Turkish Journal of Physics

Volume 47 | Number 4 Article 4

2023

Development and benchmark of a 1d3v electrostatic PIC/MCC numerical code for gas discharge simulations

İBRAHİM ARDA

İSMAİL RAFATOV

TÜBİTAK

Turkish Journal of Physics

http://journals.tubitak.gov.tr/physics/

Research Article

Turk J Phys
(2023) 47: 198 – 213

doi:10.55730/1300-0101.2746

Development and benchmark of a 1d3v electrostatic PIC/MCC numerical code for gas discharge simulations

İbrahim ARDA[1,2,*]; **İsmail RAFATOV**[1]

[1]Department of Physics, Middle East Technical University, Ankara, Turkiye

[2]Department of Astronautical Engineering, University of Turkish Aeronautical Association, Ankara, Turkiye

Received: 05.06.2023 • **Accepted/Published Online:** 11.08.2023 • **Final Version:** 25.08.2023

Abstract: Gas discharge plasmas are low-temperature nonequilibrium plasmas that have a wide range of scientific and technological applications. The particle-in-cell/Monte Carlo collision (PIC/MCC) method is a reliable approach for numerical analyses and simulations of such plasmas. In this study, we first provide a detailed description of the basics of this method. We then verify the programme code that we developed in Fortran by benchmarking the code against a widely referred reference study simulating a radiofrequency capacitively coupled plasma (RFCCP) in helium for various discharge conditions. We show that the results of the present study are in good agreement with that of the reference benchmark study. We also demonstrate that although PIC/MCC is a computationally demanding method, it is still possible to conduct some one-dimensional plasma simulations with standard personal computers within hours, thanks to recent advancements in computing technologies. More rigorous and especially high-dimensional problems, however, require acceleration and parallelization strategies.

Keywords: Kinetic plasma simulation, particle-in-cell/Monte Carlo collision, radiofrequency capacitively coupled plasma, benchmark

1. Introduction

Modelling and simulation are essential elements in the studies in applied sciences, and plasma science is not an exception to that. Among the various methods developed for a wide variety of plasma types, particle-based simulation techniques stand out, especially in the study of low-temperature and nonequilibrium plasmas. The particle-in-cell (PIC) method, specifically, has been used for more than half a century for various plasma simulations since its first appearance in the late 1950s and the beginning of the 1960s by Buneman [1] and Dawson [2]. In the late 1960s, the Monte Carlo collision (MCC) method was already appended to the PIC for handling collisional events [3–5]. Since then, the PIC/MCC method has been used in the analysis of numerous plasma phenomena, such as plasma thrusters [6, 7] and impurity identification in gases [8].

Radiofrequency capacitively coupled plasma (RFCCP) is a common type of low-pressure gas discharge. In this type of discharge, the plasma is sustained by coupling the electrical power to the plasma directly through electrodes. Capacitive discharges are thoroughly investigated in the literature

*Correspondence: iarda@thk.edu.tr

[9]. RFCCP has various application areas, such as thin film deposition, etching, microelectronics, and space propulsion systems [10, 11]. PIC/MCC has recently gained popularity among several models and simulation methods that have been developed for RFCCP discharges over the years [12].

PIC/MCC is a stochastic method that is implemented through computer simulations. Verifying the results of any computer simulation is an indispensable stage. Although the best way of this verification is to compare the simulation results with empirical outcomes, experimental results or even the experiments themselves are not always accessible. Benchmarking the developed programme codes with other independently developed codes is an appropriate alternative at this point. Actually, it would not be an overstatement to suggest that benchmarking a programme implementation of any discharge model is a minimum requirement [13].

In the present study, we describe the fundamentals of the PIC/MCC method and utilize this method in a 1d3v electrostatic programme code in Fortran. The main purposes of this paper are to provide a detailed introduction of the PIC/MCC method and verify the developed code through a benchmark study by Turner et al. [14]. This is a well-chosen benchmark study for a couple of reasons. First, the discharge conditions in that study were determined so that they could be associated with the experimental studies of Godyak et al. [15]. Second, their study has similar characteristics to an earlier benchmark study by Surendra [16], yet has more consistent results. Finally, they examined the effects of various numerical parameters to specify any remaining ambiguity in the benchmark data and shared all outcomes in an easy-to-access electronic supplement*.

The layout of the present study is as follows: The details of the PIC/MCC method and benchmark implementation are given in Section 2. In Section 3, the results of the present and the reference benchmark studies are compared to verify the developed programme code. Finally, the study is concluded with a brief summary in Section 4.

2. Methodology and implementation

2.1. PIC/MCC method

Hockney and Eastwood [17] identified the PIC as a particle-mesh (PM) type method. The fundamental philosophy behind this methodology is that each particle is followed in a continuous space, and in the meantime, field quantities such as densities, potentials, etc., are approximated on the mesh points of a spatial mesh. And thence, the electromagnetic forces on each particle can be calculated using the discretised field values on the mesh points. The most significant achievement of this approach is that consideration of the mutual interactions of each particle turns out to be unnecessary. The other principal part of the method is the concept of superparticles. Each particle in the numerical method represents many (typically thousands of) real physical particles. These are the essence of the PIC method that makes plasma simulations even possible, as it would be a hopeless endeavour to follow millions of trillions of particles and account for their mutual interactions with one another.

The plasma-chemical processes (i.e. collision events), on the other hand, are taken into account stochastically through the Monte Carlo collision (MCC) method. Whether a particle experiences a collision event and what type of collision event is to take place are determined through comparing random numbers with the probabilities obtained from collision frequencies. The basic steps of the electrostatic PIC/MCC procedure can be summarized in a main algorithm given in Figure 1.

*The supplementary material is available at https://doi.org/10.1063/1.4775084.

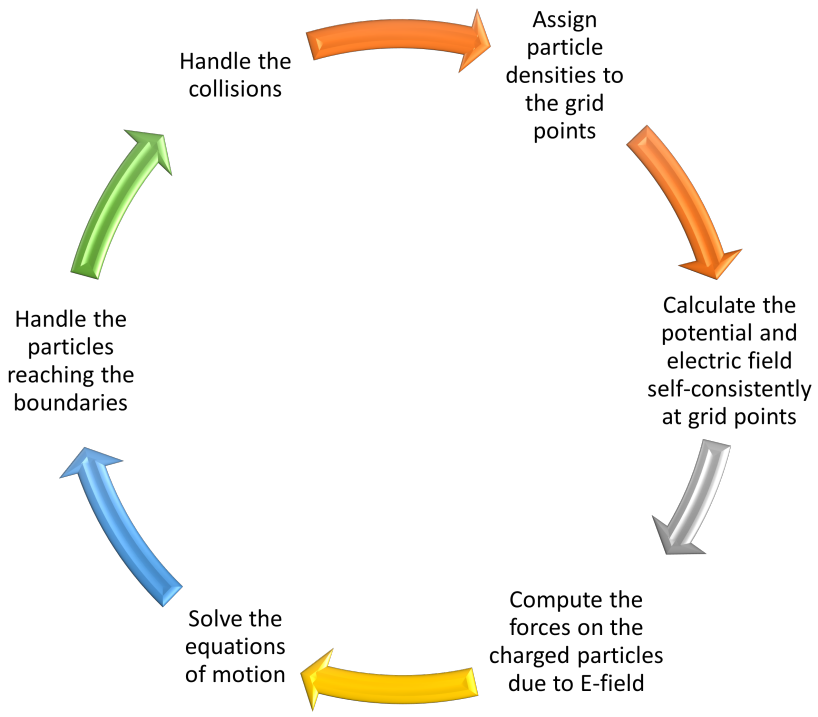

Figure 1: The cycle of the PIC/MCC procedure for the electrostatic case.

A prerequisite to the method is the construction of a spatial mesh. A one-dimensional grid with equal lengths of cell spacings is utilized for the construction of the spatial mesh in this study. Numbering of the cell points starts with 0; hence, the length of each cell spacing becomes $\Delta x = L/(N-1)$ for a grid of length L with a total number of N grid points. Subsequently, charge densities are assigned to grid points depending on the positions of particles. When this assignment is done according to the first-order accuracy, it amounts to the linear interpolation of the number densities to the grid points, and the particles effectively become triangular-shaped clouds according to the grid points. This approach is either called the particle-in-cell (PIC), by following the former viewpoint [18], or the cloud-in-cell (CIC) model, by following the latter viewpoint [19]. Let X_j and x_i be the positions of the j-th grid point and i-th particle. Then, for a particle with position x_i, where $X_j \leq x_i \leq X_{j+1}$, the number density is distributed on j-th and $(j+1)$-th grid points as follows [12]:

$$\delta n_j = \left[(j+1) - \frac{x_i}{\Delta x}\right] \frac{W}{A\Delta x}, \tag{2.1}$$

$$\delta n_{j+1} = \left(\frac{x_i}{\Delta x} - j\right) \frac{W}{A\Delta x}. \tag{2.2}$$

Here, W is the weight that gives the number of real physical particles constituting each superparticle and A is the cross-sectional area of the discharge region. Note that in the case of a one-dimensional simulation, this area is only used to establish the relation between superparticles and number density of real particles. Its value and unit are arbitrary and can be set to, for instance, $A = 1\ \mathrm{cm}^2$ [12].

δ notation in Eqs. (2.1) and (2.2) points out the contribution of each individual superparticle. For a total of M superparticles of particle species s which contribute the grid point, say, j, the total number density of the corresponding species on the specified grid point is simply obtained by adding the contribution of each superparticle:

$$n_{j,s} = \sum_{k=1}^{M} (\delta n_{j,s})_k \,. \tag{2.3}$$

Consequently, the total charge density on j-th grid point becomes

$$\rho_j = \sum_s q_s n_{j,s}, \tag{2.4}$$

where q_s is the charge of species s.

Charged particles both modify and are affected by potential and electric field distributions within the discharge region. To resolve this complex situation, Poisson's equation ($\nabla^2\varphi = -\rho/\varepsilon_0$) needs to be coupled to the system. To this end, it is discretized with the second-order central difference scheme, as follows:

$$\frac{\varphi_{j-1} - 2\varphi_j + \varphi_{j+1}}{\Delta x^2} = -\frac{\rho_j}{\varepsilon_0}. \tag{2.5}$$

Eq. (2.5) is applied for internal nodes of the grid, whereas potentials on the electrodes are included to the system as boundary conditions. Along with the boundary conditions, Eq. (2.5) constitutes a tridiagonal system of equations in its matrix form and the Thomas algorithm is utilized for the solution of this system due to its efficiency. Subsequently, electric field values at grid points are obtained from the relation $\mathbf{E} = -\nabla\varphi$. Applying the second-order central difference scheme on this relation yields

$$E_j = \frac{\varphi_{j-1} - \varphi_{j+1}}{2\Delta x}. \tag{2.6}$$

Grid points standing on the boundaries, however, require special treatment [12]:

$$E_0 = \frac{\varphi_0 - \varphi_1}{\Delta x} - \frac{\rho_0 \Delta x}{2\varepsilon_0}, \tag{2.7}$$

$$E_{N-1} = \frac{\varphi_{N-2} - \varphi_{N-1}}{\Delta x} + \frac{\rho_{N-1}\Delta x}{2\varepsilon_0}. \tag{2.8}$$

Having obtained the electric field values at grid points, it is then possible to calculate electromagnetic forces through the Lorentz force law: $\mathbf{F} = q\mathbf{E}$ (for the electrostatic case). It is important to notice, however, that calculation of forces on the particles requires electric field values at the positions of particles. Thus, electric field values calculated at grid points should be interpolated back to particle positions. In order to avoid the development of a self-force while performing this back interpolation, Birdsall and Langdon [18] advised using the same order interpolation scheme as that used in distributing the particle densities to grid points. Consequently, for a particle of charge q at position x_i that is situated within grid points j and $j+1$, the electrostatic force is given by $F_i = qE_i$, where

$$E_i = E_j \frac{X_{j+1} - x_i}{\Delta x} + E_{j+1} \frac{x_i - X_j}{\Delta x}. \tag{2.9}$$

After the forces on particles are determined, the equations of motion can be solved to update particle positions and velocities as a result of their response to the fields. The explicit leapfrog integration method [17] is an appropriate option due to its attractive features, such as its simplicity, second-order accuracy, and symplectic nature. In this method, velocity updating time steps are not synchronized with those of the positions. Rather, the positions are calculated at integer time steps, whereas the velocities are calculated at half integer time steps, lagging a half step size:

$$v^{k+1/2} = v^{k-1/2} + \frac{q}{m} E^k \Delta t, \tag{2.10}$$

$$x^{k+1} = x^k + v^{k+1/2} \Delta t. \tag{2.11}$$

Note that the superscripts in Eqs. (2.10) and (2.11) show time steps. Note also that when necessary, the velocity can be advanced through the following equation for another half step size to yield integer time step value:

$$v^{k+1} = v^{k+1/2} + \frac{qE^{k+1}}{m} \frac{\Delta t}{2}. \tag{2.12}$$

While moving particles for a time step size, it is possible that some particles may reach the discharge boundaries. It is important to identify those particles and to handle their interactions with boundary surfaces. Those interactions may occur in a variety of ways: particles may be absorbed by or reflected off of the boundary surfaces, or they may rip off new (secondary) particles from the surfaces. Reflection and secondary particle emission processes are stochastically included through relevant coefficient values that depend on the material of boundary surfaces and they may play significant roles on various discharge characteristics depending on the discharge conditions. Hence, it is important to perceive the extent of the study and include necessary boundary processes correctly. The effects of boundary processes for various discharge characteristics have been studied in the literature, e.g., [20–24].

The final step of the main algorithm is handling collisions through the Monte Carlo collision method. MCC is a stochastic method based on comparing random numbers with the probabilities of events to determine whether they occur. To determine, for instance, whether a particle is to experience a collision event during a time step Δt, a random number R_{0-1}^U is compared with the collision probability of the particle [12]:

$$P_{\Delta t} = 1 - \exp\left[-n_g v_r \sigma\left(v_r\right) \Delta t\right]. \tag{2.13}$$

Note that the notation R_{0-1}^U indicates that the random number is drawn from a uniform distribution on the interval $[0, 1)$. Note also that in Eq. (2.13), n_g is the number density of neutral background gas, v_r is the relative speed of incident particle with respect to the target particle, and σ is the cross section of the collision event. If the value of this probability is greater than R_{0-1}^U, a collision event takes place for the particle under investigation. It is important to notice from Eq. (2.13) that the time

step size directly affects the value of collision probability. It is also important to emphasize that any particle can experience at most one collision event per time step in MCC method. This introduces an error in handling the collisions, depending on the number of missed collisions. Hence, the choice of time step size, Δt, has a crucial role in the reliability of simulations. Vahedi and Surendra [25] advised that Δt shall satisfy the following equation:

$$n_g v_r \sigma\left(v_r\right) \Delta t \leq 0.1 \tag{2.14}$$

This equivalently means that the collision probability is not to exceed 0.095. Donko et al. [12], on the other hand, suggested to keep the collision probability below about 0.05. The stability and accuracy criteria including the time step size of the PIC/MCC method have been recently revisited and investigated thoroughly in [26].

In the classical MCC method, the collision checks are conducted for all the particles one by one. During the checks, for those particles that are to undergo a collision event, the type of collision is also determined by comparing a random number with the probabilities of each collision event. The probability of, for instance, j-th collisional process is calculated from [12]

$$P_j = \frac{\sigma_j}{\sigma_T}, \tag{2.15}$$

where σ_j is the cross section for j-th collisional process and σ_T is the total cross section for all collision types. Considering a total number of N collisional processes, the interval $[0, 1)$ is divided into N segments, of length P_j for the j-th process. The ordering of these processes follows the ascending order of the corresponding events' threshold energies. Then, saying that a random number R_{0-1}^U belongs to the first interval means that it satisfies the inequality $0 < R_{0-1}^U \leq P_1$. Similarly, if R_{0-1}^U belongs to the second interval, then $P_1 < R_{0-1}^U \leq P_1 + P_2$, and so on. Hence, if the random number has a value that satisfies

$$\sum_{k=1}^{l-1} P_k < R_{0-1}^U \leq \sum_{k=1}^{l} P_k, \tag{2.16}$$

then the l-th collisional process occurs. A visual representation of this procedure is given in Figure 2.

Collision handling procedure includes determining energy transfers between colliding partners, the details of which depend on the type of collision event, as well as calculating the velocities of particles after the collision. The latter involves assessing the deflection angles and magnitudes of the velocities of scattered particles. The deflection angles are comprised of two components: the scattering angle χ, which denotes the angle between the velocity vectors of the incoming particle before and after the scattering event, and the azimuth angle η, which represents the angle between the postcollisional velocity and the normal to the precollisional velocity. In our convention, assuming the precollisional velocity is oriented in the east direction, η is measured from the north direction. It is important to note that the collisions and deflection angles are resolved in the centre of mass (CoM) frame. Hence, the velocities of particles should be written in the CoM frame when handling the collisions. From the definition of CoM, the velocity of CoM for a system of two colliding particles that consists of constant masses is given by

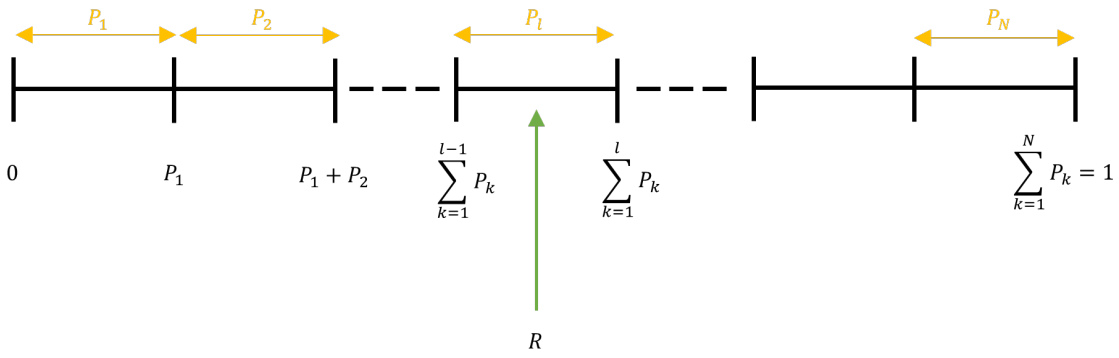

Figure 2: Intervals of collisional processes. P_j is the probability for the j-th process, calculated from Eq. (2.15). The processes are arranged in ascending order of threshold energies, from lowest to highest. If the random number R_{0-1}^{U} falls into the shown interval, then the l-th collisional process occurs.

$$\mathbf{v}_{CoM} = \frac{m_p \mathbf{v}_p + m_t \mathbf{v}_t}{m_p + m_t}, \tag{2.17}$$

where m_p, m_t, $\mathbf{v}_p$, and $\mathbf{v}_t$ are the masses and velocities of projectile and target particle, respectively. It is important to note that in this study, only electron-neutral and ion-neutral collisions are considered. Thus, the primary particles being followed are either electrons or ions, and the target particles are the neutral background atoms or molecules. Eq. (2.17) can be written for postcollisional velocities, as well:

$$\mathbf{v}'_{CoM} = \frac{m_p \mathbf{v}'_p + m_t \mathbf{v}'_t}{m_p + m_t}, \tag{2.18}$$

where the primes indicate the velocities of the colliding particles after the collision. Eq. (2.18) can be solved for $\mathbf{v}'_p$ to have

$$\mathbf{v}'_p = \left(1 + \frac{m_t}{m_p}\right) \mathbf{v}'_{CoM} - \frac{m_t}{m_p} \left(\mathbf{v}'_p - \mathbf{v}'_r\right),$$

where $\mathbf{v}'_r \equiv \mathbf{v}'_p - \mathbf{v}'_t$ is the relative velocity of colliding particles after the collision. Further rearrangements yield

$$\mathbf{v}'_p = \mathbf{v}'_{CoM} + \frac{m_t}{m_p + m_t} \mathbf{v}'_r. \tag{2.19}$$

Note that since the total momentum is conserved and we consider a closed system, the velocity of the CoM does not change after the collision, i.e. $\mathbf{v}'_{CoM} = \mathbf{v}_{CoM}$ and hence

$$\mathbf{v}'_p = \mathbf{v}_{CoM} + \frac{m_t}{m_p + m_t} \mathbf{v}'_r. \tag{2.20}$$

Similarly, solving Eq. (2.18) for $\mathbf{v}_t'$ with $\mathbf{v}_{CoM}' = \mathbf{v}_{CoM}$ yields,

$$\mathbf{v}_t' = \mathbf{v}_{CoM} - \frac{m_p}{m_p + m_t}\mathbf{v}_r'. \tag{2.21}$$

Note that in the case of electron-neutral collisions, neutral particles can be assumed to be stationary with respect to electrons due to the sufficiently low electron-to-neutral mass ratio. This assumption is called the cold gas approximation [12]. In this case, it is reasonable to assume that $\mathbf{v}_t = 0$, and hence the relative velocity simply equates to the electron velocity, $\mathbf{v}_r = \mathbf{v}_e$. In ion-neutral collisions, however, this is not a valid assumption as the ions and neutral particles are comparable in mass and thus in momentum.

The procedures for determining the deflection angles and the postcollision relative velocity are distinct for elastic and inelastic collisions. Furthermore, various approaches might be used to calculate the scattering angle χ depending on whether an isotropic or anisotropic scattering case is considered for a specific collision type. When a model or a set of experimental data exists for the differential cross section as a function of both energy and the scattering angle, one can consider the anisotropic scattering case[†] and obtain the scattering angle χ through the relation [29]

$$\frac{\int_0^{\chi} \sigma(\varepsilon, \chi') \sin\chi' d\chi'}{\int_0^{\pi} \sigma(\varepsilon, \chi') \sin\chi' d\chi'} = R_{0-1}^U. \tag{2.22}$$

For instance, in [27], an analytical expression is given for an anisotropic scattering case in an attempt to correct an erroneous approach that had been given in a previous study in [28]. In the present study, however, the isotropic scattering case is considered. For elastic collisions, the scattering angle χ and the azimuth angle η are calculated through Eqs. (2.23) and (2.24) [29], respectively:

$$\chi = \cos^{-1}\left(1 - 2R_{0-1}^U\right), \tag{2.23}$$

$$\eta = 2\pi R_{0-1}^U. \tag{2.24}$$

The magnitude of the relative velocity does not change in the case of an elastic collision, hence $v_r{}' = v_r$.

For electron-neutral inelastic collisions, on the other hand, the magnitude of the relative velocity is not constant, and its change is calculated based on the change in the kinetic energy of the system of colliding particles. In an inelastic collision case for which the threshold energy of the chemical process is $\varepsilon_{th.}$, the kinetic energy of the system of colliding particles after the collision event is given by

$$\varepsilon' = \varepsilon - \varepsilon_{th.}. \tag{2.25}$$

It is important to recall that the collision events are handled in the CoM frame. Hence, the energies in Eq. (2.25) are calculated for the reduced system of the colliding particles:

$$\varepsilon = \frac{1}{2}\mu(v_r)^2, \tag{2.26}$$

[†]Note that in the case of an anisotropic scattering, a transformation of the velocity vector of the incoming particle is necessary prior to and after the collision event. A detailed explanation is given in [12].

$$\varepsilon' = \frac{1}{2}\mu(v_r')^2, \tag{2.27}$$

where μ is the reduced mass of the system of colliding particles. In an excitation collision, the deflection angles are determined from Eqs. (2.23) and (2.24), whereas in an ionization collision, they are obtained through [12]

$$\chi_{sc.} = \cos^{-1}\left(\sqrt{\frac{\varepsilon_{sc.}}{\varepsilon'}}\right), \quad \eta_{sc.} = 2\pi R_{0-1}^{U}, \tag{2.28}$$

and

$$\chi_{ej.} = \cos^{-1}\left(\sqrt{\frac{\varepsilon_{ej.}}{\varepsilon'}}\right), \quad \eta_{ej.} = \eta_{sc.} + \pi, \tag{2.29}$$

where the subscripts *sc.* and *ej.* indicate scattered and ejected electrons after the collision event, respectively. This requires a choice of sharing the remaining energy ε' between the scattered and ejected electrons. This may be done according to various formulations obtained from experimental data.

For cases involving large numbers of superparticles, the classical MCC method can be rather time-consuming as it involves the computationally expensive operations of calculating the kinetic energies, collision probabilities, postcollision velocities, and angles of each particle participating in the collision event. In an attempt to resolve this, the null collision method is used. This method is based on the idea of obtaining a constant total collision frequency by the addition of a new collisional process (null collision) and was first introduced by Skullerud [5] for the problem of determining the free times between collisions for ion motion in a gas that is subjected to a constant electric field. Since then, the method has been adopted and further improved in various PIC/MCC and swarm studies (e.g., [25, 29–33]) and has become a standard method for improving the performance of the codes implemented for those studies. In the present study, the procedure given in [25] is followed to implement the null collision method. The constant total collision frequency is obtained through

$$\nu_c = \max\left(n_g \sigma_T v\right) = n_g \max\left(\sigma_T v\right), \tag{2.30}$$

since the background gas density is assumed to be spatially and temporally uniform. The main strength of the method comes from the fact that the check for whether a particle collides in each time interval is performed for only a subset of all particles rather than all of them. The number of particles to be checked for collision is determined by multiplying the total number of particles by the factor

$$P_{null} = 1 - \exp\left(-\nu_c \Delta t\right). \tag{2.31}$$

Those particles are randomly chosen among all particles without duplicates; i.e. there are no particles that are checked more than once whether they participate in collisions, and hence any particle can only experience a collision once at most within one time step. Once those particles are chosen, a random number is compared with fractional collision frequencies of each collision type and the constant collision frequency. Let

$$q_i \equiv \frac{\sum_{k=1}^{i} \nu_k}{\nu_c}, \tag{2.32}$$

where the subscript i designates the type of collision. If the relation $q_{i-1} < R_{0-1}^U \leqslant q_i$ holds for the random number R_{0-1}^U, then the i-th collision event occurs, where the ordering of collision events is done according to the threshold energies of collision types in an ascending manner. For a total of M collision types, if

$$\frac{\sum_{k=1}^{M} \nu_k}{\nu_c} < R_{0-1}^U, \tag{2.33}$$

then the null collision occurs. In that case, no real collision takes place and the iteration continues with the next particle. The null collision method imposes an additional computational load to select colliding particles randomly and in a nonduplicated manner. If the value of P_{null} is sufficiently low, however, reduction in the number of the floating-point operations compared to the classical MCC method can be rather high.

2.2. Benchmark implementation

Simulations are carried out with a 1d3v electrostatic code that was developed by authors according to well-known algorithms [12, 25, 29]. The code is written serially in Fortran. For the validation of the code, a relatively recent and widely referred benchmark study by Turner et al. [14] is utilized. They developed their benchmark study for low-pressure capacitive discharge in helium, at a sinusoidally varying voltage of frequency 13.56 MHz, so that simulations would be associated with the experiments of Godyak et al. [15]. They introduced a benchmark problem with well-defined physical and numerical conditions for various cases and applied five independently established codes, some developed in different programming languages and even with different architectures. The conditions, in this study, are taken to be identical in terms of physical and computational parameters. Some basic parameters of the benchmark cases are given in Table 1. In all simulation cases, data collection for these parameters is conducted for the last 32 RF cycles of the simulation.

Table 1: Basic parameters of benchmark in [14].

Parameter	Symbol [unit]	Case 1	Case 2	Case 3	Case 4
Discharge gap	L [cm]	6.7	6.7	6.7	6.7
Background gas pressure	p [Torr]	0.03	0.1	0.3	1
Background gas temperature	T [K]	300	300	300	300
Discharge voltage	V [V]	450	200	150	120
Simulation time	t [s]	$1280/f$	$5120/f$	$5120/f$	$15360/f$
Cell size	Δx [m]	$L/128$	$L/256$	$L/512$	$L/512$
Time step size	Δt [s]	$(400f)^{-1}$	$(800f)^{-1}$	$(1600f)^{-1}$	$(3200f)^{-1}$
Particle weight	W	26172	52344	52344	78516

Following the benchmark requirements, the plasma-chemical processes included in this study are given in Table 2. Cross section data are taken from the LXCat data centre[‡] and supplied to the code as look-up tables. As in the benchmark study [14], the Biagi 7.1 and Phelps data sets are used for electron-neutral and ion-neutral collisions, respectively. These tables include cross section data only for some discrete energy values. Cross section values for energies other than those are calculated through linear interpolation.

Table 2: Plasma-chemical processes included in the present study.

Process	Reaction	Threshold energy (eV)
Elastic scattering	$e^- + He \rightarrow He + e^-$	–
Excitation (triplet)	$e^- + He \rightarrow He^*(2^3S_1) + e^-$	19.82
Excitation (singlet)	$e^- + He \rightarrow He^*(2^1S_0) + e^-$	20.61
Ionization	$e^- + He \rightarrow He^+ + e^- + e^-$	24.587
Isotropic elastic scattering	$He^+ + He \rightarrow He^+ + He$	–
Backward elastic scattering	$He^+ + He \rightarrow He + He^+$	–

The Biagi 7.1 data set includes one cross section set for the elastic momentum transfer process, two sets for the excitation (singlet and triplet) process, and one set for the ionization process, all of which suppose that the collisions result in isotropic scattering in the CoM frame. The Phelps data set, on the other hand, contains two sets of collisional processes: isotropic scattering and backward scattering, which are the components for an approximation of anisotropic scattering according to Phelps' proposal [34].

Finally, note that, in the case of an electron-neutral ionization collision, the residual energy is shared in half by the scattered and ejected (i.e. ripped off from neutral He atoms) electrons, in parallel with the benchmark study in [14].

3. Results and discussion

The code is implemented on a PC with Intel Core i7-1065G7 CPU processor via Intel IFORT compiler for cases given in Table 1. Run times of simulations for each case are given in Table 3. It is seen that despite the computationally intensive nature of the PIC/MCC method, it may be possible to conduct simulations with a standard personal computer on some occasions. Case 4, on the other hand, reflects how computationally demanding the method is, especially in more complex cases, and that the use of acceleration techniques may become inevitable.

The relative errors (in percent) of basic physical parameters as compared to the corresponding values from the reference study in [14] are demonstrated in Table 4. Note that n_i is the value of ion density at the middle of the discharge region, S_e and S_i are the line integrals of electrical power of electrons and ions, and J_i is the ion current density at the boundaries (i.e. electrodes).

Table 5 displays the relative errors of some numerical parameters in comparison to results from [14]. Time-averaged values of electron number density and mean electron energy in the middle of the

[‡]LXCat is an open-access website that contains useful data for low-temperature plasma modelling. The website can be accessed through https://us.lxcat.net/home/.

Table 3: Total number of simulation time steps and run times (in minutes) of each simulation case in the present study.

Case	Simulation time steps	Run time (min)
1	512,000	14
2	4,096,000	342
3	8,192,000	1716
4	49,152,000	14521

Table 4: Relative errors of some basic physical parameters of the discharge in comparison to the reference values given in [14]. The values of the ion density n_i and the electron temperature T_e are evaluated at the centre of the discharge region. S_e and S_i are the line integrals of electrical power of electrons and ions through the discharge region, respectively. Ion currents J_i are values calculated at either of the two electrodes.

Parameter	Case 1 (%)	Case 2 (%)	Case 3 (%)	Case 4 (%)
n_i	3.570	0.483	$< 10^{-3}$	0.778
$k_B T_e$	3.630	1.070	0.506	1.370
S_e	2.620	0.581	0.939	2.070
S_i	2.870	2.930	2.810	3.690
J_i	0.457	$< 10^{-3}$	0.513	1.080

Table 5: Relative errors of some basic numerical parameters of the discharge in comparison to the reference values given in [14]. The time-averaged values are taken at the centre of the discharge, except for N_P which is the total number of superparticles in the entire discharge region.

Parameter	Case 1 (%)	Case 2 (%)	Case 3 (%)	Case 4 (%)
$\omega_p \Delta t$	$< 10^{-3}$	$< 10^{-3}$	0.909	10^{-3}
$\lambda_D / \Delta x$	1.340	0.935	$< 10^{-3}$	0.935
N_P	2.820	$< 10^{-3}$	$< 10^{-3}$	0.912

discharge region are used to calculate the plasma frequency and the Debye length.

In Figure 3, ion number densities are compared with the reference study [14] results. The reference data sets actually correspond to one standard deviation of fluctuations around equilibrium values after the stationary state is achieved in an extended simulation of one of the codes involved in the benchmark study in [14]. These data sets can be found in the electronic supplements[§] of that study. The curves illustrated in Figure 3 suggest a good agreement in ion density results. In order to provide a more convincing claim of this agreement, the maximum relative errors in ion densities are calculated

[§]The supplementary material of [14] is available at https://doi.org/10.1063/1.4775084.

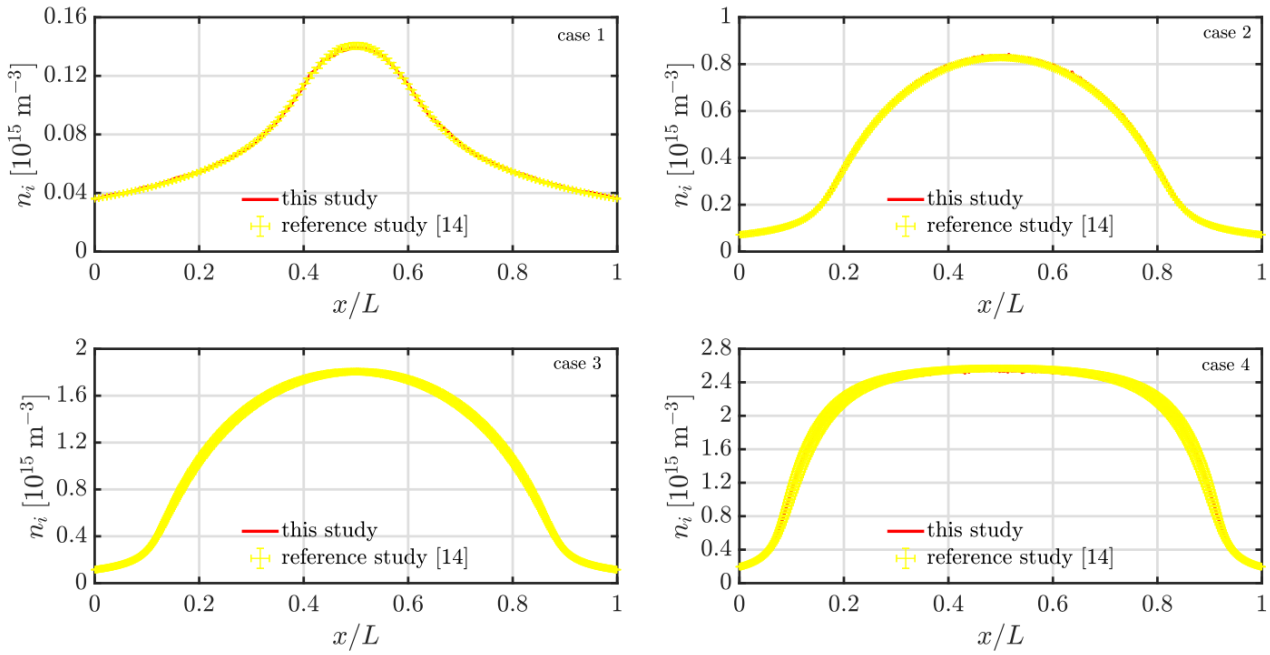

Figure 3: Distributions of ion number densities in comparison to results from [14]. The reference data sets represent one standard deviation of fluctuations around equilibrium values after the stationary state is achieved in an extended simulation of one of the codes involved in the benchmark study in [14].

by considering the reference data from [14] as true values of ion densities. These quantitative results are given in Table 6. It is seen that the divergence in the agreement of results tend to grow from case 1, where the maximum error is 2.48%, to case 4, where the maximum error increases to 4.68%.

Table 6: The maximum relative errors in ion number density distributions of the present study as compared to the reference values from [14].

	Maximum relative errors (in %) of ion number densities
Case 1	2.48
Case 2	3.45
Case 3	4.64
Case 4	4.68

Figure 4 shows comparisons of period-averaged values of the ionization source term $R_{ion.}$ (top left), power density of electrons P_e (top right), and power density of ions P_i (bottom left), respectively. Satisfying agreements are observed in these figures as well. In each of these figures, it is obvious that ionization source and power densities tend to concentrate in sheath regions as the pressure is increased.

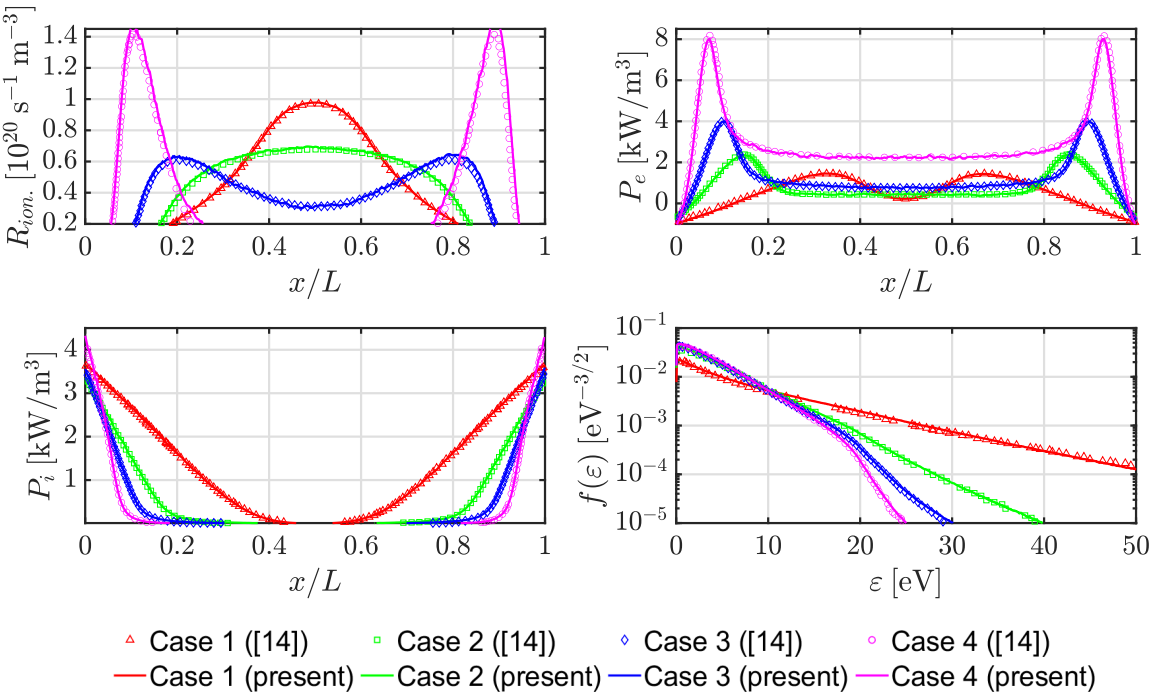

Figure 4: Temporally averaged ionization source terms (top left), electron power densities (top right), ion power densities (bottom left) and spatio-temporally averaged electron energy probability functions (bottom right) in comparison to results from [14].

In Figure 4, temporally and spatially averaged results of electron energy probability function (EEPF) (bottom right) are also compared. A good agreement is obtained in all cases. Note that EEPF is normalized as $\int_0^\infty f(\varepsilon)\sqrt{\varepsilon}d\varepsilon = 1$. It is observed in Figure 4 that the EEPF deviates from the Maxwellian distribution when proceeding from case 1 to 4.

4. Conclusion

The main objective of this study was to develop a reliable computer programme for kinetic simulations of gas discharge plasmas. To achieve this goal, we employed the particle-in-cell/Monte Carlo collision (PIC/MCC) method to write a 1d3v electrostatic programme code. A comprehensive and detailed description of the PIC/MCC method was provided in this study.

To validate the code, we conducted tests using several simulation cases of a well-established benchmark study by Turner et al. [14]. Our comparisons of ion number densities, time-averaged ionization source terms, power densities of electrons and ions, and time and space-averaged electron energy probability function outputs demonstrated satisfactory agreement. It is worth noting that Turner et al. [14] emphasized the sensitivity of ion number density distributions to numerical effects and variations in simulation applications. Therefore, we calculated the maximum relative percent errors in the ion number density distributions in each case to facilitate a more perceptible comparison. These calculations revealed that the results agreed with an error of no more than about 5%, which falls within an acceptable margin.

Based on these findings, we assert that the developed code is a reliable tool for conducting 1d3v electrostatic radiofrequency capacitively coupled plasma (RFCCP) simulations.

References

[1] O. Buneman, "Dissipation of Currents in Ionized Media," *Physical Review* **115** (1959) 503–517.

[2] J. Dawson, "One-Dimensional Plasma Model," *The Physics of Fluids* **5** (1962) 445–459.

[3] P. Burger, "Elastic Collisions in Simulating One-Dimensional Plasma Diodes on the Computer," *The Physics of Fluids* **10** (1967) 658–666.

[4] R. Shanny, J. M. Dawson, and J. M. Greene, "One-Dimensional Model of a Lorentz Plasma," *The Physics of Fluids* **10** (1967) 1281–1287.

[5] H. R. Skullerud, "The stochastic computer simulation of ion motion in a gas subjected to a constant electric field," *Journal of Physics D: Applied Physics* **1** (1968) 1567.

[6] Y. Yamashita, Y. Tani, R. Tsukizaki, D. Koda, and K. Nishiyama, "Numerical investigation of plasma properties for the microwave discharge ion thruster μ10 using PIC-MCC simulation," *Physics of Plasmas* **26** (2019) 073510.

[7] J. W. Koo and I. D. Boyd, "Computational model of a Hall thruster," *Computer Physics Communications* **164** (2004) 442–447.

[8] C. K. Sarikaya, I. Rafatov, and A. A. Kudryavtsev, "Particle in cell/Monte Carlo collision analysis of the problem of identification of impurities in the gas by the plasma electron spectroscopy method," *Physics of Plasmas* **23** (2016) 063524.

[9] M. A. Lieberman and A. J. Lichtenberg, "Capacitive Discharges" in *Principles of Plasma Discharges and Materials Processing*, John Wiley & Sons Ltd (2005) 387–460.

[10] T. Makabe and Z. L. Petrovic, "Plasma Electronics: Applications in Microelectronic Device Fabrication," CRC Press (2006).

[11] A. Quraishi and I. Kronhaus, "Modelling of capillary capacitively coupled radio frequency thruster," *Journal of Physics D: Applied Physics* **54** (2021) 425201.

[12] Z. Donkó, A. Derzsi, M. Vass, B. Horváth, S. Wilczek et al., "eduPIC: an introductory particle based code for radio-frequency plasma simulation," *Plasma Sources Science and Technology* **30** (2021) 095017.

[13] Z. Donkó, A. Derzsi, I. Korolov, P. Hartmann, S. Brandt et al., "Experimental benchmark of kinetic simulations of capacitively coupled plasmas in molecular gases," *Plasma Physics and Controlled Fusion* **60** (2018) 014010.

[14] M. M. Turner, A. Derzsi, Z. Donkó, D. Eremin, S. J. Kelly, "Simulation benchmarks for low-pressure plasmas: Capacitive discharges," *Physics of Plasmas* **20** (2013) 013507.

[15] V. A. Godyak, R. B. Piejak, and B. M. Alexandrovich, "Measurement of electron energy distribution in low-pressure RF discharges," *Plasma Sources Science and Technology* **1** (1992) 36–58.

[16] M. Surendra, "Radiofrequency discharge benchmark model comparison," *Plasma Sources Science and Technology* **4** (1995) 56.

[17] R. W. Hockney and J. W. Eastwood, "Computer Simulation Using Particles," CRC Press (1988).

[18] C. K. Birdsall and A. B. Langdon, "Plasma Physics via Computer Simulation," CRC Press (2004).

[19] C. K. Birdsall, "Particle-in-cell charged-particle simulations, plus Monte Carlo collisions with neutral atoms, PIC-MCC," *IEEE Transactions on Plasma Science* **19** (1991) 65–85.

[20] Z. Donkó, "Apparent secondary-electron emission coefficient and the voltage-current characteristics of argon glow discharges," *Physical Review E* **64** (2001) 026401.

[21] D. Mariotti, J. A. McLaughlin, and P. Maguire, "Experimental study of breakdown voltage and effective secondary electron emission coefficient for a micro-plasma device," *Plasma Sources Science and Technology* **13** (2004) 207.

[22] H. C. Kim, F. Iza, S. S. Yang, M. Radmilović-Radjenović, and J. K. Lee, "Particle and fluid simulations of low-temperature plasma discharges: benchmarks and kinetic effects," *Journal of Physics D: Applied Physics* **38** (2005) R283.

[23] I. Korolov, A. Derzsi, Z. Donkó, E. Schüngel, and J. Schulze, "The influence of electron reflection/sticking coefficients at the electrodes on plasma parameters in particle-in-cell simulations of capacitive radio-frequency plasmas," *Plasma Sources Science and Technology.* **25** (2016) 015024.

[24] A. Sun, M. M. Becker, and D. Loffhagen, "PIC/MCC simulation of capacitively coupled discharges in helium: boundary effects," *Plasma Sources Science and Technology* **27** (2018) 054002.

[25] V. Vahedi and M. Surendra, "A Monte Carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges," *Computer Physics Communications* **87** (1995) 179–198.

[26] M. Vass, P. Palla, and P. Hartmann, "Revisiting the numerical stability/accuracy conditions of explicit PIC/MCC simulations of low-temperature gas discharges," *Plasma Sources Science and Technology* **31** (2022) 064001.

[27] A. Okhrimovskyy, A. Bogaerts, and R. Gijbels, "Electron anisotropic scattering in gases: A formula for Monte Carlo simulations," *Physical Review E* **65** (2002) 037402.

[28] M. Surendra,D. B. Graves, and G. M. Jellum, "Self-consistent model of a direct-current glow discharge: Treatment of fast electrons," *Physical Review A* **41** (1990) 1112–1125.

[29] Z. Donkó, "Particle simulation methods for studies of low-pressure plasma sources," *Plasma Sources Science and Technology* **20** (2011) 024001.

[30] K. Nanbu, "Probability theory of electron-molecule, ion-molecule, molecule-molecule, and Coulomb collisions for particle modeling of materials processing plasmas and cases," *IEEE Transactions on Plasma Science* **28** (2000) 971–990.

[31] K. Koura, "Null-collision technique in the direct-simulation Monte Carlo method," *The Physics of Fluids* **29** (1986) 3509–3511.

[32] M. J. Brennan, "Optimization of Monte Carlo codes using null collision techniques for experimental simulation at low E/N," *IEEE Transactions on Plasma Science.* **19** (1991) 256–261.

[33] S. L. Lin and J. N. Bardsley, "Optimization of Monte Carlo codes using null collision techniques for experimental simulation at low E/N," *Journal of Chemical Physics* **66** (1977) 435–445.

[34] A. V. Phelps, "The application of scattering cross sections to ion flux models in discharge sheaths," *Journal of Applied Physics* **76** (1994) 747–753.