

Particle-In-Cell/Monte-Carlo Modeling of Collisional and Non-Collisional Plasmas

G. N. Marques
A. J. Preto
S. Stephany

*Laboratório Associado de Computação e
Matemática Aplicada – LAC/INPE, Brasil
[gleber,airam,stephan]@lac.inpe.br*

A. Passaro
N. M. Abe
A.C.J.Paes

*Laboratório de Engenharia Virtual –
IEAv/CTA, Brasil
[angelo,nancy,acjpaes]@ieav.cta.br*

Abstract

This paper presents an approach to model collisional and non-collisional plasmas using particles. The behavior of the plasma is governed by the Boltzmann equation. The particle evolution in the phase space is approximated by an expansion for the phase space distribution function, in which the collisional and non-collisional terms derived from Boltzmann equation can be treated separately. An electrostatic Particle-In-Cell model solves the non-collisional motion using the Lorentz and the Newton equations. The Finite Element Method is employed to compute the electric field using the Poisson equation. In this approach, only elastic and Coulomb collisions are considered, modeled by a Monte Carlo method.

Keywords: PIC, PIC/MC, plasma simulation, Coulomb collision, elastic collision

1. Introduction

Recent developments of plasma-based technologies can take advantage of the simulation of different plasma types in several operating conditions. Some interesting topics of plasmas simulation research can be found in [4]-[8],[12],[14],[16],[17],[20],[21]. It is important to make a proper choice of the model for simulating a specific experiment setup, which depends not only on the aim of the investigation, but mainly on the plasma characteristics and source interactions - e.g. the plasma density, plasma-surface interactions, laser or discharge plasma interactions, as well as the electronic affinity of the chemical species of the plasma. For instance, discrepancies of more than 70% (in eV^{-1}) were obtained when comparing the electron energy distribution function (EEDF) in three different simulation models of an inductively coupled argon plasma experiment [18]. In the first case, no Coulomb collision was treated, while in the second case only electron-electron Coulomb collision was considered.

Finally, in the more accurate model, electron-electron and electron-ion Coulomb collisions were taken into account. In addition, charge-neutralizing collisions of positive and negative ions may take place, for instance, in the simulation of electronegative plasmas. This is the case of coupled RF discharge in electronegative gases, where processes of loss of negative ions are very important and charge-neutralizing collisions are the dominant process. The complexity of the computational model depends on the physical and/or chemical processes that are included in order to describe properly the behavior of the specific plasma.

Besides the interdisciplinary of these topics on plasmas, one must consider the high diversity of particle simulation methodologies, which have been developed separately for some decades. Two main streams can be distinguished: the plasma physicists, following Birdsall [2] (PIC); and the aerodynamicists, following Bird's approach [3] (DSMC). This scenario results in a large, but fragmented amount of publications. As already discussed by many authors [9]-[11],[18], the knowledge exchange between these areas provides an important effort to the development of Computational Fluid Dynamics (CFD) capabilities mainly for those working in the limits of fluid-kinetic descriptions. In this sense, this article contributes to those who want to extend the capability of an electrostatic PIC model towards a PIC/MC one, in order to model additional collisional processes.

2. Boltzmann equation in particle simulation

As an example, consider a weakly ionized plasma consisting of electrons (e), ions (A) and molecules (B) for which the Coulomb collisions between ions and electrons can be disregarded, thus we must take into account only e - B , A - B and B - B collisions. The particle system evolution is governed by the Boltzmann equations associated with each particle species, which,

under these assumptions, are linear for the electrons and ions and nonlinear for the molecules because of the term related to collisions between like particles. Thus, the Boltzmann equation for ions A can be written as:

$$\frac{\partial f_A}{\partial t} + v_j \frac{\partial f_A}{\partial x_j} + \frac{1}{m_A} \frac{\partial}{\partial v_j} (F_j f_A) = \iint [f_A(\mathbf{v}') f_B(\mathbf{w}') - f_A(\mathbf{v}) f_B(\mathbf{w})] g \sigma^{AB} d\Omega d\mathbf{w}, \quad (1)$$

where, summation over suffix j is implied, F_j is the force acting on particle A with a mass m_A , $f_A(\mathbf{v})$ and $f_B(\mathbf{w})$ represent the velocity distribution functions, $f_A(\mathbf{x}, \mathbf{v}, t)$, of species A and B , \mathbf{v}' and \mathbf{w}' are the post-collision velocities, $g = |\mathbf{v} - \mathbf{w}|$ is the relative speed, $d\Omega = \sin \chi d\chi d\psi$ is the solid angle, σ^{AB} is the differential cross section that is function of the relative speed g and of the deflection angle χ .

The force \mathbf{F} is given by the Lorentz force equation:

$$\mathbf{F} = q_A (\mathbf{E} + \mathbf{v} \times \mathbf{B}), \quad (2)$$

where q_A is the charge of ion A , and \mathbf{E} and \mathbf{B} are the electric and magnetic fields, respectively.

Expanding $f_A(\mathbf{v}, \mathbf{x}, t)$ for a small Δt we can write:

$$f_A(\mathbf{v}, \mathbf{x}, t + \Delta t) = f_A(\mathbf{v}, \mathbf{x}, t) + \Delta t \left(\frac{\partial f_A}{\partial t} \right)_t = (1 + \Delta t \cdot J)(1 + \Delta t \cdot D) f_A(\mathbf{v}, \mathbf{x}, t), \quad (3)$$

after substitution of (1) into (3) and having defined the non-collisional and collisional operators D and J , respectively, by:

$$D(f_A) = -v_j \frac{\partial f_A}{\partial x_j} - \frac{1}{m_A} \frac{\partial}{\partial v_j} (F_j f_A), \quad (4)$$

$$J(f_A) = \iint [f_A(\mathbf{v}') f_B(\mathbf{w}') - f_A(\mathbf{v}) f_B(\mathbf{w})] g \sigma^{AB} d\Omega d\mathbf{w}.$$

Equation (3) represents a decoupling of the non-collisional and collisional parts of the particle motion, its validity is well discussed in the work of Nanbu [18], but essentially the time increment must be much smaller than the mean free time τ . Moreover, equation (3) suggests a two step scheme for solving the Boltzmann equation.

In the first step, using the initial velocity distribution function $f_A(\mathbf{v})$, we obtain the collisionless part of the solution $f_A^* \equiv (1 + \Delta t \cdot D) f_A(\mathbf{v}, \mathbf{x}, t)$. In fact, we need to solve the non-collisional Boltzmann equation, also called Vlasov equation,

$$\frac{\partial f_A}{\partial t} + v_j \frac{\partial f_A}{\partial x_j} + \frac{1}{m_A} \frac{\partial}{\partial v_j} (F_j f_A) = 0. \quad (5)$$

The procedure used to solve this equation in PIC models is to advance particles in time accordingly to Newton's motion law, in which the force is given by the Lorentz relation (2) and the electric (and magnetic) field is given by the set of Maxwell equations on an adequately spaced mesh covering the entire domain.

In the second step we must calculate the collisional part of the motion, $(1 + \Delta t \cdot J) f_A^*$, we must solve

$$\frac{\partial f_A}{\partial t} = \iint [f_A(\mathbf{v}') f_B(\mathbf{w}') - f_A(\mathbf{v}) f_B(\mathbf{w})] g \sigma^{AB} d\Omega d\mathbf{w}, \quad (6)$$

by using the intermediate initial distribution function f_A^* . This equation represents the collisional relaxation of the velocity distribution in a spatially uniform state of species A , and it can be applied only to small cells with linear dimensions of the order of $\min\{\lambda_{mA}, \lambda_D\}$,

where λ_D is the Debye length and λ_{mA} is the mean free path of species A , since the applicability of (6) depends on the assumptions of uniform gradient density, temperature, and flow velocity of species A inside the cell. It can be mentioned that for properly resolving plasma sheaths, the linear dimensions of the cells must be only a fraction of the Debye length, since strong gradients of electron and ion densities occur around these regions whose thickness are of the order of the Debye length.

The ratio of the number of real particle to that of simulated particles is called weight. The definition of weights is very convenient for many reasons. Firstly, it enables the simulation of larger particle systems since it strongly reduces the number of simulated particles, and secondly, it provides a way to control, or keep approximately equal, the statistical fluctuations expected in each cell since the number of simulated particles per cell can be approximately equalized by properly adjusting the particle weighting factors [9]. Note that since weighting is a multiplicative factor for both mass and charge, the PIC formulation remains unchanged for any weighting scheme because the charge-mass ratio does not change. However, collisional modeling require a more careful analysis concerning to the definition of super-particles [15],[18]. We adopt the terminology "simulated particle" instead of "super-particle".

In order to solve equation (6), we have found the phase space distribution of the particles, f_A^* ,

$$f_A^*(\mathbf{v}, \mathbf{x}, t) = \frac{n_A}{N_A} \sum_{i=1}^{N_A} \delta^3(\mathbf{v} - \mathbf{v}_{Ai}), \quad (7)$$

where, $\delta^3(\bullet)$ is the Dirac delta function, n_A is the number density of species A for the cell, and \mathbf{v}_{Ai} is the collisionless advanced velocity before calculated. An analogous expression can be found for species B by changing the suffixes A by B and possibly i by j . By using these probability density expressions, in which the probabilities are introduced by the delta functions, and after algebraic manipulation of

$$f_A(\mathbf{v}, \mathbf{x}, t + \Delta t) = (1 + \Delta t \cdot J) f_A^*, \quad (8)$$

where the operator J is defined in (4), the complete solution, that is, the gathering of the collisionless and collisional motion can be expressed as

$$f_A(\mathbf{v}, \mathbf{x}, t + \Delta t) = \frac{n_A}{N_A} \sum_{i=1}^{N_A} f_{Ai}(\mathbf{v}) \quad (9)$$

where $f_{Ai}(\mathbf{v})$ is the probability density function for the velocity of particle A_i at $t + \Delta t$, and it is given by:

$$f_{Ai}(\mathbf{v}) = (1 - P_{Ai}) \delta^3(\mathbf{v} - \mathbf{v}_{Ai}) + P_{Ai} Q_{Ai}(\mathbf{v}). \quad (10)$$

In the equation above, P_{Ai} is the collision probability of particle A_i , given by

$$P_{Ai} = \sum_{j=1}^{N_B} P_{Ai, Bj}, \quad (11)$$

where $P_{Ai, Bj}$ is the probability of the collision between particle A_i and B_j in time Δt , written as

$$P_{Ai, Bj} = \frac{n_B g_{Ai, Bj} \sigma_T^{AB}(g_{Ai, Bj}) \Delta t}{N_B}, \quad (12)$$

and $Q_{Ai}(\mathbf{v})$ is the probability density function for the post collision velocity of particle A_i , and is given by

$$Q_{Ai}(\mathbf{v}) = \sum_{j=1}^{N_B} \frac{P_{Ai, Bj}}{P_{Ai}} Q_{Ai, Bj}(\mathbf{v}) \quad (13)$$

where the two terms of the product in the summation can be interpreted as the conditional probability of A_i - B_j pair collision given that A_i collides, $P_{Ai, Bj}/P_{Ai}$, and the other as the density function for the postcollision velocity of particle A_i when its collision partner is particle B_j , $Q_{Ai, Bj}$, which can be written as

$$Q_{Ai, Bj}(\mathbf{v}) = \frac{\sigma^{AB}(g_{Ai, Bj}, \chi) \cdot \delta(\|\mathbf{g}'\| - g_{Ai, Bj})}{(g_{Ai, Bj})^2 \cdot M_B^3 \cdot \sigma_T^{AB}(g_{Ai, Bj})}, \quad (14)$$

having defined $M_A = m_A / (m_A + m_B)$, $M_B = m_B / (m_A + m_B)$,

$$\mathbf{g}' = \frac{1}{M_B} (\mathbf{v} - M_A \mathbf{v}_{Ai} - M_B \mathbf{v}_{Bj}), \quad (15)$$

the deflection angle χ as the angle between \mathbf{g}' and $\mathbf{v}_{Ai} - \mathbf{v}_{Bj}$, and the differential cross section $\sigma^{AB}(g_{Ai, Bj}, \chi)$. Note that if $P_{Ai} = 0$ in equation (10), no collisional event takes place and, in this case, the velocity of particle A_i remain unchanged at the end of the collisional stage of the simulation procedure. More detailed presentation of the collisionless and collisional operators derived from the Boltzmann equation, also called Nambu's method, can be found in [18], and a formal convergence proof in [1].

3. The use of FEM in PIC simulations

The development of our research on plasma simulations has begun on the basis of the PIC methodology. An electrostatic PIC model solves the Vlasov equation by indirectly solving the Newton's motion equations for which the dominant force acting on the charged particles is given by the Lorentz force equation (2), therein using the Poisson's equation to determine the electric field. Once the force acting on

the particles, as well as its acceleration, is known, Newton's equation can be solved.

In PIC models the simulation domain is decomposed by a set of regular and non-intersecting geometrical elements whose linear dimensions are of the order of the Debye length; these geometrical elements compose a mesh over the entire domain and are called cells in the PIC methodology, while from a FEM point-of-view they are the so called finite elements. Although the linear dimensions of the cells are subjected to the Debye length, hereafter we make no distinction in the use of the terminologies cells and finite elements, since in a PIC-FEM approach, the FEM mesh must be sufficiently refined for fairly describe the plasma, such that is natural that the cells and the finite elements coincide [19].

We start from a set of initial conditions for the particle species of the plasma: the positions \mathbf{x}_i^0 , the velocities \mathbf{v}_i^0 , the charges q_i , and the mass m_i , $i=1..N$. The PIC simulation cycle consists of four stages: 1) charge distribution; 2) field resolution; 3) field interpolation, and the 4) particle advance stage. At the end of stage 4), the electrostatic collisionless plasma simulation model has been advanced by Δt in time.

A special feature of the PIC-FEM approach is that, since the finite element shape functions satisfy the partition of unit condition [13], these functions can be used in the charge assignment in stage 1) where the particles inside each cell must be partitioned among the cell vertexes. Since this partitioning scheme must preserve the total particle's charge, the partition of unit criteria must hold for the set of functions used to perform this task in the cells. A well detailed discussion can be found in [19]. After this, one has obtained an approximated charge distribution for the charge density function values at the cell vertexes. In stage 2) we use the scalar potential formulation for the electric field, $\mathbf{E} = -\nabla\Phi$, reducing the set of Maxwell equations to the Poisson equation, which in turn is solved by the FEM for the state variable Φ . The electric field interpolation is straight-forward in FEM formulations, and it is written as

$$\mathbf{E}(\mathbf{x}_p) = -\nabla\Phi \cong -\sum_i \phi_i \nabla N_i(\mathbf{x}_p) \quad (16)$$

where ϕ_i 's are the state variable values (electric potential) at the nodes i of the cell (e.g., $i=1..3$ in the case of triangular elements), and ∇N_i 's are the gradient of the finite element shape functions at the particle position \mathbf{x}_p . Thereafter, we can compute the acceleration of each particle p as

$$\mathbf{a}_p = q_p / m_p \mathbf{E}(\mathbf{x}_p), \quad (17)$$

and finally, by using an integrative method for computing the new position and velocity, as for example, the following leap frog scheme,

$$\mathbf{v}^{(n+\frac{1}{2})} = \mathbf{v}^{(n-\frac{1}{2})} + \mathbf{a}^{(n)} \Delta t \quad (18)$$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \mathbf{v}^{(n+\frac{1}{2})} \Delta t, \quad (19)$$

stage 4) is completed and the particle system has been advanced in time by Δt . The latter can be taken sufficiently small in function of the electron-plasma frequency, for instance, $\Delta t = 0.2\omega_{pe}^{-1}$.

4. Modeling elastic and Coulomb collisions in PIC/MC simulations

While in the DSMC simulation the way of thinking is “when will the next collisional event probably occur?”, in the PIC/MC frame we ask for “what are the relevant collisions that probably have occurred in Δt ?”. Thus, after performing the collisionless part of particle motion, a collisional procedure is performed in order to introduce the collisional scatterings due to collisional events that may have taken place in Δt . In next subsections we shortly outline two algorithms, which resolve specific instances of the Boltzmann equation: one for simulating Coulomb collisions (ion-ion, electron-ion, electron-electron) and the other for modeling elastic collision between heavy particles.

4.1 Collisional algorithm for a gas mixture

For simplicity, let us consider a binary mixture of species A and B ; the extending to more component mixtures is straight. Let us define A as being a molecule and B as an ion or molecule. One must keep in mind that when B is chosen to be an ion, the B - B collisions are Coulomb collisions and must be treated separately. In time, this algorithm is based on the variable hard sphere (VHS) model, thus adequate for modeling heavy particle interactions. Let us consider A - B collisions, since collisions between like particles are a particular case in this modeling. Let N_A and N_B be the number of simulated particles inside a cell, and W_A and W_B be the weight of species A and B , respectively. This weighting scheme is referred as species dependent weighting; the simpler scheme is the equally weighting scheme ($W_A = W_B = W$) and the more general one is the particle dependent weighting, in which particles of the same species may have different weighting. Since to each simulated particle can be assigned a different weight, it is natural to think of a dynamic weighting scheme since it may be powerful for controlling the statistical fluctuation in the cells due to the random character of the collisional models. One can expect the degree of complexity involved in such a modeling. Indeed, recent efforts have been published in the Literature exploring these advanced topics in particle simulation [9],[11]. The probability that molecule A_i collide with some particle B_j can be expressed as

$$P_{Ai} = n_B g \sigma_T^{AB} \Delta t = \sum_{j=1}^{N_B} \left(\frac{n_B g_{Ai,Bj} \sigma_T^{AB} \Delta t}{N_B} \right) = \sum_{j=1}^{N_B} P_{Ai,Bj}, \quad (20)$$

where $n_B = W_B N_B V_c^{-1}$ is the number density of species B and V_c^{-1} is the cell volume. Also, $P_{Ai,Bj}$ represents the probability that molecule A_i collides with particle B_j .

Using the idea of maximum collision number, one can estimate [17],[18] a maximum relative speed g_{\max}^{AB} in the cell for all the possible collision partners B_j , which is used to obtain a maximum probability that works as an upper bound for all the collision probabilities of the pairs $\{(A_i B_j): j=1..N_B\}$ for a given molecule i . From these considerations it can be shown [18] that the maximum number of collisions between pairs $\{(A_i - B_j): i=1..N_A, j=1..N_B\}$, can be written as

$$N_{\max}^{AB} = n_B N_A g_{\max}^{AB} \sigma_T^{AB} (g_{\max}^{AB}) \Delta t, \quad (21)$$

in the case of $W_A = W_B$, and as

$$N_{\max}^{AB} = \max\{N_A, N_B\} \cdot (N_A N_B V_c^{-1} g_{\max}^{AB} \sigma_T^{AB} (g_{\max}^{AB}) \Delta t), \quad (22)$$

when using a species dependent weighting scheme. Similarly, the maximum collision number for collisions between like particles is:

$$N_{\max}^{SS} = (n_S (N_S - 1) g_{\max}^{SS} \sigma_T^{SS} (g_{\max}^{SS}) \Delta t) / 2. \quad (23)$$

Once we have defined

$$q_{ij}^{s,r} = \frac{g_{s_i, r_j} \sigma_T^{r,s} (g_{s_i, r_j})}{g_{\max}^{s,r} \sigma_T (g_{\max}^{sr})}, \quad (24)$$

the algorithm is given as follows, for each cell:

1. For $k_{AA} = 1..N_{\max}^{AA}$ do
 - a. Obtain a random pair $(A_i A_j)$ and compute $q_{ij}^{A,A}$;
 - b. Using a random number $U \sim U[0,1]$, compare: if $U > q_{ij}^{A,A}$ the pair does not collide, otherwise, the collision occur and the post-collision velocities are: $\mathbf{v}'_i = \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_j + g_{ij} \mathbf{R})$ and $\mathbf{v}'_j = \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_j - g_{ij} \mathbf{R})$, where \mathbf{R} is a unit random vector.
2. For $k_{BB} = 1..N_{\max}^{BB}$ do
 - a. Obtain a random pair $(B_i B_j)$ and compute $q_{ij}^{B,B}$;
 - b. Using a random number $U \sim U[0,1]$, compare: if $U > q_{ij}^{B,B}$ the pair does not collide, otherwise, the collision occurs and the post-collision velocities are: $\mathbf{v}'_i = \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_j + g_{ij} \mathbf{R})$ and $\mathbf{v}'_j = \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_j - g_{ij} \mathbf{R})$, where \mathbf{R} is a unit random vector.
3. For $k_{AB} = 1..N_{\max}^{AB}$
 - a. Choose a random pair $(A_i B_j)$ and compute q_{ij}^{AB} ;
 - b. Using a random number $U \sim U[0,1]$, compare: if $U > q_{ij}^{AB}$ then the pair does not collide, else the collisional event occurs, however it is performed in the following way: the velocity of particle A_i is updated with probability $W_B / \max\{W_A, W_B\}$ and the velocity of particle B_j is updated with probability $W_A / \max\{W_A, W_B\}$. The post-collision velocities are given by:

$$\mathbf{v}'_A = (m_A + m_B)^{-1} (m_A \mathbf{v}_A + m_B \mathbf{v}_B + (m_B g_{AiBj}) \cdot \mathbf{R}),$$

$$\mathbf{v}'_B = (m_A + m_B)^{-1} (m_A \mathbf{v}_A + m_B \mathbf{v}_B - (m_A g_{Ai,Bj}) \cdot \mathbf{R}),$$

and, as before, \mathbf{R} stands for a random unit vector, and $g_{Ai,Bj}$ is the relative speed.

The collisional procedure presented above can be interpreted as follows. In the first stage of the collisional process, a first tentative collision is regarded as true with a probability of P_{max} (this drives the selection of the collision partners). Let us call the latter as the first tentative collision. In the second stage, the first tentative collision is regarded as true with probability $q_{ij}^{s,r}$, which we call as the second tentative collision that is evaluated by using a random number; and, in the third collisional stage, simulated particle A_i undergoes the collision with simulated particle B with probability $W_B / \max\{W_A, W_B\}$ and particle B undergoes the collision with probability $W_A / \max\{W_A, W_B\}$. The latter stage introduces the role that different weighting schemes play in the collision.

4.2 Coulomb collision algorithm

Although based on physical considerations, the Nanbu's Coulomb collision algorithm [15] is one for solving the Landau-Fokker-Planck equation, and it is based on the idea that many small-angle Coulomb collisions can be grouped into one large one, which allows for using larger time steps that results in larger scattering angles, e.g., it may be convenient to match the time increment used to evolve the collisionless motion and the collisional one. Alternative approaches and developments can be found in [10],[11].

Essentially, Coulomb collisions are a many-body interaction, provided they are long range. Due to the Debye shielding in plasmas, a charged particle interacts only with other charged particles closer than the Debye length; thus, the collisions can be treated separately for each cell. Assuming a binary mixture of charged particles e and i , and the species dependent particle-weighting scheme, the maximum number of e - i collisions can be written as

$$N_{\max}^{e,i} = \max\{N_e', N_i'\}, \quad (25)$$

where, $N_e' = \frac{N_e \cdot n_i}{n}$ and $N_i' = \frac{N_i \cdot n_e}{n}$ are the mean number of collisions between unlike simulated particles inside the cell, N_e and N_i are the number of simulated particles of species e and i , respectively, $n = n_i + n_e$ is the total number density that is the summation over the number densities, $n_e = N_e \cdot W_e \cdot V_c^{-1}$, of each species in the cell, W_e and W_i are the species dependent weight and V_c is the cell volume. Thereby, the algorithm is:

1. Make two random vectors of simulated particles e and i in a cell: $(e_1, e_2, \dots, e_{N_e}), (i_1, i_2, \dots, i_{N_i})$;

2. Make $N_{\max}^{e,i}$ pairs of the form (e_j, i_j) . In the case of

$N_{\max}^{e,i} > N_i'$, make the pairs $\{(e_j, i_j) : j=1, \dots, N_i'\}$;

$(e_k, i_l') : k=N_i'+1, \dots, N_{\max}^{e,i}, l=1, \dots, N_{\max}^{e,i} - N_i'\}$, where the i_l' represents the particle i_l after its first Coulomb collision; the same can be done for the case $N_{\max}^{e,i} > N_e'$.

3. For each random pair (e_j, i_j) , the particle e_j collides with particle i_j with probability $W_i / \max\{W_i, W_e\}$, and particle i_j undergoes the collision with probability $W_e / \max\{W_e, W_i\}$ as well (see reference [15]). Using a random number we can decide which collisions occur or not. Note that at least one simulated particle is scattered. The velocity(s) of the particle(s) that collides is scattered as follows:

a. Calculate the parameter $\tau = A_{ei} \Delta t / \|\mathbf{v}_{e_j} - \mathbf{v}_{i_j}\|^3$ that indirectly regulates the advance in time, compute

$$A_{ei} = \frac{n}{4\pi} \left(\frac{q_e q_i}{\epsilon_0 \mu_{e,i}} \right) \ln \Lambda_{e,i}, \text{ where } q_i \text{ and } q_e \text{ are the}$$

charge of the simulated particles, ϵ_0 is the dielectric constant of vacuum, n is the total number

density, $\Lambda_{ei} = \frac{4\pi \epsilon_0 \mu_{e,i} \langle g_{e,i}^2 \rangle \lambda_D}{|q_e q_i|}$, $\mu_{e,i}$ is the

reduced mass, and the mean square relative velocity $\langle g_{e,i}^2 \rangle$ can be approximated by using the Maxwellian distributions for species e and i , $\langle g_{e,i}^2 \rangle = 3kT_e/m_e + 3kT_i/m_i + (\langle \mathbf{v}_e \rangle - \langle \mathbf{v}_i \rangle)^2$, where T_e is the temperature of species e and $\langle \mathbf{v}_e \rangle$ is the flow velocity.

b. Compute $\cos \chi = [A(\tau)]^{-1} \cdot \ln(\exp(-A(\tau)) + 2U \sinh A(\tau))$ where χ is the deflection angle, $U \sim U[0,1]$ and A must satisfy $\coth A - A^{-1} = \exp(\tau)$, (see [18]);

c. If simulated charged particle e_i collides, then its post-collision velocity is given by:

$$\mathbf{v}'_{e_j} = \mathbf{v}_{e_j} - \frac{m_i}{(m_e + m_i)} (\mathbf{g}_{e,i} \cdot (1 - \cos \chi) + \mathbf{h} \cdot \sin \chi),$$

similarly, if particle i_j truly collides, the resulting scatterings in its velocity components are given

$$\text{by: } \mathbf{v}'_{i_j} = \mathbf{v}_{i_j} + \frac{m_e}{(m_e + m_i)} (\mathbf{g}_{e,i} \cdot (1 - \cos \chi) + \mathbf{h} \cdot \sin \chi),$$

where $\mathbf{g}_{e,i} = \mathbf{v}_e - \mathbf{v}_i$, and $\mathbf{h} = (h_x, h_y, h_z)$, with:

$$\begin{cases} h_x = g_{\perp} \cos \psi \\ h_y = -(g_x g_y \cos \psi + g g_z \sin \psi) / g_{\perp} \\ h_z = -(g_x g_z \cos \psi + g g_y \sin \psi) / g_{\perp} \end{cases}$$

$g_{\perp} = (g_y^2 + g_z^2)^{1/2}$, and ψ represents an azimuthal angle of the collisional plane, $\psi = 2\pi U$, $U \sim U[0,1]$;

4. For species e and i , form the pairs of like particles that have not been scattered yet:

$\{(e_j, e_{j+1}): j = N'_e + 1, \dots, N_e; (i_k, i_{k+1}): k = N'_i + 1, \dots, N_i\}$. In the case of $N_e - (N'_e + 1)$ is an odd number, the last particle of the particle list of species e , e_{N_e} , will not have a “non-collided partner” available, thus, the last pair is defined as $(e_{N_e}, e'_{N'_e+1})$ where $e'_{N'_e+1}$ denotes the second collision of the simulated particle $e_{N'_e+1}$. The same is done when $N_i - (N'_i + 1)$ is an odd number. Note however that, in this model, when a particle collides it has been advanced by Δt , then if a particle collides twice in “one time step”, the parameter τ must be shorten, or adjusted, in step 3.a., Actually, the referred particle would have been advanced by $2\Delta t$ while the physical system only Δt ; thus τ should be adjusted as $\tau^{SS} = (N''_S(N''_S + 1)^{-1}) \cdot \tau$ where $N''_S = N_S - N'_S$ and $S = \{e/i\}$. The particles scatterings in the velocities are as in step 3.

5. Final comments

Non-collisional plasma simulation was formerly implemented by a electrostatic PIC model coupled to a FEM field solver [19], as described above. In this work, we have presented an approach to include the modeling of elastic and Coulomb collisions. The corresponding algorithms are based on a Monte Carlo method and are currently being integrated to the PIC implementation. The next step will be the inclusion of other collisional effects, according to the type of plasma being studied.

6. References

- [1] Babovsky, H. & Illner, R., 1989. A convergence proof for Nanbu's simulation method for the full Boltzmann equation. *SIAM Journal of Num. Anal.*, vol. 26, pp.45-65.
- [2] Birdsall, C. K. & Langdon, A. B., 1985. *Plasma Physics Via Computer Simulation*, McGraw-Hill.
- [3] Bird, G. A. 1976. *Molecular Gas Dynamics*, Oxford, U.K.: Clarendon Press.
- [4] Cheng, S., Santi, M., Celik, M., Martinez-Sanches, M. & Paire, J., 2004. Hybrid PIC-DSMC simulation of a Hall thruster plume on unstructured grids. *Computer Physics Communications*, vol. 164, pp.73-79.
- [5] Dubinov, A. E., 2000. A particle-in-cell simulation of a process of avalanche developing at a non-completed sliding discharge. *Plasma Sources Sci. Technol.*, vol. 9, pp.597-599.
- [6] Ercolano, B., Barlow, M. J., Storey, P. J. & Liu, W.-W., 2003. MOCASSIN: a fully three dimensional Monte-Carlo photoionization code. *Mon. Not. R. Astron. Soc.*, vol. 340, pp.1136-1152.
- [7] Garrigues, L., Heron, A., Adam, J. C. & Boeuf, J. P., 2000. Hybrid and particle in cell model of a stationary plasma thruster. *Plasma Sour. Sc. Tech.*, vol. 9, pp. 219-226.
- [8] Greschik, F. & Kull, H.-J., 2004. Two dimensional PIC simulation of atomic clusters in intense laser fields. *Laser and Particle Beams*, vol. 22, pp.137-145.
- [9] Hewett, D. W., 2003. Fragmentation, Merging, and internal dynamics for PIC simulation with finite size particles. *J. Comput. Physics*, vol. 189, pp. 390-426.
- [10] Lewandowski, J. L. V., 2005. Low noise collision operators for particle-in-cell simulations. *Physics of Plasma*, vol. 12, paper number 052322, pp.1-10.
- [11] Larson, D., 2003. A Coulomb collision model for PIC plasma simulation. *J. Comp. Physics*, vol. 188, pp.123-138.
- [12] Matyash, K. Schneider, R., Bonnin, X., Coster, D., Rohde, V. & Kersten, H., 2005. Modeling of parasitic plasma under the divertor roof baffle. *J. Nuclear Materials*, vol. 337-339, pp. 237-240
- [13] Melenk, J. M. & Babuska, I., 1996. The partition of unit finite element method: Basic theory and applications. *Comput. Meth. Appl. Engn.*, vol. 139, pp.289-314.
- [14] Nam, S.K., Donnelly, V.M. & Economou, D.J., 2005. Particle-in-Cell simulation of ion flow through a hole in contact with plasma. *IEEE Trans. Plasma Sc.*, vol. 33(2), pp. 232-233.
- [15] Nanbu, K. & Yonemura, S., 1998. Weighted Particles in Coulomb Collision Simulations Based on the Theory of Cumulative Scattering Angle, *J. Comput. Physics*, vol. 145, pp. 639-654.
- [16] Nanbu, K. & Denpoh, K., 1998. Monte Carlo collision simulation of positive-negative ion recombination for a given rate constant. *J. Phys. Soc. Jpn.*, vol.67, pp.1288-1290.
- [17] Nanbu, K., Mitsui, K. & Kondo, S., 2000. Self-consistent particle modelling of dc magnetron discharges of na O_2/Ar mixture. *J. Physics D.: Appl. Phys.*, vol. 33, pp. 2274-2283.
- [18] Nanbu, K., 2000. Probability Theory of Electron-Molecule, Ion-Molecule, and Coulomb Collisions for Particle Modeling of Materials Processing Plasmas and Gases. *IEEE Trans. Plasma Sc.*, vol. 28, pp.971-990.
- [19] Passaro, A., Abe, N. M., Paes, A. C. J., Marques, G. N., Preto, A. J. Stephany, S. 2004. A Parallel and Object-Oriented Plasma Simulation Code Base don Electrostatic PIC Model and the Finite Element Method. In *2004, XXV Iberian Latin American Congress on Comput. Meth. Engng. (CILAMCE 2004)*, Recife – PE, Brazil.
- [20] Rantamäki, K. M, Pättikangas, T. J. H., Karttunen, S. J. *et al.*, 1999. Particle-in-cell simulations of parasitic absorption of lower hybrid power in edge plasmas of tokamaks. *Plasma Phys. Control. Fusion*, vol. 41, pp. 1125-1133.
- [21] Shang, J. S., 2001. Recent research in magneto-aerodynamics, *Progress in Aerospace Sc.*, vol. 37, pp. 1-20.