Application of sparse grid combination techniques to low temperature plasmas particle-in-cell simulations. I. Capacitively coupled radio frequency discharges

Laurent Garrigues, B. Tezenas Du Montcel, G. Fubiani, F. Bertomeu, F. Deluzet, Jacek Narski

Application of Sparse grid combination techniques to low temperature plasmas Particle-In-Cell simulations. Part 1: Capacitively coupled radio frequency discharges

L. Garrigues[1,*] B. Tezenas du Montcel[1], G. Fubiani[1], F. Bertomeu[1,2], F. Deluzet[2], and J. Narski[2]

[1] LAPLACE, Université de Toulouse, CNRS, 31062 Toulouse, France.

[2] Université de Toulouse ; UPS, INSA, UT1, UTM, Institut de Mathématiques de Toulouse, CNRS, Institut de Mathématiques de Toulouse, UMR 5219, 31062 Toulouse, France

The use of Particle-In-Cell (PIC) algorithm with explicit scheme to model low temperature plasmas is challenging due to computational time constrains related to resolving both the electron Debye length in space and the inverse of a fraction of the plasma frequency in time. One recent publication [Ricketson and Cerfon, Plasma Phys. Control. Fusion **59**, 024002 (2017)] has demonstrated the interest of using a sparse grid combination technique to accelerate explicit PIC model. Simplest plasma conditions were considered. This paper is the demonstration of the capability and the effectiveness of the sparse grid combination technique embedded in the PIC algorithm (hereafter called "sparse PIC") to self-consistently model capacitively coupled radio frequency discharges. For two-dimensional calculations, the sparse PIC approach is shown to accurately reproduce the plasma profiles as well as the energy distribution functions compared to the standard PIC model. The plasma parameters obtained by these two numerical methods differ by less than 5 % while a speed up in the executable time between 2 and 5 is obtained depending on the set-up.

[*] Corresponding author; email: laurent.garrigues@laplace.univ-tlse.fr

I. Introduction

Particle-In-Cell (PIC) simulations provide a general description of plasmas through a kinetic description of charged particles. They have been extensively described in text books [1-3] and review articles (e.g. [4-8]). In the domain of low temperature plasmas, they are preferably used when the mean free path of electrons becomes larger or in the order of the size of the plasma discharge. They are widely employed in the context of capacitive [9-12], inductively coupled [13], magnetron [14-15], magnetic field barrier [16-20], nanosecond pulse [21-22], and streamer [23-25] discharges.

An explicit scheme that uses charged particle properties at previous time step to calculate electromagnetic field characteristics at next time step is rather straightforward but time consuming since the smallest electron scales in space and time must be resolved for stability reasons. Consequently, severe constrains exist on the grid spacing Δx (that must be smaller than the electron Debye length $\lambda_{D,e}$) and the time step Δt (smaller than a fraction of the inverse electron plasma frequency $\omega_{p,e}$), avoiding numerical heating [1], [17]. Note that the Courant-Friedrichs-Lewy (CFL) condition is naturally satisfied when both constrains are respected. For a typical discharge size of 5 centimeters length during 10 μs with an electron density and a temperature of 10^{18} m^{-3} and 3 eV, respectively, the numerical parameters must be small as Δx ~ 10 μm and Δt ~ 1 ps, leading to 5 000 cells in one direction and 10 million of time steps. In 3D, the number of cells increases to 1.25×10^{11}. The use of a finite number of computational particles to statistically sample the charged particle distribution functions can also affect the accuracy of the obtained solution [17]. The addition of a Monte Carlo procedure to treat collisions can deteriorate the plasma properties. To maintain the effect of physical collisions dominant, Okuda and Birdsall [26], and Turner [27] show that the numerical collision frequency must be smaller than $10^{-4}\times\omega_{p,e}$. That fixes a minimum number of particles in a Debye sphere to make negligible the effect of non-physical collisions due to the numerical particle discreteness. Taking for example 100 computational particles per cell and per species, $500\,000^d$ particles per species must be followed, d being the number of direction. Even on massively parallel machines, high plasma densities in three dimensional geometries are difficult to simulate without weeks or months of computational time.

To alleviate the constrains on the numerical parameters, a sub-cycling method can be implemented. It consists in assuming that the motion of the ions may be disregarded during a few time steps while satisfying the constrains related to the electron dynamics [28]. The gain in computational efficiency remains limited (a factor of two when one ion specie is considered). A density scaling method (reducing the initial plasma density by a constant factor larger than 1) has also been proposed. The solution in the quasineutral region is identical, but the sheath description is affected since its thickness is proportional to the electron Debye length and is hence enlarged for a lower plasma density. The limited scaling factor is linked to the fact that the sheath properties can be changed (e.g. becoming collisional – the sheath thickness becomes larger than collisional mean free path, and/or magnetized - the sheath thickness becomes larger than particle gyroradius - if not before). For example, this method has been successfully used to model the operation of the BATMAN source designed to produce and extract negative ions for the neutral beam injector of ITER [16]. A factor of 6×10^4 (~ 10^{13} m^{-3} in the simulation compared to 6×10^{17} m^{-3} in the real source) has been employed in the 3D simulations of the ion source associated with a computational domain of 192 × 128 × 256 grid nodes that makes a simulation running time acceptable (54 days on 6 processors). The adaptive mesh refinement method (together with an adaptive particle management algorithm) has also been considered in 3D models of streamer propagation to reduce computational time [23].

In the context of the acceleration of PIC simulations, first implicit methods have been implemented in the 80's, in a time where the architecture of processors was not able to perform calculations in parallel. They are based on the calculations of electromagnetic fields and particle motions at the same time. Stability constrains (and associated discretization parameters) can be relaxed when the resolution of the electron Debye length and plasma frequency are not necessary to capture long-wavelength and low-frequency phenomena (plasma expansion, instabilities, MHD, etc.). The issue stems from the non-linear coupling of particles and field equations. The difficulty is overcome by a linearization of the system of equations introducing fluid moments - momentum and mass – as in the Implicit Moment Method [29], or through a linearization of the shape functions used for the deposition of charges and current densities onto the grid in the Direct Implicit Method [30]. Most of the time, constrains in time step and grid spacing are relaxed by a factor of ~ 10. Nevertheless, the predictor-corrector scheme in the Direct Implicit Method requires solving two times the pusher of particles, the most expansive procedure of the PIC algorithm. Moreover, the matrix providing the electric field is time dependent, requiring at each time step a costly matrix factorization if a direct solver is used. A Fast Fourier Transform method can advantageously be used when periodic boundary conditions are employed [31]. Finally, fully implicit methods have recently been addressed by Chen and co-workers [32] and Markidis and Lapenta [33] for both electrostatic and electromagnetic PIC models. The particle equations of motion and the field equations system are differenced implicitly and non-linearly solved by means of a sequence of linear problems. The algorithm is stable for any choice of the time step irrespective to the electron stability constrains as in the explicit method. The particle pusher is however sophisticated. Since particles may cross several cells in a time step, their motion needs to be decomposed by cells in order conserve the total energy in the system. The efficiency of fully implicit method is limited by that of the preconditioner used to invert the matrix [34].

Another critical issue related to the number of computational particles required to properly sample the physical system is the question of computer memory usage. Obviously, a one dimensional PIC model with for instance a total of 500 000 macroparticles per species is easy to manage on a computer in term of memory. This number grows in 2D and 3D so that the total number of particles can become large enough to exceed the maximum memory resources available on the even most performant computers. Ricketson and Cerfon have proposed to apply the so called "sparse grid" combination method to accelerate PIC simulations by reducing the number of cells in the simulation domain and consequently the total number of macroparticles [35]. An increase of the grid cell sizes allows to reduce the statistical noise in the simulation without increasing the total particle number They have verified this new approach with three benchmarking tests encountered in plasma physics (linear and nonlinear landau damping in 3D and diocotron instability in 2D). These benchmarks remain limited with regard to different characteristics generally encountered in plasmas. Purely periodic boundary conditions are systematically applied, with ions at rest defining a neutralizing background, and collisions are disregarded. Our purpose is to assess the merit of the sparse grid PIC-MCC technique in the context of low temperature plasma calculations. This includes the simulation of the motions of all charged particles (including ions), collisions (including ionization to self-sustain the discharge), as well as non-trivial boundary conditions such as metallic walls. In this article (part one), the results derived from the sparse grid method are compared to the plasma profiles provided by the standard PIC-MCC algorithm for configurations typically encountered in representative computations of capacitive discharges. A work of verification between different PIC models has already been addressed for one dimensional simulation of capacitive discharges [10]. These results will serve as reference cases for the benchmark investigated within the present work. In the companion paper (part two), the same verification will be performed in an E × B magnetized discharge [36]. The rest of the paper is organized as follows. Section II is devoted to the adaptation of the sparse grid combination technique in the context of PIC-MCC models. In section III, verification and estimation of the limit of the sparse PIC method will be addressed for capacitive discharges. In section IV, the gain in computational efficiency is outlined. Main results are finally summarized in section V.

II. Sparse grid combination technique applied to PIC algorithm

Sparse grid methods aim at reducing the complexity of discrete problem by breaking the exponential increase of the number of degrees of freedom with respect to the dimensionality of the problem. For classical discretization on uniform Cartesian grid meshes, with M degree of freedom in each direction, the total number of unknowns is indeed proportional to M^d, d being the dimensionality of the problem. The purpose of sparse grid techniques is to compute approximation of a function on a hierarchy of anisotropic grids with a coarser resolution, hence reducing the total number of degrees of freedom. A combination of approximation carried out on the different anisotropic grids allows the reconstruction of the solution of the initial Cartesian mesh with a comparable approximation to a classical interpolation performed directly from the M^d degrees of freedom carried out by a Cartesian grid [37]-[39]. For standard piecewise linear interpolation functions, the number of degrees of freedom of sparse grids is decreased to $M\log(M)^{d-1}$ while the precision of the reconstructed numerical approximation is proportional to $h^2\log(M)^{d-1}$ (in L_2 and L_∞ norms). This means that the second order precision expected on a Cartesian grid with a mesh h is recovered with a significant gain regarding the computational efficiency for problems with a large dimensionality ($d > 1$).

Sparse grid methods have been used in fluid mechanics for the resolution of Navier-Stokes equation [40-41], quantum mechanics for the resolution of Schrödinger equation [42], plasma physics for the resolution of the gyrokinetic Vlasov equation [43], [44], and financial mathematics for the resolution of the Black-Scholes equation [45]. In sub-section A, we recall the basics of the sparse grid combination technique and we refer to Ref. [34] for the background (and references therein for more details about the mathematics). In sub-section B, the standard PIC algorithm is enhanced to take benefits from the sparse grid combination technique. In sub-section C, elements of parallelization and optimization are outlined.

A. Grid definition and combination technique

Different sparse grids may be defined (see [46] for more details), this work being devoted to the combination technique, only the sparse grids specific to this class of numerical approximation are detailed in the sequel. The number of cells in each direction is assumed to be defined as $M = 2^N$ with a corresponding number of nodes equals to $2^N + 1$. For simplicity, the description of the method will be specified for a two-dimensional problem (*x* and *y* directions) and a Cartesian initial grid with a uniform mesh width in both directions $h_x = h_y = h = 2^{-N}$. Let us denote $H_{I,J}$ the Cartesian grid with depth I and J and mesh widths $h_x^I = 2^{-I}$ and $h_y^J = 2^{-J}$, where I and J are strictly positive. The sparse grid approximation U_N on the regular Cartesian grid $H_{N,N}$ can be obtained via a linear combination of numerical approximations $U_{I,J}$ computed on the coarser meshes $H_{I,J}$ [47] with:

$$U_N = \sum_{I+J=N+1} U_{I,J} - \sum_{I+J=N} U_{I,J} \quad (1).$$

As mentioned before, the recombined numerical solution provides an approximation with an accuracy comparable to that of obtained on uniform Cartesian mesh. Specifically, the error of sparse grid approximation is proportional to $h^2 \log(M)^{d-1}$ to be compared to h^2 for standard piecewise linear interpolation on a grid with a mesh size h. This marginal deterioration of the precision is balanced by a substantial improvement of computational efficiency. The combination technique becomes very advantageous for large depth N. In sparse grids, the total number of cells G_{sparse} summing all the cells for which $I + J = N + 1$ and $I + J = N$ (for 2D problems) is equal to:

$$G_{\text{sparse}} = N2^{N+1} + (N-1)2^N = 2^N(3N-1) \text{ (2)},$$

growing slower with N than $G_{\text{r}} = 2^N \times 2^N$ the total number of cells for a regular grid. The ratio $G_{\text{r}}/G_{\text{sparse}}$ as a function of depth N is plotted in figure 1. For N = 9 (M = 512), the reduction of the number of cells is close to 20.

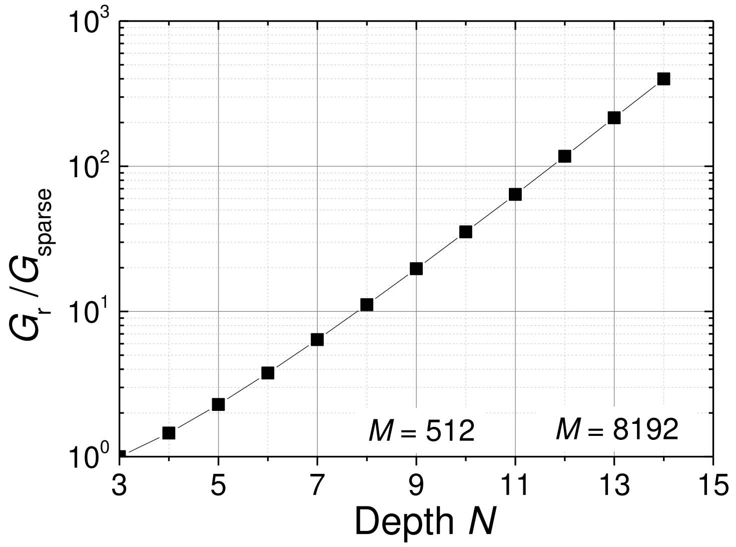


Figure 1: Total number of cells for the Cartesian mesh (with a mesh size 2^{-N}) divided by its sparse grids hierarchy counterpart $G_{\text{r}}/G_{\text{sparse}}$ as a function of the depth N.

B. Sparse grid PIC algorithm

In the standard electrostatic PIC algorithm, particles are defined in phase-space through positions and velocities while the electric field components (and applied magnetic field components if any) are defined on the nodes of the computational grid [1-5]. Assuming at the beginning of the Δt cycle a given profile of electric field (and eventually magnetic field), new positions and velocities of particles have to be calculated. Since particles are defined continuously in space while electromagnetic fields being defined at discrete locations, an interpolation is used to assign the electromagnetic force acting on particles at their positions. The implementation of a center leap-frog scheme with the Boris algorithm [2] to advance particles involves the use of an offset of $\Delta t/2$ between position and velocity. Boundary conditions are applied to particles (losses and/or emission at the walls). If collisions occur in the discharge volume, a Monte Carlo procedure is added in the cycle. The most efficient in computation time is the null collision method [5]. To displace the particles under the new electrical force acting on them, the electric potential profile is computed from Poisson's equation (Gauss's Law $\varepsilon_0 \nabla . \mathbf{E} = \rho$ combined with the scalar definition of electric potential $\emptyset$, $\mathbf{E} = -\nabla\emptyset$):

$$\Delta\emptyset = -\frac{\rho}{\varepsilon_0} = -\frac{1}{\varepsilon_0}\sum_\alpha \rho_\alpha \text{ (3),}$$

where ρ is the net charge density calculated from the sum over all the charge densities ρ_α of specie α. In equation (3), $\emptyset$ is calculated on the grid nodes from the source term ρ. The latter is evaluated on a given grid node using the charge carried by the macroparticles in the neighboring cells. Classically, the charge of a particle is weighted to the nearest nodes via the shape functions. These are called b-spline functions and correspond to different orders of precision (order 0, the charge is assigned on the nearest grid point), order 1 (with a linear spline to the closest grid points), order 2 (quadratic splines), etc. The most common function used in the standard PIC method in 2D is a piecewise bilinear spline shape (also called Cloud In Cell).

Following the formal analysis conducted by Ricketson and Cerfon [35] a sparse grid approximation of charge density$\rho_{N,\alpha}$ (α denoting the particle species) may be reconstructed on the regular grid using Eq. (1):

$\rho_{N,\alpha} = \sum_{I+J=N+1} \rho_{I,J,\alpha} - \sum_{I+J=N} \rho_{I,J,\alpha}$ (4).

Note that computing the electrostatic potential thanks to eq. (3) with $\rho_{N,\alpha}$ as a source term requires the numerical resolution of the Poisson's equation on the initial mesh (with hence a large number of cells) and is therefore as time consuming as the standard PIC method. We have hence adopted a different strategy (also detailed in ref. [35]) to benefit from the advantage of the sparse grid combination technique by resolving Poisson's equation on each of the $2N-1$ sub-grids ($2N-1$ corresponding to the total number of sub-grids that are constructed from a depth Nin the maximum-norm based grid). Due to the reduction of the size of linear systems solved on each of the sparse grids, a substantial gain in numerical efficiency is achieved. The electric field at the particle position is calculated using Eq. (1) directly recombined from combination from electric fields calculated by linear interpolation on each of the $2N-1$ sub-grids. The cycle of calls to subroutines in the sparse PIC algorithm is sketched in figure 2. The initialization, integration of equations of motions, boundary conditions and collisions are strictly identical to the ones of the standard PIC approach. For Dirichlet boundary conditions, the electric field component normal to the walls is obtained from a linear extrapolation of electric field component at forward points [17]. As previously mentioned, the charge deposition (in order to calculate $\rho_{I,J,\alpha}$) together with Poisson's equation routine are calculated on each of the sub-grids. The calculation of Poisson's equations has also to be performed on each of the sub-grids, while the electric field is reconstructed at the particle location by combination.

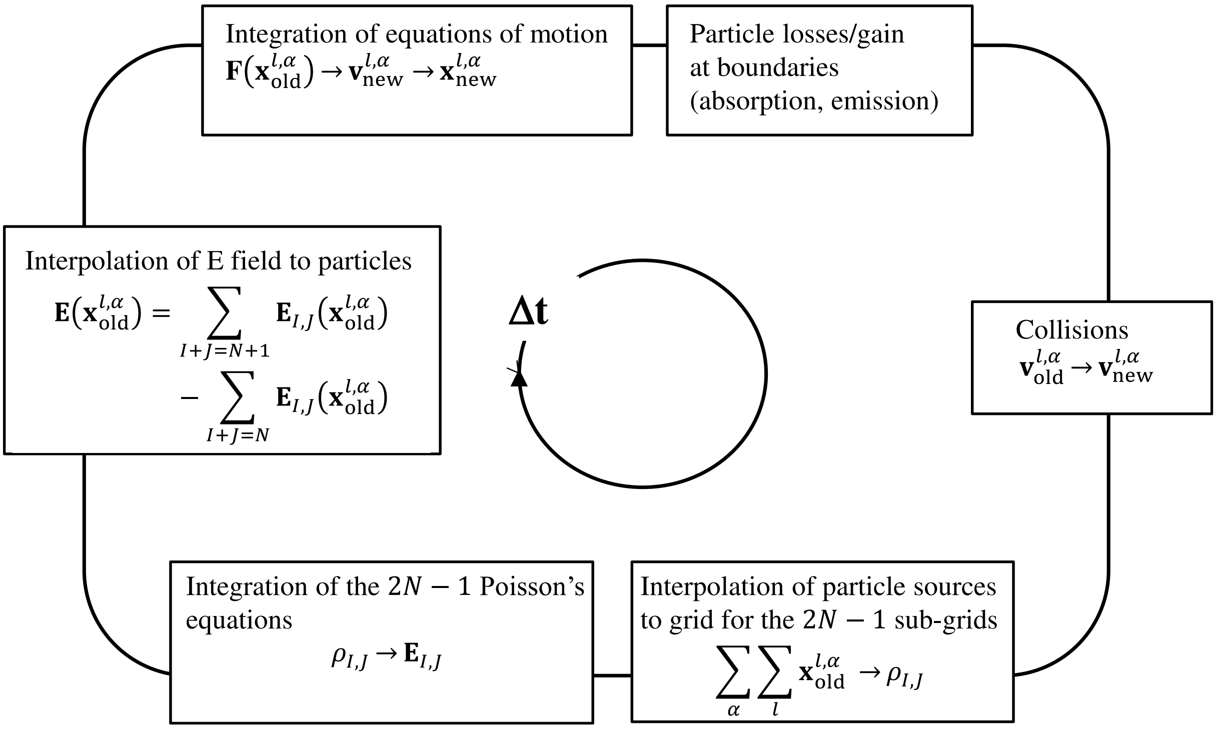

Integration of equations of motion
$\mathbf{F}(\mathbf{x}_{\mathrm{old}}^{l,\alpha}) \rightarrow \mathbf{v}_{\mathrm{new}}^{l,\alpha} \rightarrow \mathbf{x}_{\mathrm{new}}^{l,\alpha}$
Particle losses/gain at boundaries (absorption, emission)
Collisions
$\mathbf{v}_{\mathrm{old}}^{l,\alpha} \rightarrow \mathbf{v}_{\mathrm{new}}^{l,\alpha}$
Interpolation of particle sources to grid for the $2N-1$ sub-grids
$\sum_{\alpha}\sum_{l} \mathbf{x}_{\mathrm{old}}^{l,\alpha} \rightarrow \rho_{I,J}$
Integration of the $2N-1$ Poisson's equations
$\rho_{I,J} \rightarrow \mathbf{E}_{I,J}$
Interpolation of E field to particles
$\mathbf{E}(\mathbf{x}_{\mathrm{old}}^{l,\alpha}) = \sum_{I+J=N+1} \mathbf{E}_{I,J}(\mathbf{x}_{\mathrm{old}}^{l,\alpha}) - \sum_{I+J=N} \mathbf{E}_{I,J}(\mathbf{x}_{\mathrm{old}}^{l,\alpha})$
$\Delta\mathbf{t}$

Figure 2: Sparse PIC algorithm. The indices l and α refer to a given particle l of specie α.

C. Parallelization and optimization strategies

As in standard PIC simulations, we have adopted a hybrid parallelization taking the advantage of distributed memory (with Message Passing Interface (MPI) libraries) between cores and shared memory (using Open Multi-Processing (OpenMP) programming with FORTRAN) between threads [16-17], [20]. Macroparticles are ascribed initially equally between MPI and OpenMP threads (so called "particle decomposition technique"); the entire simulation domain is accessible by each threads. A sorting algorithm is implemented where macroparticles are ordered per grid cell, allowing a sequential access to the computer memory in the pusher subroutine [48]. Calling the sorting algorithm each between 10 to 50 iterations in the PIC cycle is a good compromise between gain (with respect to memory access) and losses (time devoted to the sorting) in computing time.

The Single Instruction Multiple Data (SIMD) capability of modern architecture processors offers the opportunity of performing vectorization, meaning that, for data stored contiguously in memory, the processor is able to perform an arithmetic operation to multiple operands at once. Vectorization does not demand supplementary instructions to push particles because additions and multiplications are performed by accessing different arrays but with the same index. This is no longer the case in the calculations of the charge densities or electric-field at particle position since different indexes - corresponding to nodes $(i,j),(i+1,j),(i,j+1)$, $(i+1,j+1)$, around the cell (i,j) - must be accessed. The charge deposition and electric field interpolation routines have been rewritten from the scalar to vectorized version (through the SIMD directive provided from OpenMP 4.0), as proposed by Vincenti *et al.* [49]. To improve the vectorization efficiency, the particle list is divided into blocks of 64 to 256 particles. Operations are performed block by block instead of elements by elements to optimize the cash reuse [49-50]. This technique has not been used for the standard PIC calculations since the time passed in those routines is negligible (see below).

In this work, the elliptic Poisson's equation is solved with the PARDISO (Parallel Sparse Direct Linear Solver) subroutine included in the Intel®Math kernel Library (Intel®MKL) [51]. PARDISO is an easy-to-use direct solver based on a LU decomposition. Explicit PIC schemes takes the advantage of time-independent matrix coefficients in the Poisson solver and hence a unique factorization must be done only once during initialization. In the sparse grid PIC algorithm, $2N-1$ matrixes are initialized and solved in parallel using the threads on core 0. Each thread within core 0 is assigned to the resolution of one of the $2N-1$ Poisson's equations. The computational time required for the resolution of the Poisson system is therefore proportional to the size of the largest system carried out by the sparse grids: 2^{N+1} (rather than 2^{2N}). The matrix factorization is proportional to $2^{3(N+1)/2}$ while the resolution of the triangular systems stemming from the factorization is proportional to $(N+1)2^{N+1}$ and is hence insignificant [52]. In total, the time passed to solve Poisson's equation becomes negligible (see below).

III. Sparse PIC modeling of capacitive discharges

Applying sparse PIC techniques in the context of capacitive discharges allows us to explore the potentiality and the limits of the method when sheaths are present (this not done in Ref. [35]). The capacitive coupled discharges also offer the opportunity to test boundary conditions varying in time, and collisions (including ionization to self-sustain the discharge). We have taken the conditions of the benchmark published by Turner and co-workers [10] as reference for simulations set-up and for comparing results. Calculations have also the convenience to be representative of “real” experimental conditions. In sub-section A, conditions and benchmark description are recalled. Results are analyzed in sub-section B.

A. Input data and benchmark description

The original work of Turner defines a one-dimensional discharge along the x direction with a distance between the two electrodes of $L_x = 6.7$ cm. For two cases, we use periodic boundary conditions in the y direction to model the 1D benchmark with a 2D system. The neutral gas is helium at a temperature of 300 K, and the neutral density is uniform and fixed in space and in time. The left electrode is grounded, while the voltage varies sinusoidally at a frequency of 13.56 MHz on the right electrode ($V = 0$ at the beginning of the simulation). Particles interacting with electrodes are absorbed, and secondary electrons emitted under charged particle impacts on the electrodes are not considered. Along the periodic direction, a particle whose y coordinate crosses L_y (respectively 0) has a new y coordinate equal to $y - L_y$ (respectively $y + L_y$) with same velocity components. In a third case, a real 2D system is modeled applying zero potential on walls (grounded electrodes) except on the right electrode where the voltage varies sinusoidally at a frequency of 13.56 MHz. For that specific case, all charged particles interacting with electrodes are absorbed (secondary emission is also not considered).

The discharge is composed of electrons and singly charged He ions. Collisions occur between charged particles and He neutral gas. For electron-neutral collisions, the cross sections compiled by Biagi [53] are considered. They are composed of elastic momentum, one ionization and two excitation levels cross sections. After ionization, the residual energy is equidistributed between the primary and secondary electrons. After collisions, isotropic scattering in the center of mass are performed. For ion-neutral collisions, cross sections are taken from the work of Phelps [54], they include isotropic and backward scattering components (both in the center of mass frame). To avoid any difference in the simulation results due to varying functions used to fit cross sections, reference 1 of the work of Turner *et al.* [10] provides the electronic files of the tabulated cross sections. For intermediate values, a linear interpolation is used. The time step and grid spacing have been chosen to resolve the electron plasma frequency and electron Debye length, respectively, to fulfill the stability criteria of the explicit PIC method, as discussed in introduction. The time step for collisions is the one used for the transport of particles. Initially, an identical number of ions and electrons representing a uniform plasma density are introduced in the computational domain. Velocities of charged particles are sampled from a Maxwellian distribution at a given temperature [17]. Initial conditions and parameters are specified in Table 1.

In the original study of Turner and co-workers, four cases varying neutral densities and the voltage amplitudes have been examined. We have chosen to illustrate two of the four cases (Case 1 and Case 4 in [10], respectively named Case A and Case B in our study) plus a third case (Case C) whose conditions are close to Case A. Our choice is dictated by the analysis of the modeling capabilities of the sparse PIC method in two different situations: a case where the plasma density is low, the sheath thickness large and the electron energy distribution (EEDF) close to a Maxwellian, and a case where the plasma density is large, the sheath thickness thin and the EEDF far from a Maxwellian. For Case B, to limit computational time with the standard PIC method, the number of grid points along y has been taken to 4 (and L_y = 0.05234 cm, accordingly to maintain Δx = Δy). The main difference between Cases A and C is related to boundary conditions as mentioned earlier.

Physical parameters		Case A	Case B	Case C
Electrode distance along x	L_x (cm)	6.7	6.7	6.7
Distance along y standard/sparse PIC	L_y (cm)	6.7/6.7	0.05234/6.7	6.7/6.7
Neutral density	N (10^{20} m^{-3})	9.64	321	19.28
Neutral temperature	T_N (K)		300	
Frequency	f (MHz)		13.56	
Voltage	V (V)	450	120	150
Physical constants		Case A	Case B	Case C
Electron mass	m_e (10^{-31} kg)		9.101	
Ion mass	m_i (10^{-27} kg)		6.67	
Initial conditions		Case A	Case B	Case C
Plasma density	n_0 (10^{14} m^{-3})	2.56	3.84	2.56
Electron temperature	T_e (K)		30 000	
Ion temperature	T_i (K)		300	
Initial parameters (regular grid)		Case A	Case B	Case C
Grid spacing - standard PIC	$\Delta x = \Delta y$	L_x/128=L_y/128	L_x/512=L_y/4	L_x/128=L_y/128
Grid spacing - sparse PIC	$\Delta x = \Delta y$	L_x/128=L_y/128	L_x/512=L_y/512	L_x/128=L_y/128

Time step	Δt (s)	$(400\,f)^{-1}$	$(1600\,f)^{-1}$	$(400\,f)^{-1}$
Number of particles per cell	N_{pc}	512	64	512
Boundary conditions		Case A	Case B	Case C
Left wall		Grounded	Grounded	Grounded
Right wall		RF	RF	RF
Top wall		Periodic	Periodic	Grounded
Bottom wall		Periodic	Periodic	Grounded

Table 1: Physical and numerical initial parameters for the benchmark.

B. Results

For each of the two cases, we have performed 2D calculations for the standard and sparse PIC algorithms and calculate the numerical error (the accuracy compared to the reference case) for profiles and the electron energy distribution function. The so-called reference cases correspond to conventional 2D PIC calculations with periodic boundary conditions simulating the configuration of the one-dimensional benchmarks of Tuner and co-workers (indicated in Table 1) for Cases A and B. In Case C, a real 2D problem is resolved. In the rest of the paper, due to periodic conditions along the *y* direction, only one-dimensional profiles of quantities along the *x* direction at mid-distance along *y* will be shown for Cases A and B. We have decided to focus our comparisons on the ion density and ionization profiles, and on the electron energy distribution functions. Physical and numerical characteristics in the center of the discharge are also indicated. Benchmark results of Turner *et al.* [10] will also be recalled. After convergence, for Case C, two-dimensional profiles of ion density and electron energy distribution functions will be shown, as well as plasma characteristics in the center of the discharge. Cases A and C include a large number of particles-per-cell (N_{pc}) and consequently one can test the influence of the total number of particles in the system in the Sparse PIC method. Case B (higher plasma density) requires a more refined grid. Since stability constrains have been taken with a margin $\Delta x/\lambda_{D,e}$~0.5 (see Table 2), it offers an opportunity to investigate different depths of sparse grids. To calculate the accuracy, we apply the method of Ricketson and Cerfon [35] by calculating for a quantity φ the error $\epsilon(\varphi)$ through the following relation:

$$\epsilon(\varphi) = \frac{\|\varphi - \varphi_{\text{ref}}\|_{L^2}}{\|\varphi_{\text{ref}}\|_{L^2}} = \sqrt{\frac{\int|\varphi - \varphi_{\text{ref}}|^2 \text{du}}{\int|\varphi_{\text{ref}}|^2 \text{du}}} \quad (5),$$

where φ_{ref} is the reference solution taken from the 2D standard PIC calculations. Integrals whose elements are equally spaced are taken in space or energy.

The first set of results corresponds to Case A. 128 cells on the regular grid indicates a depth of N = 7. The number of sub-grids in the sparse grids hierarchy is equal to 13. The simulations with the regular grid have been performed with N_{pc} = 512, as in 1D benchmark, leading to N_{T} = 8.38×10^6 particles in total. We have performed sparse PIC simulations with a total number of particles ranging initially from 2.57×10^5 to 8.38×10^6 which corresponds to N_{pc} between 100 and 3280. The case with N_{pc} = 3280 is chosen keep the total number of particles identical to the one in the standard 2D PIC calculations. In figure 3, time-averaged axial profiles of ion density and ionization rates are shown. For comparison, physical and numerical characteristics in the center of the discharge are indicated in Table 1. The time and space averaged EEDF (represented by the normalized probability distribution function and plotted in such a way that a Maxwellian distribution would appear as a straight line) is shown figure 4.

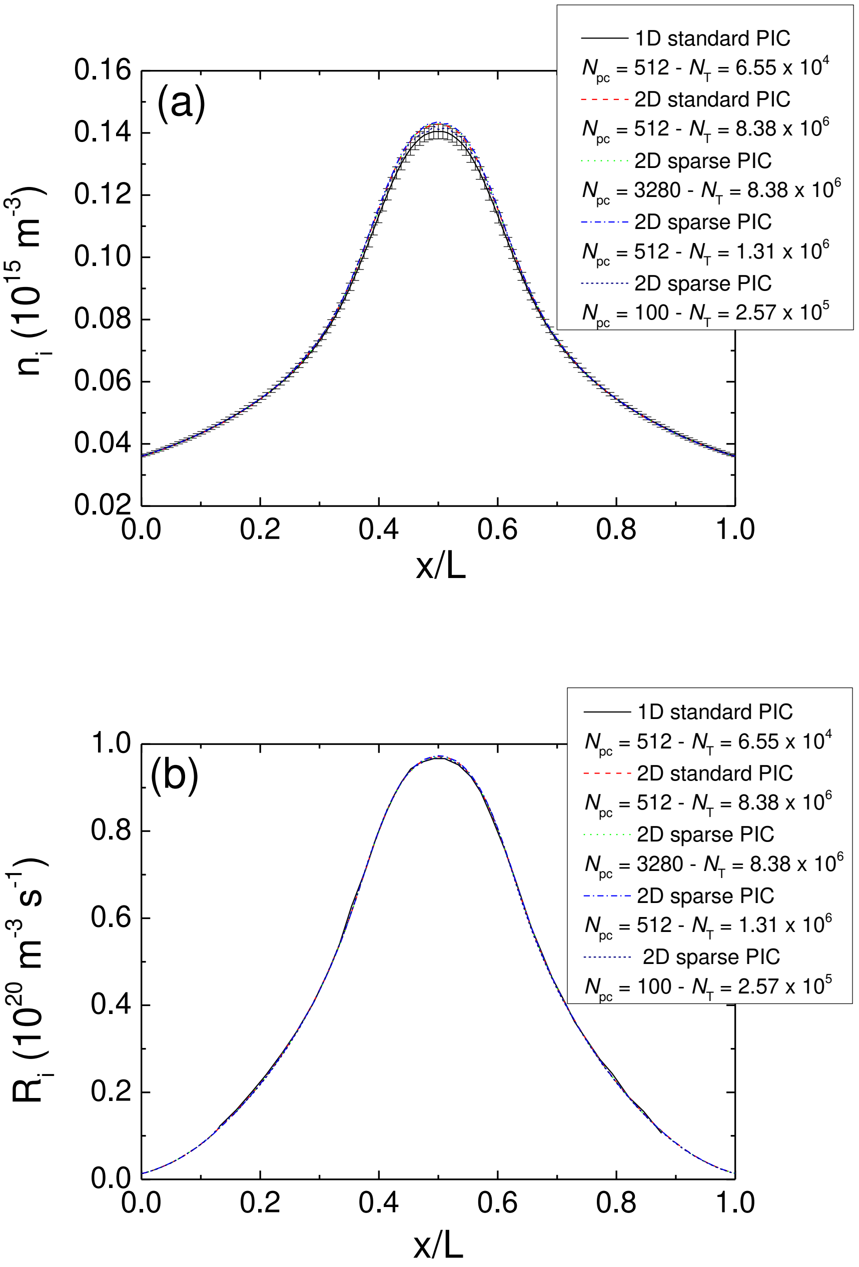


Figure 3: Test-Case A: Time-averaged axial profiles at mid-distance along y of (a) ion density n_i and (b) ionization rate R_i at steady state. One-dimensional results are taken from Ref. [10]. The error bars indicate the standard deviation from an extended study presented in Ref. [10]. For each computational case, the initial total number N_{T} and the number of particles per cell N_{pc} are indicated. Sparse PIC results correspond to N = 7.

Physical and numerical characteristics

	standard 1D	standard 2D	sparse 2D	sparse 2D	sparse 2D
	$N_{\mathrm{pc}} = 512$	$N_{\mathrm{pc}} = 512$	$N_{\mathrm{pc}} = 3280$	$N_{\mathrm{pc}} = 512$	$N_{\mathrm{pc}} = 100$
n_i (10^{15} m^{-3})	0.140	0.142	0.139	0.143	0.141
$k_B T_e$ (eV)	9.36	8.98	8.86	8.85	8.96
$\Delta x/\lambda_{D,e}$	0.27	0.28	same resolution as standard grid		
$\omega_p \Delta t$	0.121	0.122	0.121	0.123	0.121

Table 2: Test-Case A: Physical and numerical characteristics at steady state in the center of the discharge. Sparse PIC results correspond to N = 7.

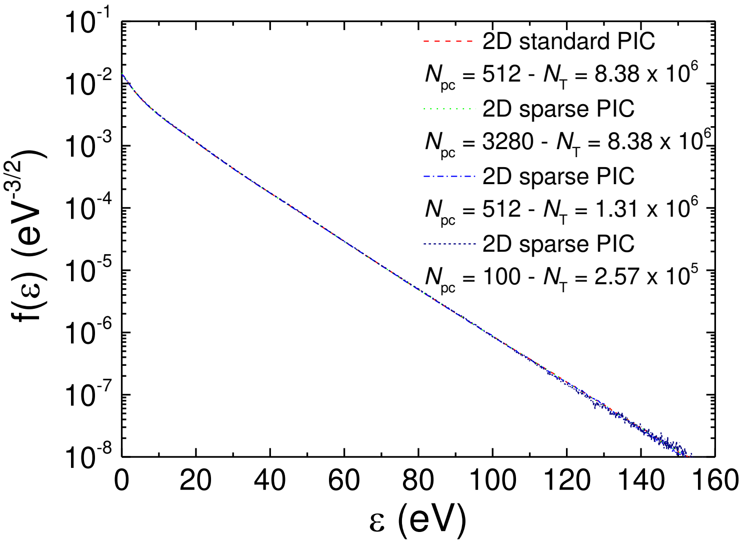


Figure 4: Test-Case A: Time and space averaged energy probability function at steady state. The data are normalized such that $\int_0^\infty \sqrt{\varepsilon} f(\varepsilon) d\varepsilon = 1$. For each computational case, the initial total number N_{T} and the number of particles per cell N_{pc} are indicated. Sparse PIC results corresponds to N = 7.

The comparison between the 1D and 2D standard PIC models results with a regular grid shows similar profiles (almost within the margin obtained by error bar displayed for the 1D case) not only in the center of the discharge but also in the sheath regions. Modeling a one-dimensional discharge with a 2D model using periodic boundary conditions while using the same number of N_{pc} - as in the 1D case - results in a larger numerical noise, but the overall difference is less than a few percent for the plasma parameters. We will not discuss in details the physics interpretation of the discharge as it is out of the scope of this work; note that at low pressure, the EEDF in a capacitively coupled radio frequency discharge is close to a Maxwellian and the electron power dissipation is almost uniform. Additional information can be found in Ref. [10].

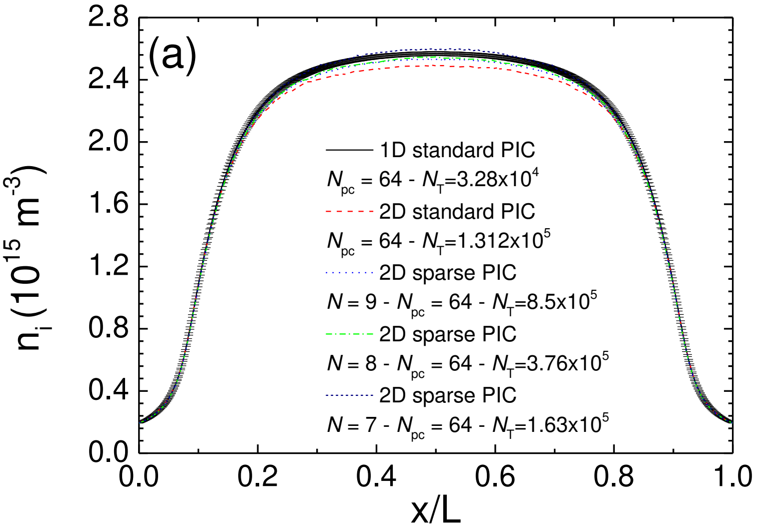
Error (%)			
	sparse 2D	sparse 2D	sparse 2D
	$N_{\mathrm{pc}} = 3280$	$N_{\mathrm{pc}} = 512$	$N_{\mathrm{pc}} = 100$
n_i	0.15	0.4	0.6
R_i	0.12	0.18	0.2
$EEDF$	0.35	0.27	3.2

Table 3: Test-Case A: Differences between standard and sparse PIC results. Sparse PIC results corresponds to $N = 7$.

The close agreement between the two methods is evident when looking at Table 3 where the error (calculated via eq. (5)) considering the 2D PIC model with a regular grid as the reference solution is reported. The time-averaged axial profiles of either ion densities or ionization source terms in figures 3 and 4 are nearly identical. More interestingly, the sparse PIC algorithm is able to capture the EEDF with a large precision up to 6 orders of magnitude. Obviously, a larger difference appears in the tail of the distribution composed of an electron population at extremely high energy (but marginal in numbers). This is related to the total number of particles in the simulation.

One interesting conclusion from that first comparison is that keeping the same number of particle-per-cell N_{pc} between the sparse and standard PIC methods seems a good strategy to keep a high degree of precision. Keeping the same N_{pc} in the sparse PIC algorithm reduces considerably the total number of particles in the system (especially since the depth N is high, see figure 2) and hence the computational time (see discussion in section IV).

We have modeled a second capacitively coupled radiofrequency discharge operating at a pressure higher by a factor 30 (Case B). The latter corresponds to another 1D simulation of Tuner *et al.* [10]. Our simulation is again in 2D with periodic boundary conditions along *y*. Conditions for Case B are listed in Table 1. We have studied the influence of the depth N which we varied from 9 (i.e., 512 regular grid cells with $\Delta x/\lambda_{D,e}$~0.5 see Table 2) to 7 (128 cells and $\Delta x/\lambda_{D,e}$~2), keeping the initial number of particle-per-cell N_{pc} to 64. The number of sparse grids in the hierarchy is either 17 for N = 9 or 13 when N = 7, respectively. As for Case A, we have successively plotted in figures 5 the time-averaged profiles for the ion density and ionization rate along the mid plane of the discharge, as well as the space and time averaged EEDF in Figure 6. Main discharge characteristics in the center of the discharge are given in Table 3. The differences between standard and sparse grids are indicated in Table 4.



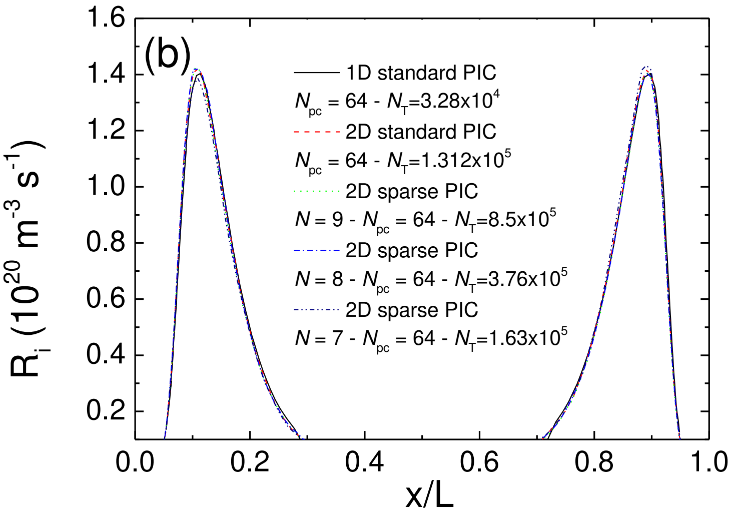


Figure 5: Test-Case B: Time-averaged axial profiles at mid-distance along y for (a) ion density n_i and (b) ionization rate R_i at steady state. One-dimensional results are taken from Ref. [10]. The error bars indicate the standard deviation from an extended study presented in Ref. [10]. Sparse PIC results correspond to depth varying from $N = 9$ to $N = 7$. For each computational case, the initial number of particles per cell N_{pc} is fixed to 64 and the initial total number N_{T} is also given.

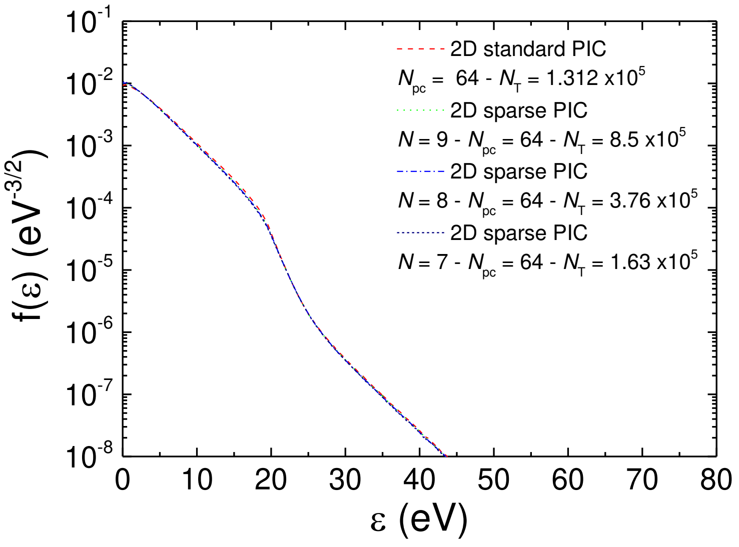


Figure 6: Test-Case B: Time and spaced averaged energy probability function at steady state. The data are normalized such that $\int_0^\infty \sqrt{\varepsilon} f(\varepsilon) d\varepsilon = 1$. For each computational case, the initial

number of particles per cell N_{pc} is fixed to 64 and the initial total number N_{T} is also given. Sparse PIC results correspond to depth varying from N = 9 to N = 7.

Physical and numerical characteristics					
	standard 1D	standard 2D	sparse 2D N = 9	sparse 2D N = 8	sparse 2D N = 7
n_i (10^{15} m^{-3})	2.57	2.50	2.53	2.54	2.59
$k_B T_e$ (eV)	3.65	3.71	3.61	3.60	3.56
$\Delta x/\lambda_{D,e}$	0.47	0.46	~0.5	~1	~2
$\omega_p \Delta t$	0.131	0. 130	0. 131	0. 131	0. 132

Table 4: Test-Case B: Physical and numerical characteristics at steady state in the center of the discharge. The initial number of particles per cell N_{pc} is fixed to 64.

Error (%)			
	sparse 2D N = 9	sparse 2D N = 8	sparse 2D N = 7
n_i	1.3	1.7	3.4
R_i	1.6	1.55	1.6
$EEDF$	3.8	4.6	5.6

Table 5: Test-Case B: Differences between standard and sparse PIC results. The initial number of particles per cell N_{pc} is fixed to 64.

From a general view, we notice different behavior between the two illustrated cases. In Case B, the plasma density is larger and more uniform in the discharge center and the peak of ionization is concentrated in the sheath regions (see figure 5). Additional data from Ref. [10] show that the power absorption and dissipation are more locally balanced in the sheath region in Case B. At a higher pressure, more inelastic processes take place leading to a non-Maxwellian distribution of the EEDF.

The error margin has been obtained at coincident axial grid points using eq. (5) and is reported in Table 5. Obviously, the agreement between the two methods is good (less than 3.5 % difference) for the ion density and ionization profiles. One notable result is that reducing the number of levels (from 9 to 7) keeping N_{pc} the same does not deteriorate two much the error. This, is certainly due to the fact that at the limit where $N = 7$, $\Delta x/\lambda_{D,e}$~2. Similarly the difference in the EEDF profiles is also quasi-independent of N. The advantage in computational time consumption will be discussed in the next section.

Case C corresponds to real 2D conditions with grounded electrodes except on the right electrode where the RF signal is applied. The pressure is two times higher than Case A, maintaining a plasma density close to Case A. A number of grid points of 128 in each direction is capable to resolve the electron Debye length (see Table 6). In figure 7, we illustrate a comparison between standard and sparse PIC methods about the ion density at steady state. The sparse PIC method is able to properly capture the ion density drop in front of the electrodes (as well as the electron density drop, not shown). These results highlight the capability of sparse PIC algorithms to capture the sheaths in 2D complex configuration with a high accuracy. The EEDF calculated with the two algorithms and integrated over all the volume are shown in figure 8. Again, the agreement between the two methods is excellent. The error calculation reported in Table 7 is less than 1.8 % for n_i and 0.1 % for the EEDF for same N_{pc}. Reducing N_{pc} with the sparse PIC approach maintains a high precision in the calculations.

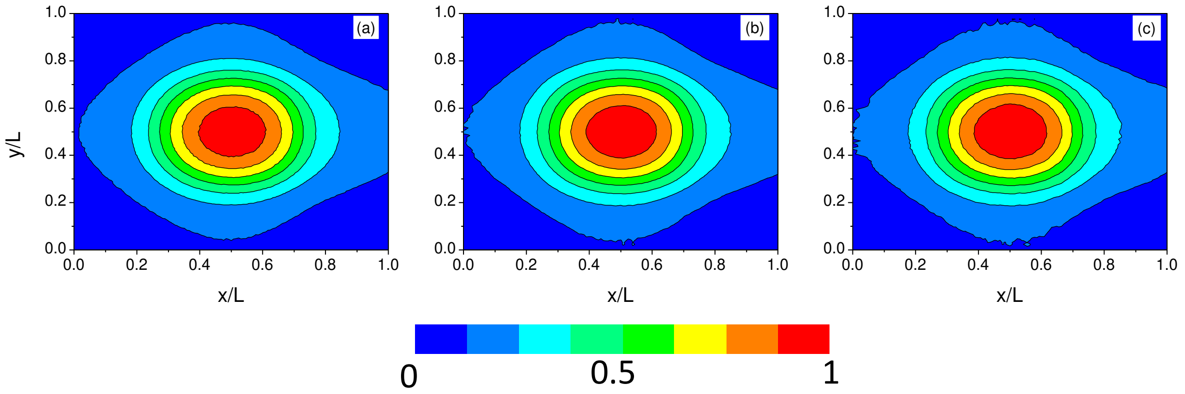


Figure 7: Test-Case C: 2D time-averaged axial profiles of ion density n_i at steady state, (a) standard conditions with N_{pc}= 512, (b) sparse with N_{pc}= 512 and (c) sparse with N_{pc}= 100. Maximum of ion density: 2.2 × 10^{14} m^{-3}. Sparse PIC results correspond to N = 7.

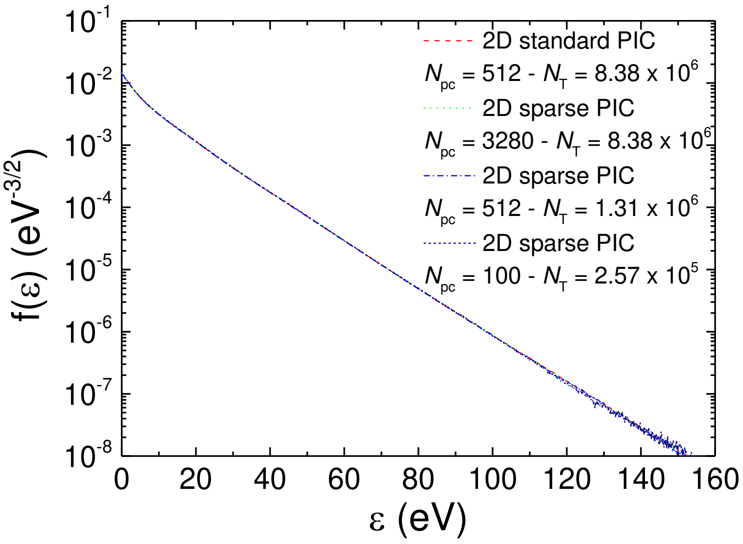


Figure 8: Test-Case C: Time and space averaged energy probability function at steady state. The data are normalized such that $\int_0^\infty \sqrt{\varepsilon} f(\varepsilon) d\varepsilon = 1$. For each computational case, the initial total number N_{T} and the number of particles per cell N_{pc} are indicated. Sparse PIC results corresponds to N = 7

Physical and numerical characteristics

	standard 2D	sparse 2D	sparse 2D	sparse 2D
	N_{pc} = 512	N_{pc} = 3280	N_{pc} = 512	N_{pc} = 100
n_i (10^{15} m^{-3})	0.212	0.213	0.214	0.217
$k_B T_e$ (eV)	6.89	6.90	6.91	6.88
$\Delta x/\lambda_{D,e}$	0.39	same resolution as standard grid		
$\omega_p \Delta t$	0.151	0.151	0.152	0.153

Table 6: Test-Case C: Physical and numerical characteristics at steady state in the center of the discharge. Sparse PIC results correspond to N = 7.

Error (%)			
	sparse 2D	sparse 2D	sparse 2D
	N_{pc} = 3280	N_{pc} = 512	N_{pc} = 100
n_i	1.55	1.8	2.44
$EEDF$	0.07	0.1	0.5

Table 7: Test-Case C: Differences between standard and sparse PIC results. Sparse PIC results corresponds to N = 7.

IV. Computational efficiency

Two types of clusters have been used with different Intel architectures and compilers versions:

- Case A and Case C, Laplace supercomputer, 2 × Haswell per node (Intel Xeon CPU E5-2699A v4, 2.40 GHz, 22 cores), with Intel compiler version 17.0.1.132 and IntelMPI version 17.0.

- Case B, Calmip supercomputer, 2 × Skylake per node (Intel Xeon Gold 6140, 2.30 GHz, 18 cores), with Intel compiler version 18.2.199 and IntelMPI version 18.2.

Standard and sparse PIC calculations have been performed on the same architectures and compiler versions, without considering an optimization of the number of processors used according to the method. Note that one node has been used for Case A and Case C and 5 nodes for Case B. Figure 9 is showing the speed up for the sparse compared to the standard PIC algorithm to reach the same physical time (time window of 5 μs). Results for Case A and Case C being close, only Case A result are shown. For Case B, to do a comparison in same conditions than standard PIC method, sparse PIC simulations have been performed with N_x=N_y=512 (with a simulation time of 300 μs). Also in Case B, calculations for N_x=N_y=128 have been performed with standard PIC method to compare executional times with sparse PIC method for same N_{pc}. Keeping the same total number of particles N_{T} in the sparse and standard PIC models results in a calculation with a slower execution time in the former (see the point at the right-hand-side and bottom side of Fig. 9). This is the results of a larger number of computations in the case of the sparse algorithm. A clear benefit is observed when the number of particles per cell is identical for a given depth N. The speedup increases from about 2 to 5 for a number of grid points N_{g} ranging between 128^2 (N = 7) to 512^2 (N = 9) as shown in Fig. 9, respectively.

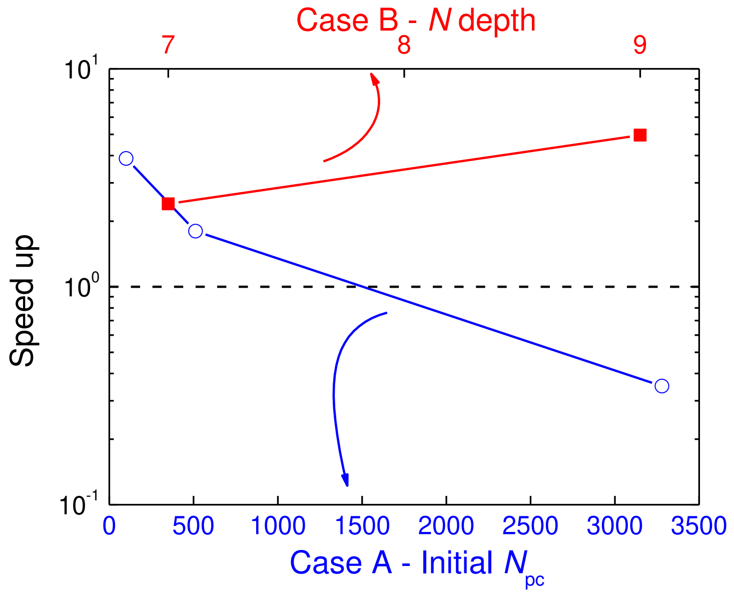


Figure 9: Speedup measured by dividing the execution time of the sparse by the one of the standard PIC method. For Case A (blue color), the depth is kept at $N = 7$ in both models while N_{pc} is varied from 3280 to 100 in the sparse PIC. $N_{\mathrm{pc}} = 512$ in the standard PIC calculation. For Case B (in red), the depth varies from $N = 9$ to $N = 7$ and N_{pc} is set at 64 in both algorithms.

Note that the speedup is expected to be significantly larger in 3D as the ratio $G_{\mathrm{r}}/G_{\mathrm{sparse}}$ is more favorable [35]. We show in figure 10 a comparison between the ratio of the number of cells used in the standard PIC versus the sparse PIC for 2D and 3D domains as a function of depth N. For instance, $N = 9$ ($M = 512$), the reduction of the number of cells is around 20 in 2D and ~ 1000 in 3D.

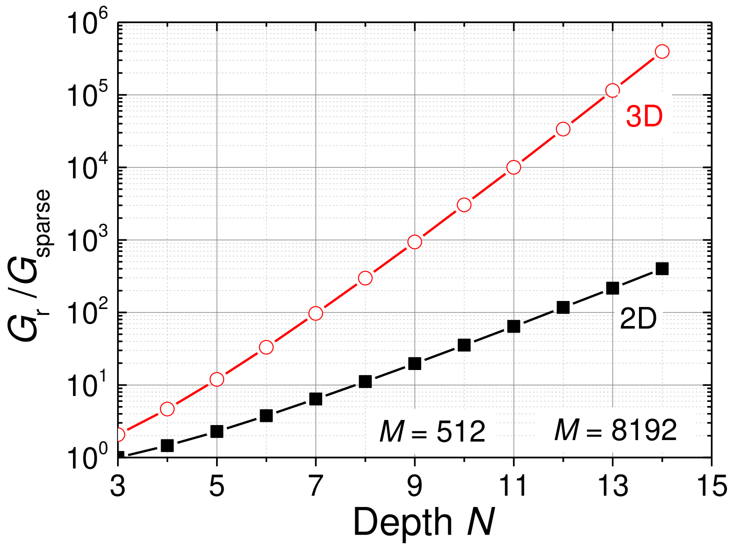


Figure 10: Comparison of the total number of cells for the Cartesian mesh (with a mesh size 2^{-N}) divided by its sparse grids hierarchy counterpart $G_{\mathrm{r}}/G_{\mathrm{sparse}}$ as a function of the depth N for 2D and 3D domains.

The computational gain is in general lower than the one estimated from a simple ratio of the number of grid cells between the standard and sparse grid method (see figure 1). It is hence instructive to consider the time passed in subroutines in order to understand the actual bottlenecks associated with the implementation of the sparse PIC algorithm. Figure 11 shows the comparison between the sparse and standard PIC technique for the same number of particles per cell. The subroutine PUSH includes the interpolation of the electric field at particle position and the push of particles. The subroutine CHARGE denotes the charge deposition on grid nodes and the reduction to calculate the total charge on each node summed over the OpenMP and MPI threads. The subroutine POISSON covers the electric potential calculation and the calculation of the electric field components using second order accurate finite difference schemes. The subroutine COLLISION corresponds to the Monte Carlo procedure to model collisions between charged particles and neutral atoms. In Fig. 11, whatever the method, we observe a reduction of the time passed in PUSH for both algorithms and an increase of the time spent in COLLISION when comparing Case A and Case B. This is due to number of particles N_T inside the simulation domain in Case B together with an increase of the collision rates because the neutral density is higher. Comparatively to the standard method, a larger fraction of the computational time is dedicated to the charge deposition onto the $2N - 1$ sub-grids (even if the total number of nodes is less than with the regular grid algorithm). Using sparse grids, the reduction of the total number of grid nodes induces a significant gain in computational time for the Poisson solver; the execution time becomes negligible in our case (more than 50 % of the time spent in the POISSON subroutine is devoted to the calculation of the electric field components from the potential profile). Lastly, comparatively to the standard method, a larger fraction of computational time is dedicated to the charge deposition onto the nodes of the $2N - 1$ grids of the sparse hierarchy. Note that our model has not been fully optimized yet in terms of parallelization.

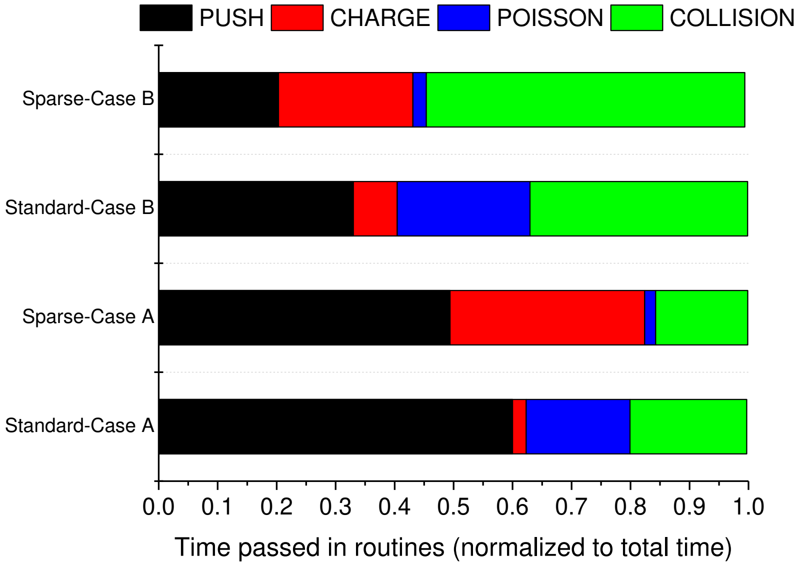


Figure 11: Time spent in subroutines. For standard PIC calculations and for Case A, $N_g = 128^2$ and $N_{pc} = 512$ and for Case B $N_g = 512^2$ and $N_{pc} = 64$. For sparse grids: Case A, $N = 7$ ($N_g = 128^2$) and $N_{pc} = 512$, while for Case B, $N = 9$ and $N_{pc} = 64$.

V. Conclusions

This paper addresses the issue of PIC algorithm acceleration using sparse grids and its application to model low temperature plasmas. The novel method is based on the construction of a hierarchy of mostly anisotropic grids (i.e., with rectangular cells in 2D) whose total number of cells is significantly reduced compared to one unique computational domain defined by cells of squared shape as in the standard PIC technique. The reduction factor (defined as the number of cells in the sparse divided by the ones in the standard PIC algorithms) increases with the dimensionality of the problem. Plasma parameters are reconstructed at a given location inside the simulation domain using the so-called combination technique. The novel PIC algorithm is very similar to the standard one on a regular grid and the same parallelization approaches can be efficiently implemented.

The PIC sparse grid combination technique has been successfully used in the context of open-boundary problems where ions are at rest and collisions omitted [35]. The goal of this paper was to verify the capability of the new method to resolve gradients encountered in the sheath region in front of electrodes in low temperature plasma discharges. We have modeled a benchmark of a one-dimensional capacitively coupled radiofrequency discharges which was already published in the literature. We considered two extreme test cases: the first one corresponds to a low pressure discharge for which the EEDF is a Maxwellian. Comparisons between standard and sparse grid PIC calculations showed a strong capability for the latter to reproduce the plasma characteristics with a high degree of accuracy; the difference in the ion density and ionization source term profiles where of the order of 0.4 %. Furthermore, the EEDF can be captured with an error on the whole distribution of less than 0.5 % with a resolution of six orders of magnitude. These results have been obtained keeping the same number of particle-per-cell in both algorithms. Reducing N_{pc} in sparse grid calculations does not degrade considerably the error (less than 0.6 % for the profiles and 3.5 % for the EEDF) offering a gain in computational time. The noise level is typically smaller compared to the standard PIC estimates. The second case corresponds to a larger pressure and a large deviation of EEDF from Maxwellian is observed. Comparisons between the two methods have been realized reducing the depth N of grids keeping N_{pc} identical to the standard method. The relative error for ion density and ionization profiles was to be less than 3.5 %. The difference in the EEDF calculation (down to 6 orders of magnitude) is less than 6 %. Finally, for conditions close to Case A, we have considered a real 2D problem using non symmetrical Dirichlet conditions for the electric potential. Again, using the sparse PIC algorithm, the EEDF can be retrieved with a large precision and an accuracy of 0.1 % for the same N_{pc}.

When N_{pc} is kept the same as in the standard PIC method and the depth constructed from the number of grid nodes on the regular grid, the gain of computational time is important, between 2 and 5, for Case A (or Case C) and B, respectively. A detailed analysis of the time spent in subroutines has shown a strong reduction of the execution time for the Poisson solver. The charge deposition to calculate the particle densities is slower with the sparse PIC but our subroutine was not fully optimized. This work was a first step to prove the applicability of the sparse PIC technique to plasmas bounded by absorbing walls (i.e., with a sheath which develops in front of the latter), periodic boundary conditions, time varying electrode potentials which impact the EEDF and the inclusion of a complex physical-chemistry. The next step will be to reproduce similar results in 3D where the computational gain is expected to be substantial compared to the standard PIC approach. In a companion paper, we have applied the sparse grid combination technique to the context of partially magnetized low temperature plasmas [36] where both plasma instabilities and an anomalous electron current are present.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This work was supported by the RTRA STAE (Réseau Thématique de Recherche Avancée Sciences et Technologies pour l'Aéronautique et l'Espace) foundation. Part of this work was granted access to the HPC resources of CALMIP supercomputing center under the allocation 2013-P1125. Authors want to thank Mathieu Lobet from Maison de la Simulation for discussions and implementation of SIMD optimization techniques.

References

[1] R. W. Hockney and J. W. Eastwood, *Computer simulation using particles* (IOP Publishing Ltd, 1989).

[2] C. K. Birdsall and A. B. Langdon, *Plasma physics via computer simulation* (Taylor and Francis, 2005)

[3] T. Tashima, *Computational Plasma Physics: With Applications To Fusion And Astrophysics* (Taylor and Francis, 2018).

[4] C. K. Birdsall, IEEE Transactions on Plasma Science **19**, 65 (1991).

[5] J. P. Verboncoeur, Plasma Physics and Controlled Fusion **47**, A 231 (2005).

[6] G. Lapenta, Journal of Computational Physics **231**, 795 (2012).

[7] C. S. Meierbachtol, A. D. Greenwood, J. P. Verboncoeur, and B. Shanker, IEEE Transactions on Plasma Science **43**, 3778 (2005).

[8] T. D. Arber, K. Bennett, C. S. Brady, A. Lawrence-Douglas, M. G. Ramsay, N. J. Sircombe, P. Gillies, R. G. Evans, H. Schmitz, A. R. Bell, and C. P. Ridgers, Plasma Physics and Controlled Fusion **57**, 113001 (2015).

[9] V. Vahedi and M. Surendra, Computer Physics Communications **87**, 179 (1995).

[10] M. M. Turner, A. Derzsi, Z. Donko, D. Eremin S. J. Kelly, T. Lafleur, and T. Mussenbrock, Physics of Plasmas **20**, 013507 (2013).

[11] M. Surendra, D. B. Graves, and I. J. Morey, Applied Physics Letters **56**, 1022 (1990).

[12] P. A. Delattre, T. Lafleur, E. Johnson and J. P. Booth, Journal of Physics D: Applied Physics **46**, 235201 (2013).

[13] S. Mattei, K. Nishida, M. Onai, J. Lettry, M. Q. Tran, and A. Hatayama, Journal of Computational Physics **350**, 891 (2017).

[14] T. Melzig, M. Siemers, A. Pflug, and R. Rank, Surface and Coatings Technology **241**, 30 (2014).

[15] T. M. Minea, C. Costin, A. Revel, D. Lundin, and L. Caillault, Surface and Coatings Technology **255**, 52 (2014).

[16] G. Fubiani, L. Garrigues, G. Hagelaar, N. Kohen, and J. P. Boeuf, New Journal of Physics **19**, 015002 (2017).

[17] L. Garrigues, G. Fubiani, and J. P. Boeuf, Journal of Applied Physics **120**, 213303 (2016).

[18] I. M. Montellano, D. Wünderlich, S. Mochalskyy, and U. Fantz, Journal of Physics D: Applied Physics **52**, 235202 (2019).

[19] F. Taccogna and P. Minelli, Physics of Plasmas **25**, 061208 (2018).

[20] J. P. Boeuf and L. Garrigues, Physics of Plasmas **25**, 061204 (2018).

[21] F. Shi, D. Wang, and C. Ren, Physics of Plasmas **15**, 063503 (2008).

[22] H. Y. Kim, M. Gołkowski, C. Gołkowski, P. Stoltz, M. B. Cohen, and M. Walker, Plasma Sources Science and Technology **25**, 055011 (2018).

[23] J. Teunissen and U. Ebert, Journal of Physics D: Applied Physics **51**, 474001 (2018).

[24] D. Levko, M. Pachuilo, and L. L. Raja, Journal of Physics D: Applied Physics **51**, 354004 (2018).

[25] F. Taccogna and F. Pelligrini, Journal of Physics D: Applied Physics **51**, 06401(2018).

[26] H. Okuda and C. K. Birdsall, Physics of Fluids **13**, 2123 (1970).

[27] M. M. Turner, Physics of Plasmas **13**, 033506 (2006).

[28] J. C. Adam, A. Gourdin Serveniere, and A. B. Langdon, Journal of Computational Physics **47**, 229 (1982).

[29] R. J. Mason, Journal of Computational Physics **51**, 484 (1983).

[30] A. B. Langdon, B. I. Cohen, and A. Friedman, Journal of Computational Physics **51**, 107 (1983).

[31] M. Drouin, L. Gremillet, J. C. Adam, and A. Héron, Journal of Computational Physics **229**, 4871 (2010).

[32] G. Chen, L. Chacón, and D. C. Barnes, Journal of Computational Physics **230**, 7018 (2011).

[33] S. Markidis and G. Lapenta, Journal of Computational Physics **230**, 7037 (2011).

[34] L. Chacón, "Fully implicit particle in cell algorithm for electromagnetic plasma simulations", Numkin 2013 (Garching, Germany).

[35] L. F. Rickeston and A. J. Cerfon, Plasma Physics and Controlled Fusion **59**, 024002 (2017).

[36] L. Garrigues, B. Tezenas du Montcel, B. Reman, and G. Fubiani, *submitted (2021).*

[37] J. Garcke, "Sparse Grids in a Nutshell", in Sparse Grid Methods and Applications, J. Garcke and M. Griebel, Lecture Notes in Computational Science and Engineering, pages 57-80 (2012).

[38] C. Zenger, “Sparse grids”, in *Parallel Algorithms for Partial Differential Equations* (W. Hackbusch, ed.), Vol. 31 of Notes on Numerical Fluid Mechanics, Vieweg, Braunschweig/Wiesbaden (1991).

[39] H. J. Bungartz and M. Griebel, Acta Numerica **13**, 147 (2004).

[40] Q. Liu, L. Ding, and Q. Liu, Mathematical Methods in Applied Sciences **37**, 870 (2014).

[41] A. Rütgers and M. Griebel, Applied Mathematics and Computation **319**, 425 (2018).

[42] J. Garcke and M. Griebel, Journal of Computational Physics **165**, 694 (2000).

[43] D. Pflüger, H. J. Bungartz, M. Griebel, F. Jenko, T. Dannert, M. Heene, C. Kowitz, A. P. Hinojosa, and P. Zaspel, “Exahd: an exa-scalable two-level sparse grid approach for higher-dimensional problems in plasma physics and beyond”, *Euro-Par 2014: Parallel Processing Workshops* (Berlin: Springer), page 565 (2014).

[44] C. Kowitz and M. Hegland, Proceda Computer Science **18**, 449 (2013).

[45] H. J. Bungartz, A. Heinecke, D. Plüfger, and S. Schraufstetter, Journal of Computational and Applied Mathematics **236**, 3741 (2012).

[46] A. Klimke, “Uncertainty modeling using sparse grids”. *PhD dissertation*, Universität Stuttgard, Germany (2006).

[47] M. Griebel, M. Schneider, and C. Zenger, “A combination technique for the solution of sparse grid problems”, in Iterative Methods in Linear Algebra, P. de Groen and R. Beauwens, editors, pages 263–281 (1992).

[48] K. J. Bowers, Journal of Computational Physics **173**, 393 (2001).

[49] H. Vincenti, M. Lobet, R. Lehe, R. Sasanka, and J. L. Vay, Computer Physics Communications **210**, 146 (2017).

[50] A. Beck, J. Dérouillat, M. Lobet, A. Farjallah, F. Massimo, I. Zemzemi, F. Perez, T. Vinci and M. Grech, Computer Physics Communications **244**, 246 (2019).

[51] O. Schenk and K. Gartner, Journal of Future Generation Computer Systems **20**, 475 (2004).

[52] G. Zumbusch, *Mesh Refinement and Loadbalancing* in *Parallel Multilevel Methods, Adaptive* (Springer, 2003), Chapter 2.

[53] Biagi-v7.1 database, http://www.lxcat.net, retrieved data 2004.

[54] A. V. Phelps, Journal of Applied Physics **76**, 747 (1994).