Low pressure gas discharges and a simple one dimensional particle in cell code

WJMB: Date: 2004/02/11 Revision: 1.18

1 Low Pressure Gas Discharges

Historically a fairly well studied type of gas discharge is the one that is created in the DC electric field from two metal electrodes. Quite a lot can be learned from it if one considers its current-voltage characteristic over a wide range of currents. Such a characteristic is shown for a low pressure (less then a few mm Hg or a few 100 Pa) discharge in figure 1. Several regions can be identified:

photo electric currents (a − b): due to cosmic and other sources of radiation erratic electrons are created inside the discharge region and at the electrode surfaces. These electrons give rise to a tiny current when a voltage is applied. No significant ionisation takes place and the fact that the discharge relies on external sources of inionisation means that it is not self-sustained.

Townsend discharge (b − c): at a certain voltage enough ions are created to drift towards the cathode and release electrons in secondary emission processes. The electrons in turn create new avalanches of ionisation and the discharge can now sustain itself this way. Even the slightest increase in this voltage (called the breakdown voltage) would result in a larger current and therefore this part of the iv-characteristic is practically flat. Space charges remain to be negligible in this region. Since hardly any light is emitted from the discharge it is often called 'dark dischare'.

subnormal glow discharge (c − d): due to the formation of a positive space charge, the potential drop is concentrated more and more towards the cathode. The decrease in voltage across the discharge at increasing current shows that ionisation becomes more effective because of this 'cathode fall'. The current is no longer only determined by the external resistor, but by the discharge as well.

glow discharge (d − e): when the current is allowed to increase further (by e.g. changing the value of the resistor) the voltage across the discharge stays constant. More current can flow due to an extension of the

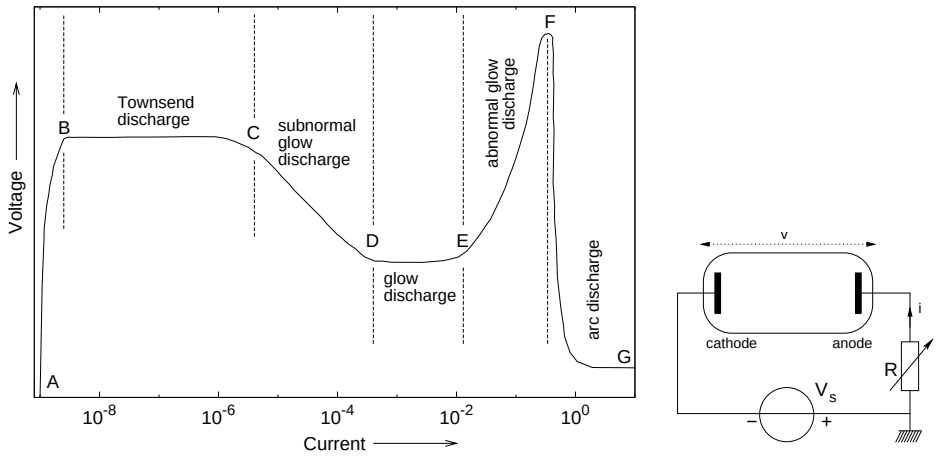

Figure 1: the current-voltage characteristics [1] of a DC discharge. Note that the range for the current is merely illustrative.

discharge over the electrode surface at constant current density. Depending on gas and geometry the light emitted by the discarge shows a fair amount of spatial structure.

above-normal glow discharge (e − f): the entire surface of the electrode is now covered by the discharge and the only way in which the current can be increased is by increasing the cathode fall, so that secondary electrons gain more energy and cause more ionisation.

arc discharge (f − g): at even larger current the discharge voltage drops again and an arc discharge establishes. Entirely different processes like gas heating, radiative cooling and constriction start playing a role in this regime.

In the glow discharge region C − F the discharge has a rather strong will of its own: it organises itself such that ionisation is most effective and/or losses are minimal. This behaviour reflects in the different luminous regions one can recognise in a glow discharge in a long tube, as schematically drawn in figure 2. The discharge establishes this structure via a close interaction between its charged particles and the electric field. In the next section a one dimensional numerical code will be described which can show how this interaction takes place and how it depends on quantities like the background pressure and applied voltage.

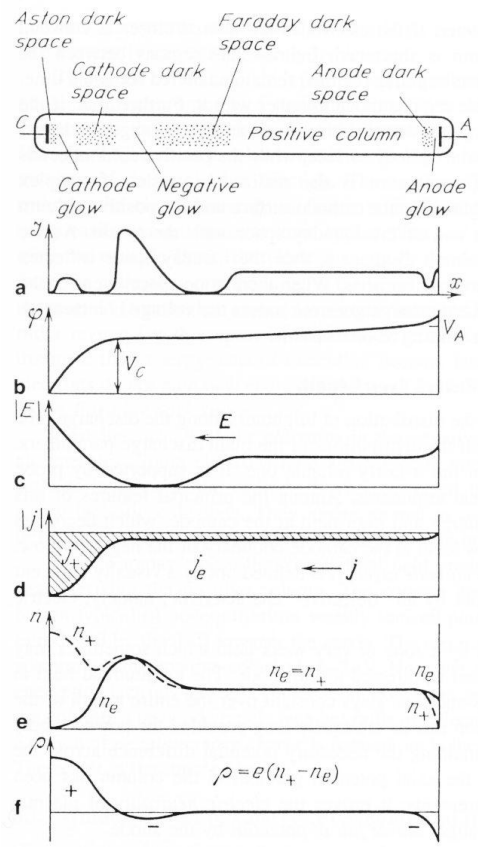

Figure 2: the structure of a glow discharge (taken from reference [2]). Note the cathode fall V_C in plot b, the electron and ion densities in plot e and the resulting space charge density ρ in plot f.

2 Particle in Cell code

This section describes briefly the code contained in the four files `argon.h`, `fluid.h`, `monte_carlo.h` and `pic_example.cpp`. With this code it is possible to model various aspects (those that one can capture in one dimension) of the glow discharge treated in section 1.

The first two files provide the basic ingredients to make various plasma models: `fluid.h` defines a one-dimensional grid and implements the so called ϕ-equation. With the classes provided in this file one can construct fluid models of discharges. `monte_carlo.h` defines the concepts necessary to create a particle model: particles, binary collisions, etc. Furthermore, `argon.h` implements some cross section data for electrons and argon ions colliding with neutral argon atoms. Finally `pic_example.cpp` uses these previous three files to actually construct a simple exemplary Particle in Cell

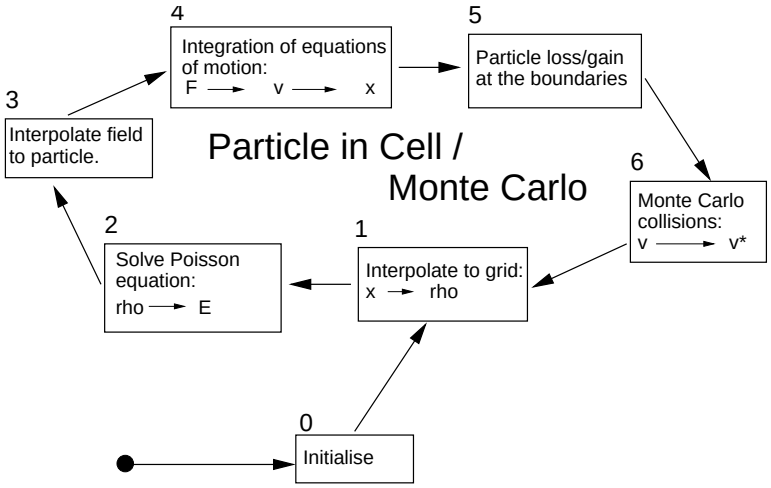

Figure 3: PIC/MC scheme.

model [3].

Figure 3 gives a schematic representation of the algorithm implemented in `pic_example.cpp`. By reading through the code in this file one should be able to recognise this scheme and to make minor modifications to it.

2.1 The Poisson equation

In `fluid.h` the basic functionality is implemented to define a one-dimensional grid and solve the general transport equation — the ϕ-equation — on this grid. The notation used is the one of Patankar [4]:

$$\frac{\partial \rho\phi}{\partial t} + \vec{\nabla} \cdot (\rho\vec{u}\phi) = \vec{\nabla} \cdot (\Gamma_\phi \vec{\nabla}\phi) + S_\phi \tag{1}$$

One recognises the density ρ, the transported quantity ϕ, the time t, the velocity $\vec{u}$, the general diffusion coefficients Γ_ϕ, and the source S_ϕ.

By removing several terms, the ϕ-equation can be reduced to the Poisson equation:

$$\vec{\nabla} \cdot \varepsilon_0 \vec{\nabla}\varphi = -(q_e n_e + q_i n_i) \tag{2}$$

and therefore the code in `fluid.h` can be used to actually solve the electrostatic ($\frac{d}{dt} = 0$) Poisson equation. For this we define a potential field φ without flow ($\vec{u} = 0$). The diffusion coefficient is taken to be equal to the permittivity of free space: $\Gamma_\phi = \varepsilon_0$. The source term of the remaining equation is the charge density $(q_e n_e + q_i n_i)$, due to the electrons e and the ions i.

In order to solve the equation, boundary conditions are necessary. For a one-dimensional discharge bounded by two electrodes, these are just the applied voltages on the electrodes: the have homogeneous Dirichlet boundary

conditions:

$$\begin{aligned} \varphi|_{\text{left}} &= -V_{\text{applied}} \\ \varphi|_{\text{right}} &= 0 \end{aligned} \tag{3}$$

2.2 Particles and Kinetics

The file `monte_carlo.h` implements different concepts like *particle*, *swarm*, *process* and *random number generator*.

A **particle** is something which has intrinsic properties like mass and charge, and extrinsic parameters like position and velocity. In order to keep the amount of particles limited[1], one simulation particle does not represent one real particle (like an electron or an ion), but a fair number of them: this is called the weight of the simulation particle. The particle class implemented in `monte_carlo.h` has several member functions that say something about or modify the particle: e.g. the `Move(double Efield, double dt)` function changes the position of the particle and its velocity in the following way:

$$\begin{aligned} x_{\text{new}} &= x_{\text{old}} + dt\left(v_x + \frac{qE}{m}\frac{dt}{2}\right) \\ v_{x,\text{new}} &= v_{x,\text{old}} + \frac{qE}{m}dt \end{aligned} \tag{4}$$

with E the electric field, dt the time step, q the charge of the particle and m the mass of the particle.

Particles with the same intrinsic properties are collected in a **swarm**. This is a convenient way of organising things since this allows one to have functions which act upon e.g. all the electrons or read information from them. A swarm has member functions like `AddParticle()`, which adds a particle to the list, `EDF()`, which calculates the energy distribution function of the particles in the swarm, `RemoveInactiveParticles()`, which removes particles that have left the discharge region, and `CalcRho()`, which calculates the charge density due to all the particles in the swarm.

Since particles undergo collisions with the background gas, a so-called **process** is defined. The process keeps the data about the cross section of the reaction and it can calculate the result of a collision between two particles. Two types of processes are found in `monte_carlo.h`: a hard-sphere process which can be used to simulate both elastic and inelastic collisions and a charge exchange process which is typically something which happens only to ions.

2.3 Reactions

In the file `argon.h` various functions are present which implement data concerning cross sections (necessary for the processes in the previous section)

[1] present day computer power and storage capacity forbids following more simulation particles than about 10^7 on a desktop PC.

Table 1: A list of reactions (see figure 4 for the cross sections).

Nr.	Reaction	Threshold	Ref.
0	$Ar + e \rightarrow Ar + e$		[5]
1	$Ar + e \rightarrow Ar^* + e$	11.5 eV	[5]
2	$Ar + e \rightarrow Ar^+ + 2e$	15.8 eV	[5]
3	$Ar^+ + Ar \rightarrow Ar + Ar^+$		
4	$Ar^+ + Ar \rightarrow Ar^+ + Ar$		

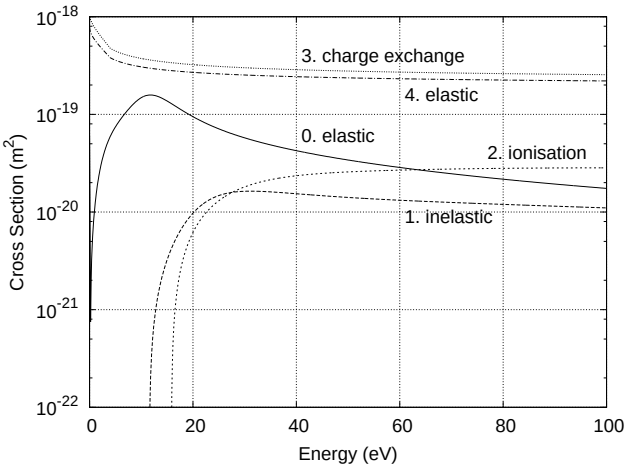

Figure 4: Cross sections of the reactions listed in table 1.

for argon. In table 1 the various reaction equations are listed: three reactions (0–2) in which electrons collide with atoms and two reactions (3–4) in which ions collide with atoms. Note the reaction number 1 really represents all possible excitation reactions by 'lumping' the different excited states together in one effective excited state. Figure 4 shows the cross sections for these reactions.

References

[1] F.M. Penning. *Electrical Discharges in Gases.* Philips' Technical Library, 1957.

[2] Yu. P. Raizer. *Gas Discharge Physics.* Springer, Berlin, 1991.

[3] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation.* Adam Hilger, 1991.

[4] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing Corporation, 1980.

[5] A. V. Phelps and Z. Lj. Petrović. Review article: Cold-cathode discharges and breakdown in argon: surface and gas phase production of secondary electrons. *Plasma Sources Sci. Technol.*, 8:R21–R44, 1999.