospheric pressure?

Dielectric Barrier Discharge (DBD)

- (A) Ignition of a discharge between electrodes
- (B) **Transition to spark** → high current, $T_g > 300K$
- (C) To prevent spark transition: dielectric layers between the electrodes

- $T_i = T_g = 300K$, $T_e > 10000K$ → **Cold plasma**
- O, OH, radicals and UV radiation
How to generate non-thermal discharges at atmospheric pressure?

Dielectric Barrier Discharge (DBD)

- (A) Ignition of a discharge between electrodes
- (B) **Transition to spark** → high current, $T_g > 300K$

- (C) To prevent spark transition: **dielectric layers between the electrodes**

- $T_i = T_g = 300K$, $T_e > 10000K$ → **Cold plasma**
- O, OH, radicals and UV radiation
How to generate non-thermal discharges at atmospheric pressure?

Dielectric Barrier Discharge (DBD)

- (A) Ignition of a discharge between electrodes
- (B) **Transition to spark** → high current, $T_g > 300K$
- (C) To prevent spark transition: **dielectric layers between the electrodes**

- $T_i = T_g = 300K$, $T_e > 10000K$ → **Cold plasma**
- O, OH, radicals and UV radiation
How to generate non-thermal discharges at atmospheric pressure?

**Dielectric Barrier Discharge (DBD)**

- Interelectrode gaps of a few mm to a few cm at $P_{atm}$

Plane-plane reactor (LPGP Orsay)

Wire-cylinder (GREMI Orléans)

### Structure of P\textsubscript{atm} discharges

- At P\textsubscript{atm}, non-thermal atmospheric pressure discharges may have filamentary or diffuse structures.

#### Filamentary discharges

- High electron density ($10^{14}$ cm\(^{-3}\)) in a filament with a radius of the order of 100 µm → high density of active species (radicals, excited species). However, local heating may be significant.

#### Diffusive discharges

- Low density of electrons, large volume of the discharge and negligible heating.
1. Introduction on non-thermal discharges at atmospheric pressure
2. Rapid overview of the characteristics of streamer discharges
3. On the modeling of streamer discharges
4. Example of code improvements: test case
5. Conclusions
1 Introduction on non-thermal discharges at atmospheric pressure

2 Rapid overview of the characteristics of streamer discharges

3 On the modeling of streamer discharges

4 Exemple of code improvements: test case

5 Conclusions
Streamer propagation in air at atmospheric pressure

In air at $P_{atm}$, the breakdown field is 30 kV/cm.

In a point to plane geometry, the electric field is enhanced close to the point electrode.

At first, the discharge will start from the point electrode and will propagate towards the grounded plane.
Streamer propagation in air at atmospheric pressure

- Typical radius of the filament = 100 µm, velocity = $10^8$ cm/s so 10 ns for 1 cm
- Almost neutral channel and charged streamer head
- In the conductive channel: low electric field (5 kV/cm) and a charged species density of $10^{13}-10^{14}$ cm$^{-3}$
- In the streamer head peak: peak electric field (140 kV/cm)
- Ions are almost immobile during propagation: streamer velocity > drift velocity of electrons
- A streamer discharge is an ionization wave
Streamer propagation in air at atmospheric pressure

- The discharge moves towards the cathode, whereas electrons move towards the anode.
- Need for seed electrons for the discharge propagation.
Streamer propagation in air at atmospheric pressure

Origin of seed electrons

- Cosmic rays (up to $10^4 \text{ cm}^{-3}$), preionization from previous discharges
- Photoionization (depends on the gas mixture) in air

Ionizing radiation is in the region $980 < \lambda < 1025 \text{ Å}$.
Streamer propagation in air at atmospheric pressure

- Seed electrons in front of the streamer
- Transport of charged species
- Screening of the streamer head by electrons
- Streamer moves forward
Streamer propagation in air at atmospheric pressure

- Seed electrons in front of the streamer
- Transport of charged species
- Screening of the streamer head by electrons
- Streamer moves forward
Streamer propagation in air at atmospheric pressure

- Seed electrons in front of the streamer
- Transport of charged species
- Screening of the streamer head by electrons
- Streamer moves forward
How to simulate non-thermal discharges at $P_{atm}$?

**Complex medium**
- Charged species (ions, electrons), atoms and molecules (excited or not) and photons
- Simplest models take into account only charged species (and photons)
- Magnetic effects are negligible: electric field derived from Poisson’s equation

**Different models**
- Microscopic model for charged particles coupled with Poisson’s equation (PIC-MCC model (Chanrion and Neubert JCP (2008) and JGR (2010))
- Most popular: macroscopic fluid model coupled to Poisson’s equation
- Hybrid models:
  - Particle model in the high field region
  - Fluid model in the streamer channel (low field, high electron densities)
  - Transition between both models:
    - In energy: *bulk-model* (Bonaventura et al., ERL (2014))
### Complex medium

- Charged species (ions, electrons), atoms and molecules (excited or not) and photons
- Simplest models take into account only charged species (and photons)
- Magnetic effects are negligible: electric field derived from Poisson’s equation

### Different models

- Microscopic model for charged particles coupled with Poisson’s equation (PIC-MCC model (Chanrion and Neubert JCP (2008) and JGR (2010))

- **Most popular**: macroscopic fluid model coupled to Poisson’s equation

- Hybrid models:
  - Particle model in the high field region
  - Fluid model in the streamer channel (low field, high electron densities)
  - Transition between both models:
    - In energy: *bulk-model* (Bonaventura et al., ERL (2014))
Table of contents

1. Introduction on non-thermal discharges at atmospheric pressure
2. Rapid overview of the characteristics of streamer discharges
3. On the modeling of streamer discharges
4. Exemple of code improvements: test case
5. Conclusions
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction on non-thermal discharges at atmospheric pressure</td>
</tr>
<tr>
<td>2</td>
<td>Rapid overview of the characteristics of streamer discharges</td>
</tr>
<tr>
<td>3</td>
<td>On the modeling of streamer discharges</td>
</tr>
<tr>
<td>4</td>
<td>Example of code improvements: test case</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions</td>
</tr>
</tbody>
</table>
Continuity equation is solved for electrons, positive and negative ions

\[ \frac{\partial n_i}{\partial t} + \text{div} \, j_i = S_i \]  

(1)

Drift-diffusion approximation

\[ j_i = \mu_i \, n_i \, E - D_i \, \text{grad} \, n_i \]  

(2)

Poisson’s equation:

\[ \varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e) \]  

(3)
Continuity equation is solved for electrons, positive and negative ions

\[
\frac{\partial n_i}{\partial t} + \text{div} \; j_i = S_i
\]  \hspace{1cm} (1)

Drift-diffusion approximation

\[
j_i = \mu_i n_i E - D_i \text{grad} \; n_i
\]  \hspace{1cm} (2)

Poisson’s equation:

\[
\varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e)
\]  \hspace{1cm} (3)

Strong non-linear coupling between drift-diffusion and Poisson’s equations
The species densities have to be calculated accurately as their difference is used to compute the potential and then the electric field
Continuity equation is solved for electrons, positive and negative ions

\[ \frac{\partial n_i}{\partial t} + \text{div } j_i = S_i \]  \hspace{1cm} (4)

Drift-diffusion approximation

\[ j_i = \mu_i \, n_i \, E - D_i \, \text{grad } n_i \]  \hspace{1cm} (5)

Source terms for air:

\[
\begin{aligned}
S_e &= (\partial_t n_e)_{\text{chem}} = (\nu_\alpha - \nu_\eta - \beta_{ep} n_p) \, n_e + \nu_{\text{det}} n_n + S_{ph}, \\
S_n &= (\partial_t n_n)_{\text{chem}} = - (\nu_{\text{det}} + \beta_{np} n_p) \, n_n + \nu_\eta n_e , \\
S_p &= (\partial_t n_p)_{\text{chem}} = - (\beta_{ep} n_e + \beta_{np} n_n) \, n_p + \nu_\alpha n_e + S_{ph} .
\end{aligned}
\]  \hspace{1cm} (6)

Local field approximation:  \( \nu_\alpha (|\vec{E}|/N), \quad \nu_\eta (|\vec{E}|/N), \quad \mu_i (|\vec{E}|/N), \quad D_i (|\vec{E}|/N) \)


Transport parameters and source terms are pre-calculated (Bolsig+ solver - http://www.bolsig.laplace.univ-tlse.fr/)
Continuity equation is solved for electrons, positive and negative ions

\[
\frac{\partial n_i}{\partial t} + \text{div} \, j_i = S_i 
\]  

(4)

Drift-diffusion approximation

\[
j_i = \mu_i \, n_i \, E - D_i \, \text{grad} \, n_i
\]

(5)

Source terms for air:

\[
\begin{align*}
S_e &= (\partial_t n_e)_{\text{chem}} = (\nu_\alpha - \nu_\eta - \beta_{ep} n_p) \, n_e + \nu_{\text{det}} n_n + S_{ph}, \\
S_n &= (\partial_t n_n)_{\text{chem}} = -(\nu_{\text{det}} + \beta_{np} n_p) \, n_n + \nu_\eta n_e, \\
S_p &= (\partial_t n_p)_{\text{chem}} = -(\beta_{ep} n_e + \beta_{np} n_n) \, n_p + \nu_\alpha n_e + S_{ph}.
\end{align*}
\]

(6)

Local field approximation: \(\nu_\alpha(|E|/N), \nu_\eta(|E|/N), \mu_i(|E|/N), D_i(|E|/N)\)


Transport parameters and source terms are pre-calculated (Bolsig+ solver - http://www.bolsig.laplace.univ-tlse.fr/)
Photoionization in air

Photoionization model in air

- Non-local phenomenon
- Photoionization rate at one position depends on all the emitters positions
- Original model requires to calculate a 3D integral for each point at each time step
- New model based on a third order approximation of the radiative transfer equation → differential model [Bourdon et al. *PSST*, 16, 656 (2007), Liu et al. *APL* 91, 211501 (2007)]
Poisson’s equation

\[ \varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e) \]  

Photoionization source term $S_{ph}$: SP3 model

It leads to solve 18+1 Poisson’s equation!
Poisson’s equation

\[ \varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e) \]  \hspace{1cm} (7)

Photoionization source term \( S_{ph} \): SP3 model


\[ \begin{aligned}
\nabla^2 \phi_{1,j}(\vec{r}) - A_{1,j} \phi_{1,j} &= S_{1,j} \\
\nabla^2 \phi_{2,j}(\vec{r}) - A_{2,j} \phi_{2,j} &= S_{2,j};
\end{aligned} \]  \hspace{1cm} (8)

\( \lambda_{j=1,3} \rightarrow \) 2 Poisson’s equation (\( \phi_{1,j} \) and \( \phi_{2,j} \) \( \times \) 3 iterations for BC

\[ \downarrow \]

6 \( \times \) 3 Poisson’s equations \( \rightarrow \) \( S_{ph} = \sum_j f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r})) \)

It leads to solve 18+1 Poisson’s equation!
Poisson’s equation

\[ \varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e) \]  (7)

Photoionization source term \( S_{ph} \): SP3 model

\[ \begin{cases} \nabla^2 \phi_{1,j}(\vec{r}) - A_{1,j}\phi_{1,j} = S_{1,j} \\ \nabla^2 \phi_{2,j}(\vec{r}) - A_{2,j}\phi_{2,j} = S_{2,j}; \end{cases} \]  (8)

\( \lambda_{j=1,3} \rightarrow 2 \) Poisson’s equation \((\phi_{1,j} \text{ and } \phi_{2,j}) \times 3 \) iterations for BC

\[
\downarrow \\
6 \times 3 \text{ Poisson’s equations} \rightarrow S_{ph} = \sum_j f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r}))
\]

It leads to solve 18+1 Poisson’s equation!
Poisson’s equation

\[ \varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e) \] (7)

Photoionization source term $S_{ph}$: SP3 model

\[
\begin{align*}
\nabla^2 \phi_{1,j}(\vec{r}) - A_{1,j} \phi_{1,j}(\vec{r}) &= S_{1,j} \\
\nabla^2 \phi_{2,j}(\vec{r}) - A_{2,j} \phi_{2,j}(\vec{r}) &= S_{2,j};
\end{align*}
\] (8)

\[ \lambda_{j=1,3} \rightarrow 2 \text{ Poisson’s equation (} \phi_{1,j}(\vec{r}) \text{ and } \phi_{2,j}(\vec{r}) \text{) } \times 3 \text{ iterations for BC} \]

\[
\downarrow
\]

\[ 6 \times 3 \text{ Poisson’s equations} \rightarrow S_{ph} = \sum_j = f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r})) \]

It leads to solve $18+1$ Poisson’s equation!
2D Fluid model for discharge in air at \( P_{\text{atm}} \)

- Poisson’s equation

\[
\varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e)
\]  

(7)

- Photoionization source term \( S_{ph} \): SP3 model

\[
\begin{aligned}
\nabla^2 \phi_{1,j}(\vec{r}) - A_{1,j} \phi_{1,j}(\vec{r}) &= S_{1,j} \\
\nabla^2 \phi_{2,j}(\vec{r}) - A_{2,j} \phi_{2,j}(\vec{r}) &= S_{2,j};
\end{aligned}
\]  

(8)

\[\lambda_{j=1,3} \rightarrow 2 \text{ Poisson’s equation (} \phi_{1,j}(\vec{r}) \text{ and } \phi_{2,j}(\vec{r}) \text{)} \times 3 \text{ iterations for BC}\]

\[\downarrow\]

\[6 \times 3 \text{ Poisson’s equations} \rightarrow S_{ph} = \sum_j f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r}))\]

- It leads to solve \(18+1\) Poisson’s equation!
Poisson’s equation

\[ \varepsilon_0 \nabla \cdot (\varepsilon_r \nabla V) = -q_e (n_p - n_n - n_e) \]  

(7)

Photoionization source term \( S_{ph} \): SP3 model
Bourdon et al., Plasma Sources Sci. Technol. 16, (2007)

\[
\begin{align*}
\nabla^2 \phi_{1,j}(\vec{r}) - A_{1,j} \phi_{1,j}(\vec{r}) &= S_{1,j} \\
\nabla^2 \phi_{2,j}(\vec{r}) - A_{2,j} \phi_{2,j}(\vec{r}) &= S_{2,j}; \\
\end{align*}
\]

(8)

\( \lambda_{j=1,3} \rightarrow 2 \) Poisson’s equation \((\phi_{1,j}(\vec{r}) \text{ and } \phi_{2,j}(\vec{r})) \times 3 \) iterations for BC

\[ S_{ph} = \sum_j f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r})) \]

It leads to solve 18+1 Poisson’s equation!
In cylindrical coordinates, Poisson’s equation can be written as

$$
- \frac{\partial}{\partial x} \left( \epsilon \frac{\partial V}{\partial x} \right) - \frac{1}{r} \frac{\partial}{\partial r} \left( \epsilon r \frac{\partial V}{\partial r} \right) = \rho(x, r),
$$

(9)

After integration it leads to

$$
V^E_{i,j} V_{i+1,j} + V^W_{i,j} V_{i-1,j} + V^S_{i,j} V_{i,j-1} + V^N_{i,j} V_{i,j+1} + V^C_{i,j} V_{i,j} = \rho_{i,j} \Omega_{i,j},
$$

(10)
We need to solve Poisson’s equation at each timestep

### Linear solver coupled with Poisson’s equation
- Algorithm based on fast fourier transform
- Iterative methods: NAG, PETSc and HYPRE library
- Direct methods: superLU, MUMPS and PaStiX
- Need for a fast and parallel library either MPI or MPI-OPENMP
- We started with sequential MUMPS solver
- We tested other parallel solvers depending on the case
We need to solve Poisson’s equation at each timestep

Linear solver coupled with Poisson’s equation

- Algorithm based on fast fourier transform
- Iterative methods: NAG, PETSc and HYPRE library
- Direct methods: superLU, MUMPS and PaStiX
- Need for a fast and parallel library either MPI or MPI-OPENMP
- We started with sequential MUMPS solver
- We tested other parallel solvers depending on the case
We need to solve Poisson’s equation at each timestep

**Linear solver coupled with Poisson’s equation**

- Algorithm based on fast fourier transform
- Iterative methods: NAG, PETSc and HYPRE library
- Direct methods: superLU, MUMPS and PaStiX
- Need for a fast and parallel library either MPI or MPI-OPENMP
- We started with sequential MUMPS solver
- We tested other parallel solvers depending on the case
Streamer discharge simulation are known to be computationally expensive

Temporal multiscale nature of explicit streamer simulation: \( \Delta t = 10^{-12} - 10^{-14} \) s

Convection: \( \Delta t_c = \min \left( \frac{\Delta x_i}{v_x(i,j)}, \frac{\Delta r_j}{v_r(i,j)} \right) \)

Diffusion: \( \Delta t_d = \min \left( \frac{(\Delta x_i)^2}{D_x(i,j)}, \frac{(\Delta r_j)^2}{D_r(i,j)} \right) \)

Chemistry: \( \Delta t_I = \min \left( \frac{n_k(i,j)}{S_k(i,j)} \right) \)

Diel. relaxation: \( \Delta t_{\text{Diel}} = \min \left( \frac{\varepsilon_0}{q_e \mu e_{(i,j)} n_e(i,j)} \right) \)

Time scale of streamer propagation in centimeter gaps is \( \sim 10 \) ns, \( \rightarrow \sim 10^4 \) time steps

For centimeter gaps of 1 cm, \( \Delta x, r = 10 - 1 \) \( \mu \)m \( \rightarrow \) nbre of points > 1 \( \times 10^6 \)
Streamer discharge simulation are known to be computationally **expensive**

Temporal multiscale nature of **explicit** streamer simulation: $\Delta t = 10^{-12} \text{ s} \cdots 10^{-14} \text{ s}$

**Convection:** $\Delta t_c = \min \left[ \frac{\Delta x_i}{v_x(i,j)}, \frac{\Delta r_j}{v_r(i,j)} \right]$

**Diffusion:** $\Delta t_d = \min \left[ \frac{(\Delta x_i)^2}{D_x(i,j)}, \frac{(\Delta r_j)^2}{D_r(i,j)} \right]$

**Chemistry:** $\Delta t_I = \min \left[ \frac{n_k(i,j)}{S_k(i,j)} \right]$

**Diel. relaxation:** $\Delta t_{\text{Diel}} = \min \left[ \frac{\varepsilon_0}{q_{\mu}e(i,j)n_{e(i,j)}} \right]$

Time scale of streamer propagation in centimeter gaps is $\sim 10 \text{ ns} \rightarrow \sim 10^4 \text{ time steps}$

For centimeter gaps of $1 \text{ cm}$, $\Delta x,r = 10 \text{ } \mu \text{m} \rightarrow \text{nbre of points} > 1 \times 10^6$
Streamer discharge simulation are known to be computationally **expensive**.

Temporal multiscale nature of **explicit** streamer simulation: $\Delta t = 10^{-12} - 10^{-14}$ s

**Convection:**  
\[ \Delta t_c = \min \left[ \frac{\Delta x_i}{v_x(i,j)}, \frac{\Delta r_j}{v_r(i,j)} \right] \]

**Diffusion:**  
\[ \Delta t_d = \min \left[ \frac{(\Delta x_i)^2}{D_x(i,j)}, \frac{(\Delta r_j)^2}{D_r(i,j)} \right] \]

**Chemistry:**  
\[ \Delta t_I = \min \left[ \frac{n_k(i,j)}{S_k(i,j)} \right] \]

**Diel. relaxation:**  
\[ \Delta t_{\text{Diel}} = \min \left[ \frac{\varepsilon_0}{q\mu_e(i,j)n_e(i,j)} \right] \]

Time scale of streamer propagation in centimeter gaps is $\sim 10$ ns, $\rightarrow \sim 10^4$ time steps.

For centimeter gaps of 1 cm, $\Delta x,r = 10 - 1 \ \mu\text{m} \rightarrow \text{nbre of points} > 1 \times 10^6$
Streamer discharge simulation are known to be computationally **expensive**

Temporal multiscale nature of **explicit** streamer simulation: \( \Delta t=10^{-12} - 10^{-14} \text{s} \)

**Convection:** \( \Delta t_c = \min \left[ \frac{\Delta x_i}{v_x(i,j)}, \frac{\Delta r_j}{v_r(i,j)} \right] \)

**Diffusion:** \( \Delta t_d = \min \left[ \frac{(\Delta x_i)^2}{D_x(i,j)}, \frac{(\Delta r_j)^2}{D_r(i,j)} \right] \)

**Chemistry:** \( \Delta t_I = \min \left[ \frac{n_k(i,j)}{S_k(i,j)} \right] \)

**Diel. relaxation:** \( \Delta t_{Diel} = \min \left[ \frac{\varepsilon_0}{q\varepsilon\mu e(i,j) n_e(i,j)} \right] \)

Time scale of streamer propagation in centimeter gaps is \( \sim 10 \text{ ns} \), \( \rightarrow \sim 10^4 \text{ time steps} \)

For centimeter gaps of 1 cm, \( \Delta x,r=10 - 1 \ \mu\text{m} \rightarrow \text{nbre of points} > 1 \times 10^6 \)
Characteristics of the initial discharge code

- 2D-axisymmetric discharge code
- Full explicit sequential code using Cartesian non-uniform static mesh
- MUMPS direct solver for Poisson’s equation and photo-ionization source term
- Explicit Improved Scharfettel-Gummel (ISG) exponential scheme for the convection-diffusion equation
- 4th order Runge-kutta scheme for the chemistry source term
- 1st order operator splitting method: \( U^{t+\Delta t} = CD^{\Delta t} R^{\Delta t} U^{t} \)

**Verification of the code:**

**Validation of the code:**
Characteristics of the initial discharge code

- 2D-axisymmetric discharge code
- Full explicit sequential code using Cartesian non-uniform static mesh
- MUMPS direct solver for Poisson’s equation and photo-ionization source term
- Explicit Improved Scharfettel-Gummel (ISG) exponential scheme for the convection-diffusion equation
- 4th order Runge-kutta scheme for the chemistry source term
- 1st order operator splitting method: \( U^{t+\Delta t} = CD^{\Delta t} R^{\Delta t} U^{t} \)

**Verification of the code:**

**Validation of the code:**
# Table of contents

1. **Introduction on non-thermal discharges at atmospheric pressure**

2. **Rapid overview of the characteristics of streamer discharges**

3. **On the modeling of streamer discharges**

4. **Example of code improvements: test case**

5. **Conclusions**
1. Introduction on non-thermal discharges at atmospheric pressure

2. Rapid overview of the characteristics of streamer discharges

3. On the modeling of streamer discharges

4. Exemple of code improvements: test case

5. Conclusions
Performances of the discharge code: Test-case

- Studied electrode geometry: **point to plane**
  
  - Constant voltage applied at the anode, \( V_{\text{anode}} = +30 \text{ kV} \)
  
  - Computational domain is 2 cm \( \times \) 2 cm with Cartesian grid
  
  - Large domain size \( n_x \times n_r = 3353 \times 1725 \) so \( 5.8 \times 10^6 \) points
Studied electrode geometry: **point to plane**

Comparison with experiments:
10 mm gap with a sharp point electrode:

- Constant voltage applied at the anode, $V_{\text{anode}} = +30 \text{ kV}$

- Computational domain is 2 cm $\times$ 2 cm with Cartesian grid

- Large domain size $n_x \times n_r = 3353 \times 1725$
  so $5.8 \times 10^6$ points
Studied electrode geometry: **point to plane**

Comparison with experiments: **Test case**

10 mm gap with a sharp point electrode

Constant voltage applied at the anode, $V_{\text{anode}} = +30 \text{ kV}$

Computational domain is 2 cm $\times$ 2 cm with Cartesian grid

Large domain size $n_x \times n_r = 3353 \times 1725$ so $5.8 \times 10^6$ points
Performances of the discharge code: Test-case

- Ignition and propagation of a positive streamer discharge

- At $t_c = 3.0$ ns, discharge impacts the cathode plane

- Time step: $\Delta t = \Delta t_{\text{Diel}} \sim 10^{-14}$ s, dielectric relaxation time step $\Delta t_{\text{Diel}}$ 10 times smaller than $\Delta t_c$, $\Delta t_d$, $\Delta t_i$

- Simulation time: $\sim$ one month with original code (memory used > 30 Go)
Performances of the discharge code: Test-case

- One time-step $\Delta t$: more than 50% of the time for solving Poisson’s equation

- Potential $V$ + photoionization source term $S_{ph}$: $1+6 \times 3$ Poisson’s equation to solve

- Save computational time: $S_{ph}$ is computed every 5 time steps (negligible influence on results)

- In original code, direct solver MUMPS to solve Poisson’s equation:
  - $1 \times 10^6$ points $\rightarrow$ Memory (factorization): $520 \text{ Mo} \times (1+6) = 3.7 \text{ Go}$
  - $6 \times 10^6$ points $\rightarrow$ Memory (factorization): $4 \text{ Go} \times (1+6) = 28 \text{ Go}$
Performances of the discharge code: Test-case

- One time-step $\Delta t$: more than 50% of the time for solving Poisson’s equation

- Potential $V$ + photoionization source term $S_{ph}$: $1+6 \times 3$ Poisson’s equation to solve

- Save computational time: $S_{ph}$ is computed every 5 time steps (negligible influence on results)

- In original code, direct solver MUMPS to solve Poisson’s equation:
  - $1 \times 10^6$ points $\rightarrow$ Memory (factorization): $520 \text{ Mo} \times (1+6) = 3.7 \text{ Go}$
  - $6 \times 10^6$ points $\rightarrow$ Memory (factorization): $4 \text{ Go} \times (1+6) = 28 \text{ Go}$
Performances of the discharge code: Test-case

- One time-step $\Delta t$: more than 50% of the time for solving Poisson’s equation

- Potential $V$ + photoionization source term $S_{ph}: 1+6 \times 3$ Poisson’s equation to solve

- Save computational time: $S_{ph}$ is computed every 5 time steps (negligible influence on results)

- In original code, direct solver MUMPS to solve Poisson’s equation:
  - $1 \times 10^6$ points $\rightarrow$ Memory (factorization): $520 \text{ Mo} \times (1+6) = 3.7 \text{ Go}$
  - $6 \times 10^6$ points $\rightarrow$ Memory (factorization): $4 \text{ Go} \times (1+6) = 28 \text{ Go}$
Performances of the discharge code: Test-case

- One time-step $\Delta t$: more than 50% of the time for solving Poisson's equation

- Potential $V$ + photoionization source term $S_{ph} = 1 + 6 \times 3$ Poisson’s equation to solve

- Save computational time: $S_{ph}$ is computed every 5 time steps (negligible influence on results)

- In original code, direct solver MUMPS to solve Poisson’s equation:
  - $1 \times 10^6$ points $\rightarrow$ Memory (factorization): $520 \text{ Mo} \times (1+6) = 3.7 \text{ Go}$
  - $6 \times 10^6$ points $\rightarrow$ Memory (factorization): $4 \text{ Go} \times (1+6) = 28 \text{ Go}$
Performances of the discharge code: Test-case

- One time-step $\Delta t$: more than 50% of the time for solving Poisson's equation.

- Potential $V$ + photoionization source term $S_{ph}: 1+6 \times 3$ Poisson's equation to solve.

- Save computational time: $S_{ph}$ is computed every 5 time steps (negligible influence on results).

- In original code, direct solver MUMPS to solve Poisson's equation:
  - $1 \times 10^6$ points $\rightarrow$ Memory (factorization): $520 \text{ Mo} \times (1+6) = 3.7 \text{ Go}$
  - $6 \times 10^6$ points $\rightarrow$ Memory (factorization): $4 \text{ Go} \times (1+6) = 28 \text{ Go}$
Performances of the discharge code: Test-case

- One time-step $\Delta t$: more than 50% of the time for solving Poisson’s equation

- Potential $V +$ photoionization source term $S_{ph}: 1+6 \times 3$ Poisson’s equation to solve

- Save computational time: $S_{ph}$ is computed every 5 time steps (negligible influence on results)

- In original code, direct solver MUMPS to solve Poisson’s equation:
  - $1 \times 10^6$ points $\rightarrow$ Memory (factorization): $520 \text{ Mo} \times (1+6) = 3.7 \text{ Go}$
  - $6 \times 10^6$ points $\rightarrow$ Memory (factorization): $4 \text{ Go} \times (1+6) = 28 \text{ Go}$

Limitations of the initial discharge code:

- Number of points for large simulated domains
- Solution time to solve Poisson’s equation
- Small time-step $\Delta t = \Delta t_{Diel} \sim 10^{-14}\text{s}$
Strategy to improve the computational efficiency of the code

**Limitations of the initial discharge code:**

- Number of points for large simulated domains
- Solution time to solve Poisson’s equation
- Small time-step $\Delta t = \Delta t_{\text{Diel}} \sim 10^{-14}$ s

**Number of points:** Adaptive Mesh Refinement (AMR)

- Parallel (MPI) AMR code (use of PARAMESH) with a fluid model for the simulation of filamentary discharge (2D-3D)

- Parallel (MPI) AMR code with a hybrid particle-fluid model for the simulation of streamer discharge (2D-3D)

**Poisson’s equation:** implement parallel protocols (OPENMP and MPI)

**Small time-steps:** improve the numerical schemes implemented
Strategy to improve the computational efficiency of the code

Limitations of the initial discharge code:

- Number of points for large simulated domains
- Solution time to solve Poisson’s equation
- Small time-step $\Delta t = \Delta t_{Diel} \sim 10^{-14} s$

- Number of points: Adaptive Mesh Refinement (AMR)
  - Parallel (MPI) AMR code (use of PARAMESH) with a fluid model for the simulation of filamentary discharge (2D-3D)
  - Parallel (MPI) AMR code with a hybrid particle-fluid model for the simulation of streamer discharge (2D-3D)

- Poisson’s equation: implement parallel protocols (OPENMP and MPI)

- Small time-steps: improve the numerical schemes implemented
On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go

High number of points → iterative solver becomes competitive

Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)

Test on TC of laplacian potential: 72 MPI processes:

Test on TC of laplacian potential: 24 MPI×3 OPENMP: 5.7 s ↘ 0.4 s

For iterative solver SMG, memory required is less than 1 Go
Improvements of the discharge code: Poisson’s solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go
- High number of points $\rightarrow$ iterative solver becomes competitive
- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)
- Test on TC of laplacian potential: 72 MPI processes:
  - Test on TC of laplacian potential: 24 MPI $\times$ 3 OPENMP: 5.7 s $\downarrow$ 0.4 s
- For iterative solver SMG, memory required is less than 1 Go
Improvements of the discharge code: Poisson’s solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go

- High number of points $\rightarrow$ iterative solver becomes competitive

- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)

- Test on TC of laplacian potential: 72 MPI processes: 5.7 s $\rightarrow$ 0.5 s

- Test on TC of laplacian potential: 24 MPI $\times$ 3 OPENMP: 5.7 s $\rightarrow$ 0.4 s

- For iterative solver SMG, memory required is less than 1 Go
On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go

High number of points → iterative solver becomes competitive

Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)

Test on TC of laplacian potential:
72 MPI processes: 5.7 s ↘ 0.5 s

Test on TC of laplacian potential:
24 MPI×3 OPENMP: 5.7 s ↘ 0.4 s

For iterative solver SMG, memory required is less than 1 Go
On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go

High number of points → iterative solver becomes competitive

Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)

Test on TC of laplacian potential:
72 MPI processes: 5.7 s \(\downarrow\) 0.5 s

Test on TC of laplacian potential:
24 MPI×3 OPENMP: 5.7 s \(\downarrow\) 0.4 s

For iterative solver SMG, memory required is less than 1 Go
Improvements of the discharge code: Poisson’s solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go

- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)

- Test on TC of laplacian potential: 72 MPI processes: 5.7 s ↓ 0.5 s

- Test on TC of laplacian potential: 24 MPI×3 OPENMP: 5.7 s ↓ 0.4 s

- For iterative solver SMG, memory required is less than 1 Go

Solved problems:

- Memory requirement
- Solution time to solve Poisson’s equation
- Small time-step: constraint of $\Delta t_{Diel}$?
Improvements of the discharge code: Poisson’s solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go

- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)

- Test on TC of laplacian potential: 72 MPI processes: 5.7 s ↓ 0.5 s

- Test on TC of laplacian potential: 24 MPI × 3 OPENMP: 5.7 s ↓ 0.4 s

- For iterative solver SMG, memory required is less than 1 Go

Solved problems:

- Memory requirement
- Solution time to solve Poisson’s equation
- Small time-step: constraint of $\Delta t_{\text{Diel}}$?
Improvements of the discharge code: "semi-implicit" scheme

- To remove $\Delta t_{\text{Diel}}$, implementation of a "semi-implicit" scheme
  Lin et al., *Computer Physics Communications* 183, (2012)
- On the test-case, we compare the implementation with the "semi-implicit" scheme with the full explicit model:

![Graph comparing semi-implicit and full explicit schemes](image)

- We can choose a time-step 10 bigger than with the explicit model.
To remove $\Delta t_{\text{Diel}}$, implementation of a "semi-implicit" scheme

On the test-case, we compare the implementation with the "semi-implicit" scheme with the full explicit model:

We can choose a time-step 10 bigger than with the explicit model.
To improve the computational efficiency, we need a transport scheme that is accurate with less points.

- Is the ISG exponential transport scheme (drift+diffusion) accurate with less points?
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x=1 \, \mu m, \Delta r=1 \, \mu m$
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x=2.5 \, \mu m, \Delta r=1 \, \mu m$

The UNO3 scheme, explicit $3^{rd}$ order accurate + explicit $2^{nd}$ order for the diffusion

Improvements of the discharge code: UNO3 convection scheme

To improve the computational efficiency, we need a transport scheme that is accurate with less points.

- Is the ISG exponential transport scheme (drift+diffusion) accurate with less points?
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x=1 \ \mu m, \Delta r=1 \ \mu m$
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x=2.5 \ \mu m, \Delta r=1 \ \mu m$

- The UNO3 scheme, explicit 3"rd order accurate + explicit 2"nd order for the diffusion

Improvements of the discharge code: UNO3 convection scheme

To improve the computational efficiency, we need a transport scheme that is accurate with less points.

- Is the ISG exponential transport scheme (drift+diffusion) accurate with less points?
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x = 1 \mu m, \Delta r = 1 \mu m$
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x = 2.5 \mu m, \Delta r = 1 \mu m$

- The UNO3 scheme, explicit 3rd order accurate + explicit 2nd order for the diffusion

Improvements of the discharge code: UNO3 convection scheme

To improve the computational efficiency, we need a transport scheme that is accurate with less points.

- Is the ISG exponential transport scheme (drift+diffusion) accurate with less points?
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x=1 \ \mu m, \Delta r=1 \ \mu m$
  - Test-case 2: $E_{axis}$ profile with close to the point $\Delta x=2.5 \ \mu m, \Delta r=1 \ \mu m$

- The UNO3 scheme, explicit 3$^{rd}$ order accurate + explicit 2$^{nd}$ order for the diffusion

Improvements of the discharge code: MPI-OPENMP discharge code

- **Full parallel MPI-OPENMP discharge code**
- **Poisson’s equation:** MPI-OPENMP iterative solver SMG
- **Small time-steps:** Semi-implicit scheme (to remove $\Delta t_{Diel}$)
- **Robustness:** Explicit UNO3 scheme 3$^{rd}$ order for convection + Explicit 2$^{nd}$ order for diffusion (not shown here)

- Test on TC one time-step:
  - Test on TC one time-step:
  - 24 MPI x 3 OPENMP: 17.05 s $\downarrow$ 0.86 s
- TC is computed in $\sim$ 3 hours (one month with initial code)
Improvements of the discharge code: MPI-OPENMP discharge code

- Full parallel MPI-OPENMP discharge code
- Poisson’s equation: MPI-OPENMP iterative solver SMG
- Small time-steps: Semi-implicit scheme (to remove $\Delta t_{\text{Diel}}$)
- Robustness: Explicit UNO3 scheme $3^{rd}$ order for convection + Explicit $2^{nd}$ order for diffusion (not shown here)

Test on TC one time-step: Full parallelization of the code

- Test on TC one time-step: 24 MPI x 3 OPENMP: 17.05 s $\downarrow$ 0.86 s
- TC is computed in $\sim$ 3 hours (one month with initial code)
Improvements of the discharge code: MPI-OPENMP discharge code

- **Full parallel MPI-OPENMP discharge code**
- **Poisson’s equation**: MPI-OPENMP iterative solver SMG
- **Small time-steps**: Semi-implicit scheme (to remove $\Delta t_{\text{Diel}}$)
- **Robustness**: Explicit UNO3 scheme 3\textsuperscript{rd} order for convection + Explicit 2\textsuperscript{nd} order for diffusion (not shown here)

- **Test on TC one time-step**: 72 MPI processes: 17.05 s $\searrow$ 0.63 s
- **Test on TC one time-step**: 24 MPI $\times$ 3 OPENMP: 17.05 s $\searrow$ 0.86 s
- **TC is computed in $\sim$ 3 hours**
  (one month with initial code)
Improvements of the discharge code: MPI-OPENMP discharge code

- **Full parallel MPI-OPENMP discharge code**
- **Poisson’s equation**: MPI-OPENMP iterative solver SMG
- **Small time-steps**: Semi-implicit scheme (to remove $\Delta t_{\text{Diel}}$)
- **Robustness**: Explicit UNO3 scheme $3^{rd}$ order for convection + Explicit $2^{nd}$ order for diffusion (not shown here)

- **Test on TC one time-step**:
  72 MPI processes: 17.05 s $\rightarrow$ 0.63 s
  24 MPI $\times$ 3 OPENMP: 17.05 s $\rightarrow$ 0.86 s
- **TC is computed in $\sim$ 3 hours**
  (one month with initial code)
Comparison of results with experiments

Numerical/Experimental comparison

- Diameter of the discharge: 8 mm / 8 mm
- Velocity of the discharge:
  \[ v_{num} = 2.6 \times 10^8 \text{ cm.s}^{-1} \]
  \[ v_{exp} = 2.6 - 3.2 \times 10^8 \text{ cm.s}^{-1} \]

Good agreement with experiments

1. Introduction on non-thermal discharges at atmospheric pressure
2. Rapid overview of the characteristics of streamer discharges
3. On the modeling of streamer discharges
4. Exemple of code improvements: test case
5. Conclusions
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction on non-thermal discharges at atmospheric pressure</td>
</tr>
<tr>
<td>2</td>
<td>Rapid overview of the characteristics of streamer discharges</td>
</tr>
<tr>
<td>3</td>
<td>On the modeling of streamer discharges</td>
</tr>
<tr>
<td>4</td>
<td>Exemple of code improvements: test case</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions</td>
</tr>
</tbody>
</table>
For plasma applications, we need of course fast, robust and parallel iterative solvers (HYPRE is used for now)

Even on 2D structured grids the computation can be very intensive

Next step is mostly to use AMR meshes as well as improving the physical model (hybrid models)

Currently in a joint project between LPP, CERFACS and SNECMA that just started, we also need to solve Poisson’s equation to do a 3D simulation of a Hall thruster

We are modifying a 3D massively parallel code AVBP to carry out these simulations but as of now this code does not solve Poisson’s equation

We are implementing now a laplacian operator in a 3D unstructured mesh framework based on a finite volume discretization coupled with the HYPRE library for the solving part

Unstructured or not we need to implement accurate discretization and fast, robust and parallel library to solve Poisson’s equation
Thank you for your attention.
5 mm gap with a point to plane geometry with a stump point electrode

Constant voltage applied at the anode, \( V_{\text{anode}} = +13 \) kV

Computational domain is 1 cm \( \times \) 17 cm with Cartesian grid

Small domain size \( n_x \times n_r = 1287 \times 1000 \) so \( 1.3 \times 10^6 \) points
Propagation of a cathode directed streamer from the point anode to the cathode plane

At $t=3.6$ ns, $E_{axis} = 110$ kV.cm$^{-1}$ and $n_e = 3 \times 10^{13}$ cm$^{-3}$

At $t_c=6.0$ ns, discharge impacts the cathode plane

General time step: $\Delta t=10^{-12}$s

Simulation time : $\sim 4$ hours

Improvements of computational time: introduce parallel protocols
Improvements of the discharge code: Test-case 1

- Poisson’s equation is the most expensive equation to solve
- 1 for Poisson’s equation and 6 (each iterated × 3) for photoionisation source term

First step: introduction of shared memory OPENMP protocols:
- Change of direct solver: MUMPS (MPI only) to PaStiX (MPI-OPENMP)
- OPENMP protocols in the rest of the code

<table>
<thead>
<tr>
<th>Code with:</th>
<th>MUMPS</th>
<th>PaStiX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory:</td>
<td>664 Mo</td>
<td>886 Mo</td>
</tr>
<tr>
<td>Nb thread</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Factorization (s)</td>
<td>64.12</td>
<td>24.87</td>
</tr>
<tr>
<td>Solution (s)</td>
<td>1.27</td>
<td>0.64</td>
</tr>
<tr>
<td>One time-step (s)</td>
<td>5.76</td>
<td>2.71</td>
</tr>
</tbody>
</table>

With 6 threads:
- Speed up Poisson’s equation: 5.77
- Speed up one time step: 4.2

Computational time for TC-1: 4 hours → ~ 40 minutes