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

François Pechereau[1] and Anne Bourdon[2]

[1]Laboratory CERFACS, Toulouse
[2]Laboratory LPP, Ecole Polytechnique
on leave from laboratory EM2C, Ecole Centrale Paris

Journée problème de Poisson, January 26, 2015, IHP, France

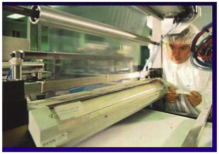
Table of contents

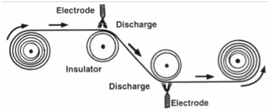
Table of contents

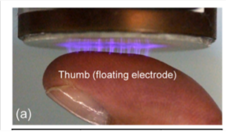
Non-thermal discharges at atmospheric pressure

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 $\rightarrow$ possible at ground pressure to reduce costs (no need for pumping systems) ?
- New applications as biomedical applications, plasma assisted combustion







Plasma assisted combustion

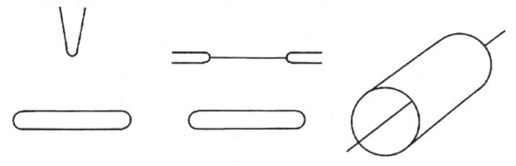
Φ= 0.8, Air flow rate = 15 m^3/h

Lean premixed burner

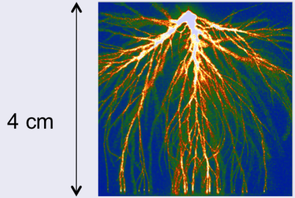
How to generate non-thermal discharges at atmospheric pressure ?

Between two metallic electrodes

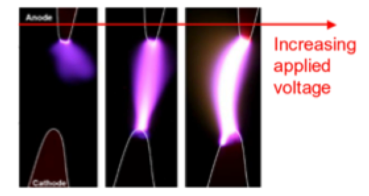
- Interelectrode gaps of a few mm to a few cm at P_{atm}



- Risk: If the voltage pulse is too long $\rightarrow$ transition to spark

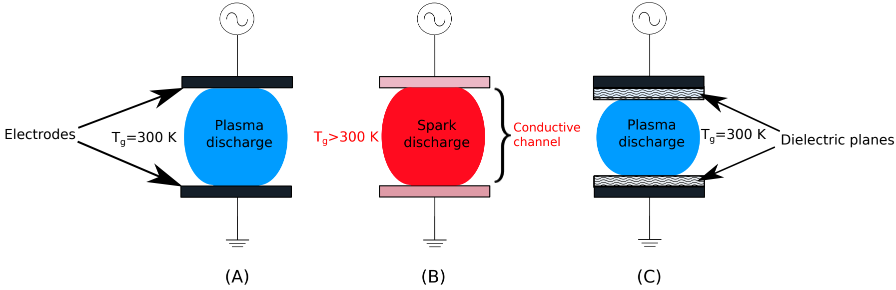


Briels, PhD (2007)



Pai, D., 2008 Ph.D. thesis, Ecole Centrale Paris, France.

How to generate non-thermal discharges at atmospheric pressure ?

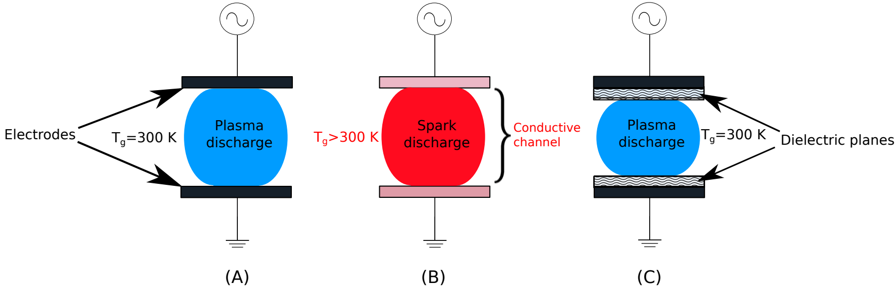


Dielectric Barrier Discharge (DBD)

- (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**

- $T_i = T_g = 300$K, $T_e > 10000K$ → **Cold plasma**
- O, OH, radicals and UV radiation

How to generate non-thermal discharges at atmospheric pressure ?

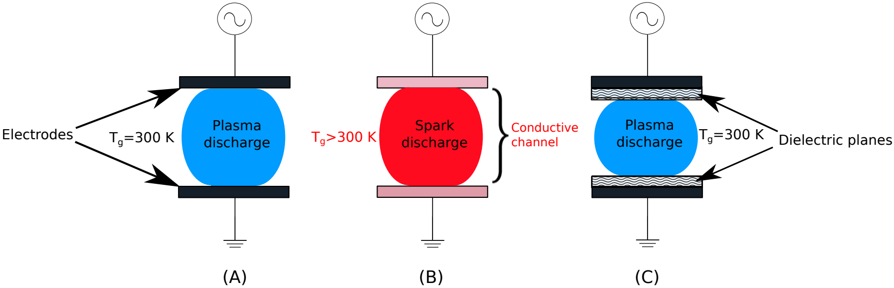


Dielectric Barrier Discharge (DBD)

- (A) Ignition of a discharge between electrodes
- (B) **Transition to spark** $\rightarrow$ 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 ?

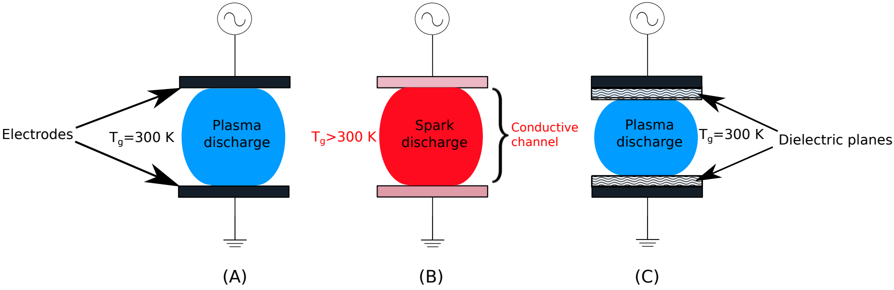


Dielectric Barrier Discharge (DBD)

- (A) Ignition of a discharge between electrodes
- (B) **Transition to spark** $\rightarrow$ 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)

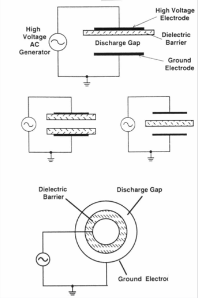
- (A) Ignition of a discharge between electrodes
- (B) **Transition to spark** $\rightarrow$ high current, $T_g > 300K$
- (C) To prevent spark transition: **dielectric layers between the electrodes**

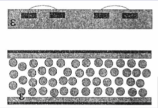
- $T_i = T_g = 300K$, $T_e > 10000K$ $\rightarrow$ **Cold plasma**
- O, OH, radicals and UV radiation

How to generate non-thermal discharges at atmospheric pressure ?

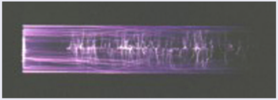
Dielectric Barrier Discharge (DBD)

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

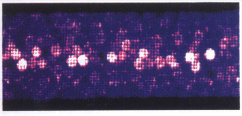




Plane-plane reactor (LPGP Orsay)



Wire-cylinder (GREMI Orléans)

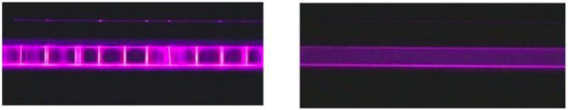


H.Russ et al , *IEEE Trans. Plasma Sci.* **27** (1999) 38

Non-thermal discharges at atmospheric pressure

Structure of P_{atm} discharges

- At P_{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 $\rightarrow$ high density of active species (radicals, excited species). However, local heating may be significant

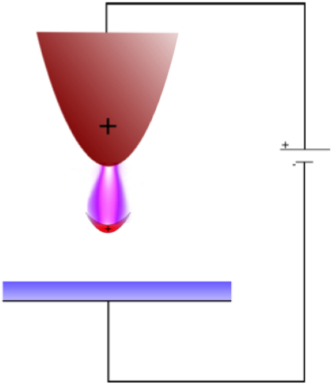
Diffusive discharges

- Low density of electrons, large volume of the discharge and negligible heating

Table of contents

Table of contents

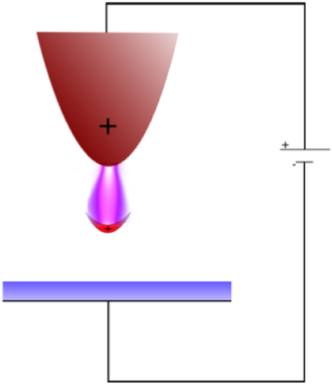
Streamer propagation in air at atmopsheric pressure



Streamer propagation

- In air at P_{atm}, the beakdown 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

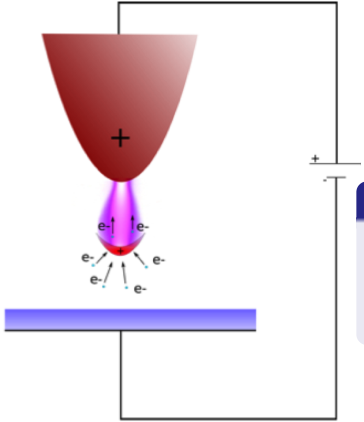
Streamer propagation in air at atmopsheric pressure



Streamer propagation

- 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

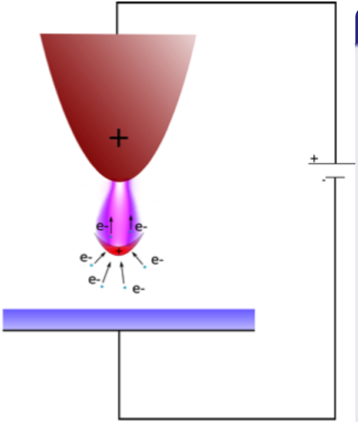
Streamer propagation in air at atmopsheric pressure



Streamer propagation

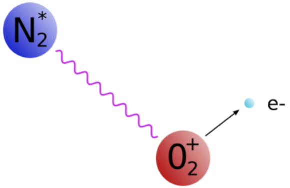
- The discharge move towards the cathode, whereas electrons move towards the anode
- Need for seed electrons for the discharge propagation

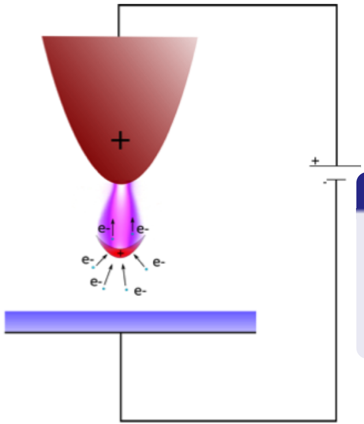
Streamer propagation in air at atmopsheric pressure



Origin of seed electrons

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

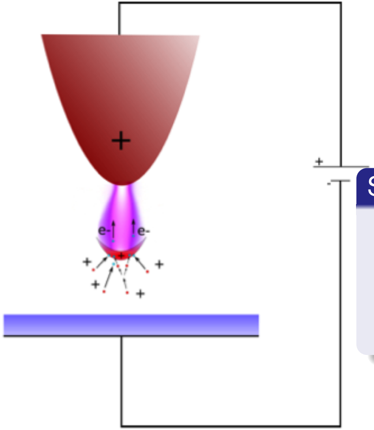




Streamer propagation

- Seed electrons in front of the streamer
- Transport of charged species
- Screening of the streamer head by electrons
- Streamer moves forward

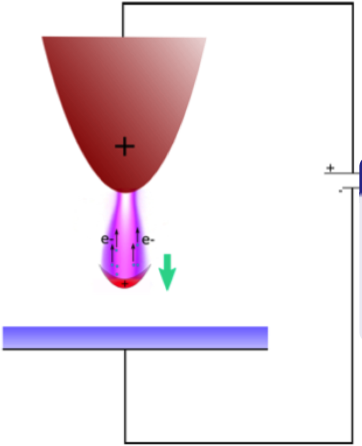
Streamer propagation in air at atmopsheric pressure



Streamer propagation

- 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 atmopsheric pressure



Streamer propagation

- 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 space: Li, Ebert and Brook IEEE Trans. Plasma Sci. (2008), Li, Ebert, Hundsdorfer, JCP (2012)
    - In energy: *bulk-model* (Bonaventura et al., ERL (2014))

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 space: Li, Ebert and Brook IEEE Trans. Plasma Sci. (2008), Li, Ebert, Hundsdorfer, JCP (2012)
    - In energy: *bulk-model* (Bonaventura et al., ERL (2014))

Table of contents

Table of contents

2D Fluid model for discharge in air at $\mathrm{P}_{\mathrm{atm}}$

- Continuity equation is solved for electrons, positive and negative ions

$$\frac{\partial n_i}{\partial t} + \mathrm{div}\, \boldsymbol{j}_i = S_i \tag{1}$$

- Drift-diffusion approximation

$$\boldsymbol{j}_i = \mu_i\, n_i\, \boldsymbol{E} - D_i\, \mathbf{grad}\, n_i \tag{2}$$

- Poisson's equation:

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_{\mathrm{r}} \boldsymbol{\nabla} V) = -q_{\mathrm{e}}(n_{\mathrm{p}} - n_{\mathrm{n}} - n_{\mathrm{e}}) \tag{3}$$

2D Fluid model for discharge in air at $\mathrm{P_{atm}}$

- Continuity equation is solved for electrons, positive and negative ions

$$\frac{\partial n_i}{\partial t} + \mathrm{div}\, \boldsymbol{j}_i = S_i \tag{1}$$

- Drift-diffusion approximation

$$\boldsymbol{j}_i = \mu_i\, n_i\, \boldsymbol{E} - D_i\, \mathbf{grad}\, n_i \tag{2}$$

- Poisson's equation:

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{r} \boldsymbol{\nabla} V) = -q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}) \tag{3}$$

- 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

2D Fluid model for discharge in air at $\mathrm{P_{atm}}$

- Continuity equation is solved for electrons, positive and negative ions

$$\frac{\partial n_i}{\partial t} + \mathrm{div}\, \boldsymbol{j}_i = S_i \tag{4}$$

- Drift-diffusion approximation

$$\boldsymbol{j}_i = \mu_i \, n_i \, \boldsymbol{E} - D_i \, \mathbf{grad} \, n_i \tag{5}$$

- Source terms for air:

$$\begin{cases} \mathrm{S_e} = (\partial_t n_e)_{\mathrm{chem}} = (\nu_\alpha - \nu_\eta - \beta_{\mathrm{ep}} n_{\mathrm{p}})\, n_{\mathrm{e}} + \nu_{\mathrm{det}} n_{\mathrm{n}} + S_{ph} \,, \\ \mathrm{S_n} = (\partial_t n_n)_{\mathrm{chem}} = -(\nu_{\mathrm{det}} + \beta_{\mathrm{np}} n_{\mathrm{p}})\, n_{\mathrm{n}} + \nu_\eta n_{\mathrm{e}} \,, \\ \mathrm{S_p} = (\partial_t n_p)_{\mathrm{chem}} = -(\beta_{\mathrm{ep}} n_{\mathrm{e}} + \beta_{\mathrm{np}} n_{\mathrm{n}})\, n_{\mathrm{p}} + \nu_\alpha n_{\mathrm{e}} + S_{ph} \,. \end{cases} \tag{6}$$

- Local field approximation: $\nu_\alpha(|\vec{E}|/N)$, $\nu_\eta(|\vec{E}|/N)$, $\mu_i(|\vec{E}|/N)$, $D_i(|\vec{E}|/N)$
Morrow et al., *J.Phys. D:Appl. Phys.* **30**,(1997)

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

2D Fluid model for discharge in air at $\mathrm{P_{atm}}$

- Continuity equation is solved for electrons, positive and negative ions

$$\frac{\partial n_i}{\partial t} + \mathrm{div}\, \boldsymbol{j}_i = S_i \tag{4}$$

- Drift-diffusion approximation

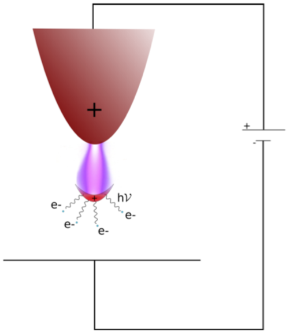
$$\boldsymbol{j}_i = \mu_i \, n_i \, \boldsymbol{E} - D_i \, \mathbf{grad} \, n_i \tag{5}$$

- Source terms for air:

$$\begin{cases} \mathrm{S_e} = (\partial_t n_e)_{chem} = (\nu_\alpha - \nu_\eta - \beta_{ep} n_p)\, n_e + \nu_{det} n_n + S_{ph} \,, \\ \mathrm{S_n} = (\partial_t n_n)_{chem} = -(\nu_{det} + \beta_{np} n_p)\, n_n + \nu_\eta n_e \,, \\ \mathrm{S_p} = (\partial_t n_p)_{chem} = -(\beta_{ep} n_e + \beta_{np} n_n)\, n_p + \nu_\alpha n_e + S_{ph} \,. \end{cases} \tag{6}$$

- Local field approximation: $\nu_\alpha(|\vec{E}|/N)$, $\nu_\eta(|\vec{E}|/N)$, $\mu_i(|\vec{E}|/N)$, $D_i(|\vec{E}|/N)$
Morrow et al., *J.Phys. D:Appl. Phys.* **30**,(1997)
- 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
- Reference model derived from experimental results *[Zheleznyak, et al., High Temp., 20, 357 (1982)]* and confirmed by recent experiments *[Aints, et al., Plasma Process. and Polym. 5, 672 (2008)]*
- 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 $\rightarrow$ differential model *[Bourdon et al. PSST, 16, 656 (2007), Liu et al. APL 91, 211501 (2007)]*

- Poisson's equation

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{r} \boldsymbol{\nabla} V) = -q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}) \tag{7}$$

- Photoionization source term S_{ph}: SP3 model
  Bourdon et al., *Plasma Sources Sci. Technol.* **16**, (2007)
- It leads to solve 18+1 Poisson's equation !

2D Fluid model for discharge in air at $\mathrm{P}_{\mathrm{atm}}$

- Poisson's equation

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{r} \boldsymbol{\nabla} V) = -q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}) \tag{7}$$

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

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

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

$$\downarrow$$

$$\boxed{6 \times 3 \text{ Poisson's equations}} \rightarrow S_{ph} = \sum_{\mathrm{j}} = f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r}))$$

- It leads to solve 18+1 Poisson's equation !

2D Fluid model for discharge in air at $\mathrm{P}_{\mathrm{atm}}$

- Poisson's equation

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_{\mathrm{r}} \boldsymbol{\nabla} V) = -q_{\mathrm{e}}(n_{\mathrm{p}} - n_{\mathrm{n}} - n_{\mathrm{e}}) \tag{7}$$

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

$$\begin{cases} \nabla^2 \phi_{1,\mathbf{j}}(\vec{r}) - A_{1,\mathbf{j}}\phi_{1,\mathbf{j}} &= S_{1,\mathbf{j}} \\ \nabla^2 \phi_{2,\mathbf{j}}(\vec{r}) - A_{2,\mathbf{j}}\phi_{2,\mathbf{j}} &= S_{2,\mathbf{j}}; \end{cases} \tag{8}$$

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

$$\downarrow$$

$$\boxed{6 \times 3 \text{ Poisson's equations}} \rightarrow S_{ph} = \sum_{\mathrm{j}} = f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r}))$$

- It leads to solve 18+1 Poisson's equation !

2D Fluid model for discharge in air at $\mathrm{P_{atm}}$

- Poisson's equation

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{r} \boldsymbol{\nabla} V) = -q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}) \tag{7}$$

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

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

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

$$\downarrow$$

$$\boxed{6 \times 3 \text{ Poisson's equations}} \rightarrow S_{ph} = \sum_{\mathrm{j}} = f(\phi_{1,j}(\vec{r}), \phi_{2,j}(\vec{r}))$$

- It leads to solve 18+1 Poisson's equation !

2D Fluid model for discharge in air at $\mathrm{P_{atm}}$

- Poisson's equation

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{r} \boldsymbol{\nabla} V) = -q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}) \tag{7}$$

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

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

$\lambda_{\mathbf{j}=\underline{1,3}} \rightarrow$ 2 Poisson's equation ($\phi_{1,\mathbf{j}}(\vec{r})$ and $\phi_{2,\mathbf{j}}(\vec{r})$) $\times$ **3 iterations for BC**

$$\downarrow$$

$$\boxed{6 \times 3 \text{ Poisson's equations}} \rightarrow S_{ph} = \sum_{\mathbf{j}} = f(\phi_{1,\mathbf{j}}(\vec{r}), \phi_{2,\mathbf{j}}(\vec{r}))$$

- It leads to solve 18+1 Poisson's equation !

2D Fluid model for discharge in air at P_{atm}

- Poisson's equation

$$\varepsilon_0 \boldsymbol{\nabla} \cdot (\varepsilon_{\mathrm{r}} \boldsymbol{\nabla} V) = -q_{\mathrm{e}}(n_{\mathrm{p}} - n_{\mathrm{n}} - n_{\mathrm{e}}) \tag{7}$$

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

$$\begin{cases} \nabla^2 \phi_{1,\mathbf{j}}(\vec{r}) - A_{1,\mathbf{j}} \phi_{1,\mathbf{j}}(\vec{r}) &= S_{1,\mathbf{j}} \\ \nabla^2 \phi_{2,\mathbf{j}}(\vec{r}) - A_{2,\mathbf{j}} \phi_{2,\mathbf{j}}(\vec{r}) &= S_{2,\mathbf{j}}; \end{cases} \tag{8}$$

$\lambda_{\mathbf{j}=\underline{1,3}} \rightarrow$ 2 Poisson's equation ($\phi_{1,\mathbf{j}}(\vec{r})$ and $\phi_{2,\mathbf{j}}(\vec{r})$) $\times$ **3 iterations for BC**

$$\downarrow$$

$$\boxed{6 \times 3 \text{ Poisson's equations}} \rightarrow S_{ph} = \sum_{\mathbf{j}} = f(\phi_{1,\mathbf{j}}(\vec{r}), \phi_{2,\mathbf{j}}(\vec{r}))$$

- It leads to solve 18+1 Poisson's equation !

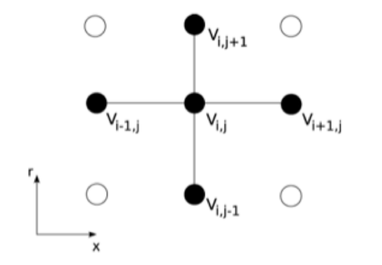
2nd order finite volume discretization in 2D

- In cylindrical coordinates, Poisson's equation can be written as

$$-\frac{\partial}{\partial x}\left(\epsilon\frac{\partial V}{\partial x}\right)-\frac{1}{r}\frac{\partial}{\partial r}\left(\epsilon r\frac{\partial V}{\partial r}\right)=\rho(x,r)\ , \tag{9}$$

- After integration it leads to

$$V^E_{i,j}V_{i+1,j}+V^W_{i,j}V_{i-1,j}+V^S_{i,j}V_{i,j-1}+V^N_{i,j}V_{i,j+1}+V^C_{i,j}V_{i,j}=\rho_{i,j}\Omega_{i,j}\ , \tag{10}$$



Discretization of Poisson's equation

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

Discretization of Poisson's equation

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

Discretization of Poisson's equation

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

Simulation of streamer discharge

- Streamer discharge simulation are known to be computationally **expensive**
- Temporal multiscale nature of **explicit** streamer simulation: Δt=$10^{-12} - 10^{-14}$s

$$\underline{\text{Convection:}} \quad \Delta t_c = \min\left[\frac{\Delta x_i}{v_{x(i,j)}}, \frac{\Delta r_j}{v_{r(i,j)}}\right] \qquad \underline{\text{Diffusion:}} \quad \Delta t_d = \min\left[\frac{(\Delta x_i)^2}{D_{x(i,j)}}, \frac{(\Delta r_j)^2}{D_{r(i,j)}}\right]$$

$$\underline{\text{Chemistry:}} \quad \Delta t_l = \min\left[\frac{n_{k(i,j)}}{S_{k(i,j)}}\right] \qquad \underline{\text{Diel. relaxation:}} \quad \Delta t_{Diel} = \min\left[\frac{\varepsilon_0}{q_e \mu_{e(i,j)} n_{e(i,j)}}\right]$$

- Time scale of streamer propagation in centimeter gaps is $\sim$ 10 ns, $\rightarrow \sim 10^4$ time steps
- For centimeter gaps of 1 cm, Δx,r=$10 - 1$ μm $\rightarrow$ nbre of points > 1×10^6

Simulation of streamer discharge

- Streamer discharge simulation are known to be computationally **expensive**
- Temporal multiscale nature of **explicit** streamer simulation: Δt=$10^{-12} - 10^{-14}$s

<u>Convection:</u> $\Delta t_c = \min \left[\frac{\Delta x_i}{v_{x(i,j)}} , \frac{\Delta r_j}{v_{r(i,j)}} \right]$ <u>Diffusion:</u> $\Delta t_d = \min \left[\frac{(\Delta x_i)^2}{D_{x(i,j)}} , \frac{\left(\Delta r_j\right)^2}{D_{r(i,j)}} \right]$

<u>Chemistry:</u> $\Delta t_l = \min \left[\frac{n_{k(i,j)}}{S_{k(i,j)}} \right]$ <u>Diel. relaxation:</u> $\Delta t_{Diel} = \min \left[\frac{\varepsilon_0}{q_e \mu_{e(i,j)} n_{e(i,j)}} \right]$

- Time scale of streamer propagation in centimeter gaps is $\sim$ 10 ns, $\rightarrow \sim 10^4$ time steps
- For centimeter gaps of 1 cm, Δx,r=$10 - 1\ \mu$m $\rightarrow$ nbre of points $> 1\times 10^6$

Simulation of streamer discharge

- Streamer discharge simulation are known to be computationally **expensive**
- Temporal multiscale nature of **explicit** streamer simulation: Δt=$10^{-12} - 10^{-14}$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: $\Delta t_d = \min\left[\frac{(\Delta x_i)^2}{D_{x(i,j)}}, \frac{\left(\Delta r_j\right)^2}{D_{r(i,j)}}\right]$

Chemistry: $\Delta t_I = \min\left[\frac{n_{k(i,j)}}{S_{k(i,j)}}\right]$ Diel. relaxation: $\Delta t_{Diel} = \min\left[\frac{\varepsilon_0}{q_e \mu_{e(i,j)} n_{e(i,j)}}\right]$

- Time scale of streamer propagation in centimeter gaps is $\sim$ 10 ns, $\rightarrow \sim 10^4$ time steps
- For centimeter gaps of 1 cm, Δx,r=$10 - 1$ μm $\rightarrow$ nbre of points $> 1\times 10^6$

Simulation of streamer discharge

- Streamer discharge simulation are known to be computationally **expensive**
- Temporal multiscale nature of **explicit** streamer simulation: Δt=$10^{-12} - 10^{-14}$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: $\Delta t_l = \min\left[\frac{n_{k(i,j)}}{S_{k(i,j)}}\right]$ Diel. relaxation: $\Delta t_{Diel} = \min\left[\frac{\varepsilon_0}{q_e \mu_{e(i,j)} n_{e(i,j)}}\right]$

- Time scale of streamer propagation in centimeter gaps is $\sim$ 10 ns, $\rightarrow \sim 10^4$ time steps
- For centimeter gaps of 1 cm, Δx,r=$10 - 1$ μm $\rightarrow$ nbre of points > 1×10^6

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
  Kulikovsky A., *Journal of Computational Physics* **11**, 149-155,(1995)
- 4^{th} order Runge-kutta scheme for the chemistry source term
- 1^{st} order operator splitting method: $U^{t+\Delta t} = CD^{\Delta t} R^{\Delta t} U^t$
- **Verification of the code:**
  Celestin et al, *Journal of Physics D:Applied Physics.* **42**, 065203 (2009)
  S. Celestin, PhD thesis, (2008)
- **Validation of the code:**
  Jánský et al., *Journal of Physics D:Applied Physics.* **44**, 335201 (2011)

Characteristics of the initial discharge code

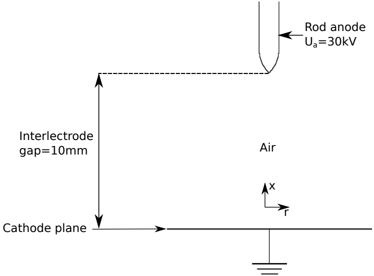
- 2D-axisymmetric discharge code
- Full explicit sequential code using Cartesian non-uniform static mesh
- MUMPS direct solver for Poisson's equation and photo-ionization source term
- Explicit Improved Scharfettel-Gummel (ISG) exponential scheme for the convection-diffusion equation
  Kulikovsky A., *Journal of Computational Physics* **11**, 149-155,(1995)
- 4^{th} order Runge-kutta scheme for the chemistry source term
- 1^{st} order operator splitting method: $U^{t+\Delta t} = CD^{\Delta t}\ R^{\Delta t}\ U^{t}$

- **Verification of the code:**
  Celestin et al, *Journal of Physics D:Applied Physics.* **42**, 065203 (2009)
  S. Celestin, PhD thesis, (2008)
- **Validation of the code:**
  Jánský et al., *Journal of Physics D:Applied Physics.* **44**, 335201 (2011)

Table of contents

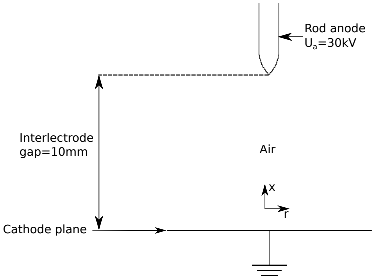
Table of contents

- Studied electrode geometry: **point to plane**
- Constant voltage applied at the anode, V_{anode} = **+30 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



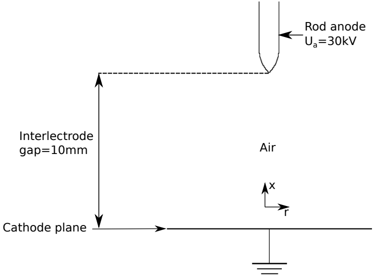
Performances of the discharge code: Test-case

- Studied electrode geometry: **point to plane**
- Comparison with experiments:
  10 mm gap with a sharp point electrode:
- Constant voltage applied at the anode,
  V_{anode} = +30 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



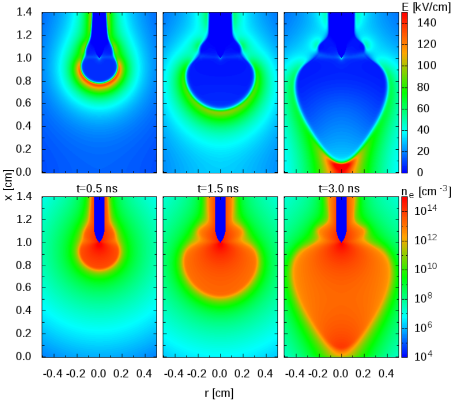
Performances of the discharge code: Test-case

- 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_{anode} **= +30 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: $\Delta t = \Delta t_{Diel} \sim 10^{-14}$s, dielectric relaxation time step Δt_{Diel} 10 times smaller than Δt_c, Δt_d, Δt_I
- Simulation time : $\sim$ **one month** with original code (memory used > 30 Go)



Performances of the discharge code: Test-case

- One time-step Δt: more than 50 % of the time for solving Poisson's equation
- Potential V + photoionization source term S_{ph}: 1+6×**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:
  - 1×10^6 points $\rightarrow$ Memory (factorization): 520 Mo $\times(1+6)=$ 3.7 Go
  - 6×10^6 points $\rightarrow$ Memory (factorization): 4 Go $\times$ (1+6) = 28 Go

Performances of the discharge code: Test-case

- One time-step Δt: 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:
  - 1×10^6 points $\rightarrow$ Memory (factorization): 520 Mo $\times$(1+6)= 3.7 Go
  - 6×10^6 points $\rightarrow$ Memory (factorization): 4 Go $\times$ (1+6) = 28 Go

Performances of the discharge code: Test-case

- One time-step Δt: 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:
  - 1×10^6 points $\rightarrow$ Memory (factorization): 520 Mo $\times$(1+6)= 3.7 Go
  - 6×10^6 points $\rightarrow$ Memory (factorization): 4 Go $\times$ (1+6) = 28 Go

Performances of the discharge code: Test-case

- One time-step Δt: 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:
  - 1×10^6 points $\rightarrow$ Memory (factorization): 520 Mo $\times$(1+6)= 3.7 Go
  - 6×10^6 points $\rightarrow$ Memory (factorization): 4 Go $\times$ (1+6) = 28 Go

Performances of the discharge code: Test-case

- One time-step Δt: 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:
  - 1×10^6 points $\rightarrow$ Memory (factorization): 520 Mo $\times$(1+6)= 3.7 Go
  - 6×10^6 points $\rightarrow$ Memory (factorization): 4 Go $\times$ (1+6) = 28 Go

Performances of the discharge code: Test-case

- One time-step Δt: more than 50 % of the time for solving Poisson's equation
- Potential V + photoionization source term S_{ph}: 1+6×**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:
  - 1×10^6 points → Memory (factorization): 520 Mo ×(1+6)= 3.7 Go
  - 6×10^6 points → Memory (factorization): 4 Go × (1+6) = 28 Go

Limitations of the initial discharge code:

- Number of points for large simulated domains
- Solution time to solve Poisson's equation
- Small time-step $\Delta t = \Delta t_{Diel} \sim 10^{-14}$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 $\Delta t = \Delta t_{Diel} \sim 10^{-14}$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)
    Pancheshnyi et al., *Journal of Computational Physics* **227**, (2008)
  - Parallel (MPI) AMR code with a hybrid particle-fluid model for the simulation of streamer discharge (2D-3D)
    Kolobov et al., *Journal of Computational Physics* **231**, (2012)
- 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

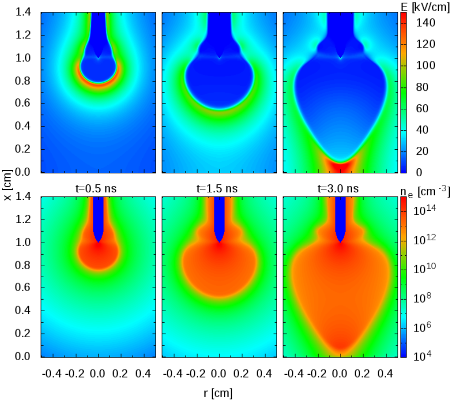
Limitations of the initial discharge code:

- 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)
    Pancheshnyi et al., *Journal of Computational Physics* **227**, (2008)
  - Parallel (MPI) AMR code with a hybrid particle-fluid model for the simulation of streamer discharge (2D-3D)
    Kolobov et al., *Journal of Computational Physics* **231**, (2012)
- Poisson's equation: implement parallel protocols (OPENMP and MPI)
- Small time-steps: improve the numerical schemes implemented

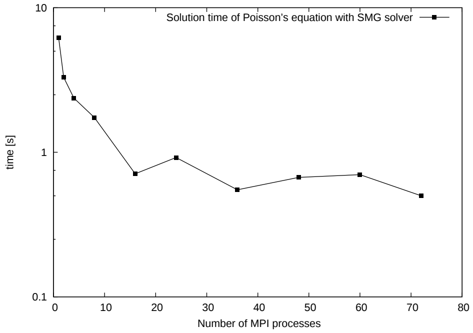
Improvements of the discharge code: Poisson's solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go
- High number of points → iterative solver becomes competitive
- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)
- Test on TC of laplacian potential:
  72 MPI pocesses:
- 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



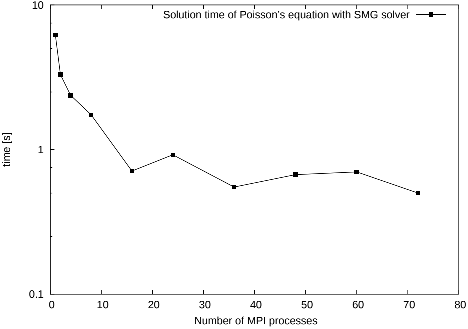
Improvements of the discharge code: Poisson's solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go
- High number of points → iterative solver becomes competitive
- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)
- Test on TC of laplacian potential: 72 MPI pocesses:
- 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



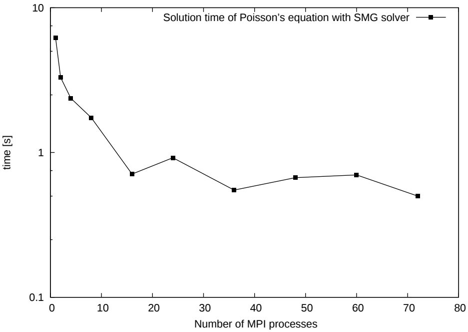
Improvements of the discharge code: Poisson's solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go
- High number of points → iterative solver becomes competitive
- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)
- Test on TC of laplacian potential: 72 MPI pocesses: 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



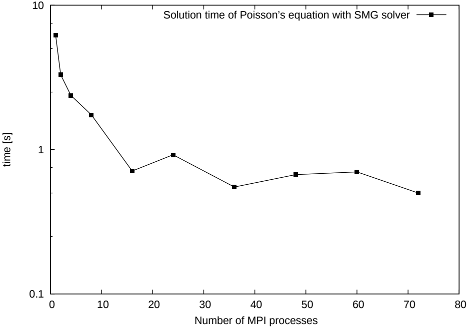
Improvements of the discharge code: Poisson's solver

- 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 pocesses: 5.7 s $\searrow$ 0.5 s
- Test on TC of laplacian potential:
  24 MPI$\times$3 OPENMP: 5.7 s $\searrow$ 0.4 s
- For iterative solver SMG, memory required is less than 1 Go



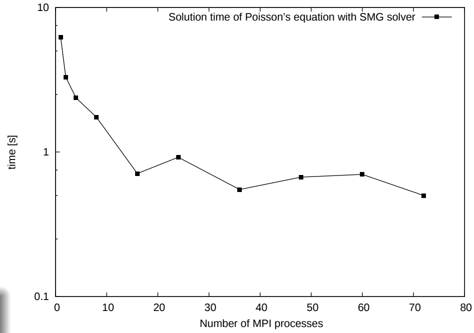
Improvements of the discharge code: Poisson's solver

- On the test-case (TC), with the MUMPS direct solver, memory required > 30 Go
- High number of points → iterative solver becomes competitive
- Implementation of the parallel MPI-OPENMP SMG solver (HYPRE library)
- Test on TC of laplacian potential:
  72 MPI pocesses: 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



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 pocesses: 5.7 s $\searrow$ 0.5 s
- Test on TC of laplacian potential:
  24 MPI$\times$3 OPENMP: 5.7 s $\searrow$ 0.4 s
- For iterative solver SMG, memory required is less than 1 Go

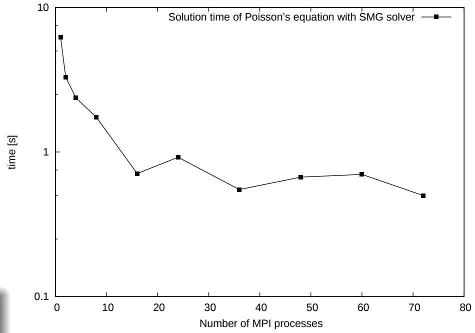


Solved problems:

- **Memory requirement**
- **Solution time to solve Poisson's equation**
- Small time-step: constraint of Δt_{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 pocesses: 5.7 s $\searrow$ 0.5 s
- Test on TC of laplacian potential:
  24 MPI$\times$3 OPENMP: 5.7 s $\searrow$ 0.4 s
- For iterative solver SMG, memory required is less than 1 Go

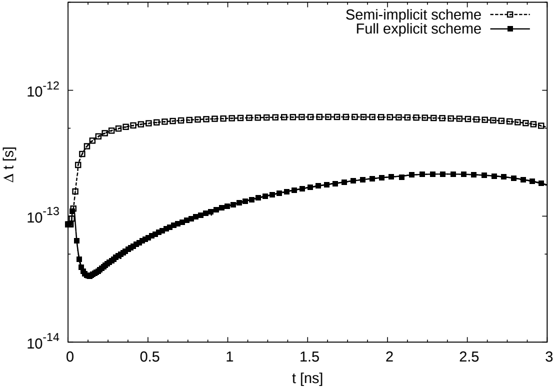


Solved problems:

- **Memory requirement**
- **Solution time to solve Poisson's equation**
- Small time-step: constraint of Δt_{Diel} ?

Improvements of the discharge code: "semi-implicit" scheme

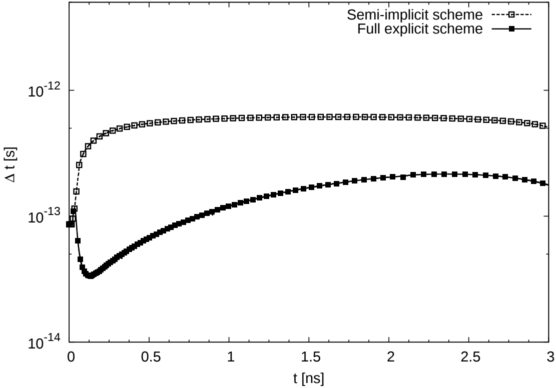
- To remove Δt_{Diel}, implementation of a "semi-implicit" scheme
  Lin et al., *Computer Physics Communications* **183**, (2012)
- On the test-case, we compare the implementation with the "semi-implicit" scheme with the full explicit model:



- We can choose a time-step 10 bigger than with the explicit model

Improvements of the discharge code: "semi-implicit" scheme

- To remove Δt_{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:



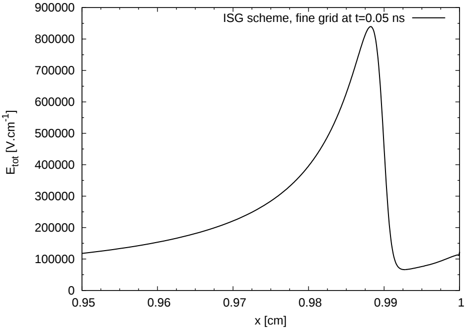
- We can choose a time-step 10 bigger than with the explicit model

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 Δx=1 μm,Δr=1 μm
  - Test-case 2: E_{axis} profile with close to the point Δx=2.5 μm,Δr=1 μm
- The UNO3 scheme, explicit 3^{rd} order accurate + explicit 2^{rd} order for the diffusion

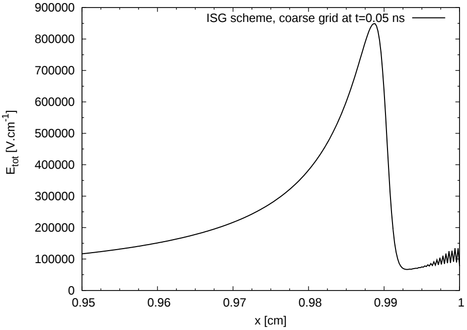
  Li et al., *Monthly Weather Review* **136**, (2008)



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 Δx=1 μm,Δr=1 μm
  - Test-case 2: E_{axis} profile with close to the point Δx=2.5 μm,Δr=1 μm
- The UNO3 scheme, explicit 3rd order accurate + explicit 2rd order for the diffusion
  Li et al., *Monthly Weather Review* **136**, (2008)

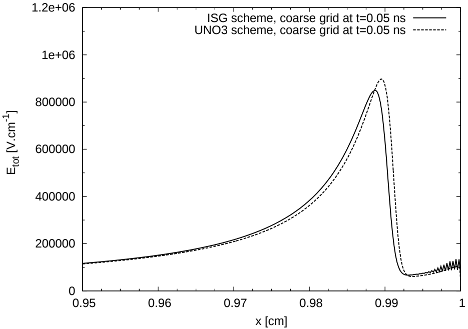


Improvements of the discharge code: UNO3 convection scheme

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^{rd} order for the diffusion

  Li et al., *Monthly Weather Review* **136**, (2008)

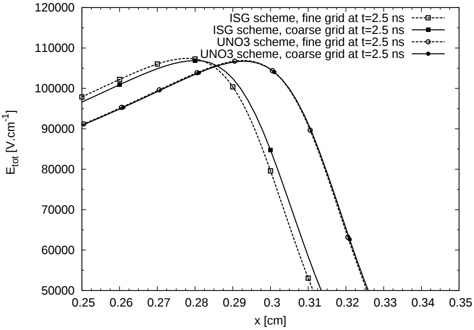


Improvements of the discharge code: UNO3 convection scheme

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 Δx=1 μm,Δr=1 μm
  - Test-case 2: E_{axis} profile with close to the point Δx=2.5 μm,Δr=1 μm
- The UNO3 scheme, explicit 3rd order accurate + explicit 2rd order for the diffusion

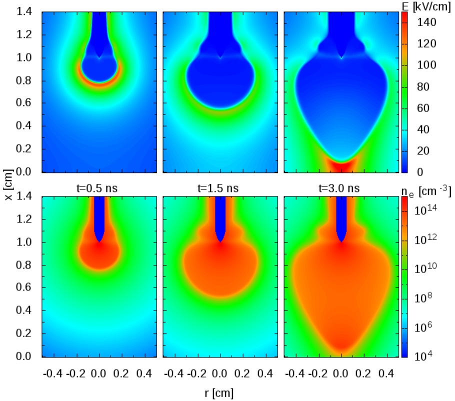
  Li et al., *Monthly Weather Review* **136**, (2008)



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 Δt_{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$\times$3 OPENMP: 17.05 s $\searrow$ 0.86 s
- TC is computed in $\sim$ **3 hours**

  (**one month** with initial code)

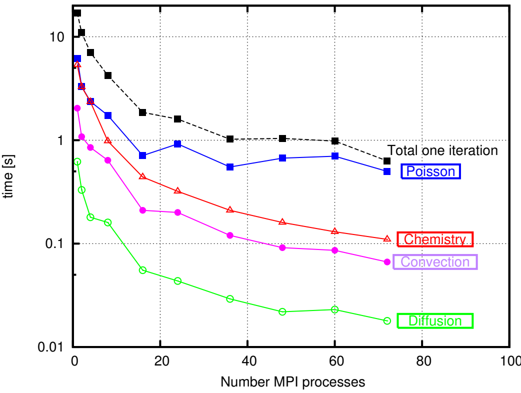


Improvements of the discharge code: MPI-OPENMP discharge code

- **Full parallel MPI-OPENMP discharge code**
- Poisson's equation: MPI-OPENMP iterative solver SMG
- Small time-steps: Semi-implicit scheme (to remove Δt_{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×3 OPENMP: 17.05 s ↘ 0.86 s
- TC is computed in ∼ **3 hours**

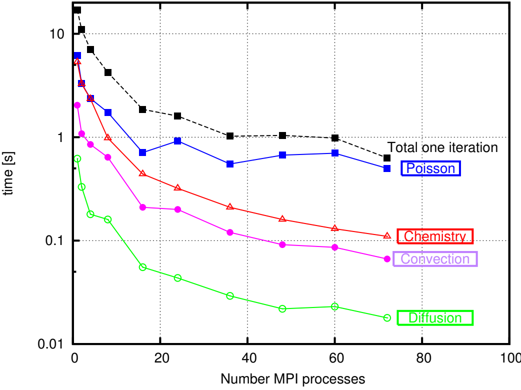
  (**one month** with initial code)



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 Δt_{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 pocesses: 17.05 s ↘ 0.63 s
- Test on TC one time-step:
  24 MPI×3 OPENMP: 17.05 s ↘ 0.86 s
- TC is computed in ∼ **3 hours**
  (**one month** with initial code)

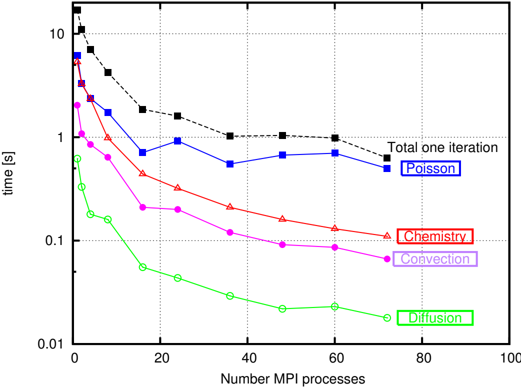


Improvements of the discharge code: MPI-OPENMP discharge code

- **Full parallel MPI-OPENMP discharge code**
- Poisson's equation: MPI-OPENMP iterative solver SMG
- Small time-steps: Semi-implicit scheme (to remove Δt_{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 pocesses: 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)

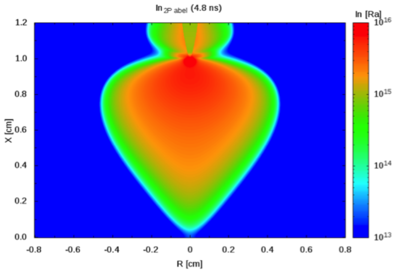


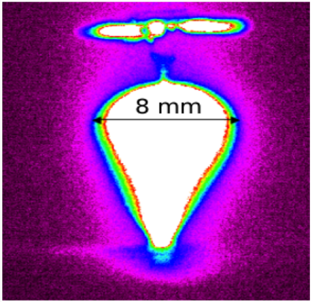
Comparison of results with experiments

Numerical/Experimental comparison

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

Good agreement with experiments





Pierre Le Delliou PhD. Thesis.

Table of contents

Table of contents

Conclusions

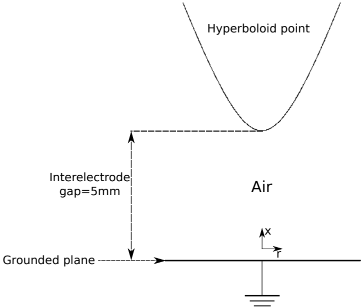
- For plasma applications, we need of course fast, robust and parallel iterative solvers (HYPRE is used for now)
- Even on 2D structured grids the computation can be very intensive
- Next step is mostly to use AMR meshes as well as improving the physical model (hybrid models)
- Currently in a joint project 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

Thanks

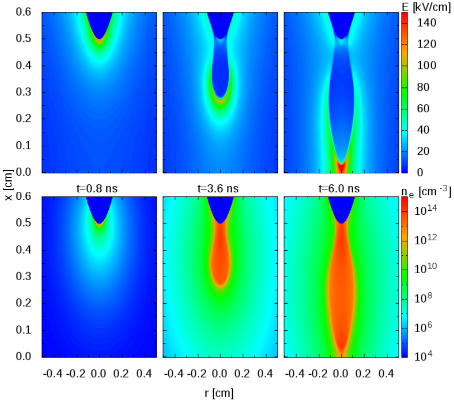
Thank you for your attention.

Improvements of the discharge code: Test-case 1

- 5 mm gap with a point to plane geometry with a stump point electrode
- Constant voltage applied at the anode, V_{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 $\times$ 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

Code with:	MUMPS	PaStiX			
Memory:	664 Mo	886 Mo			
Nb thread	1	1	2	4	6
Factorization (s)	64.12	24.87	21.55	19.91	19.48
Solution (s)	1.27	0.64	0.38	0.25	0.22
One time-step (s)	5.76	2.71	1.76	1.51	1.36

- With 6 threads:
  - Speed up Poisson's equation: 5.77
  - Speed up one time step: 4.2
- Computational time for TC-1: 4 hours $\rightarrow \sim$ 40 minutes