Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2007) 143–155
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Adaptive particle management in a particle-in-cell code

D.R. Welch *, T.C. Genoni, R.E. Clark, D.V. Rose

Voss Scientific, LLC, 418 Washington SE, Albuquerque, NM 87108, United States

Received 2 March 2007; received in revised form 25 June 2007; accepted 19 July 2007
Available online 3 August 2007

Abstract

In particle-based plasma simulation, when dealing with source terms such as ionization, emission from boundaries, etc., the total number of particles can grow, at times, exponentially. Problems involving the spatial expansion of dynamic plasmas can result in statistical under representation of particle distributions in critical regions. Furthermore, when considering code optimization for massively parallel operation, it is useful to maintain a uniform number of particles per cell. Accordingly, we have developed an algorithm for coalescing or fissioning particles on 2D and 3D orthogonal grids that is based on a method of Assous et al. [F. Assous, T. Pougeard Dulimbert, J. Segre, J. Comput. Phys. 187 (2003) 550]. We present the algorithm and describe in detail its application to particle-in-cell simulations of gas ionization/streamer formation and dynamic, expanding plasmas.

Keywords: Numerical methods; Particle-in-cell; Plasma simulation; Gas breakdown

1. Introduction

The broad applicability of the particle-in-cell (PIC) method to fundamental and applied problems in plasma physics is due to the economy of representing six-dimensional phase space by a finite number of macroparticles [1,2]. For many problems, the number of macroparticles used in a simulation is easily controlled by limiting the initial number or the frequency of new particle creation. In many cases, however, the number of particles can grow without bound. This behavior is particularly troublesome in complex simulations where particle interactions with other particles and with surfaces are treated. Neutral gas ionization in which secondary ionization or avalanche occurs can lead to exponential growth of macroparticles. Other related problems involve the stimulated emission of particles by surface bombardment of other particles. Conversely, a related problem in particle simulation involves having too few particles in a cell. This can lead to artificial field enhancement and inadequate representation of particle distribution functions.

Pairwise coalescence (see Ref. [3]) of particles has been utilized previously to manage macroparticle number in PIC simulations. The method searches for particles that have matching momenta to a specified tolerance in

* Corresponding author.
E-mail address: dale.welch@vosssci.com (D.R. Welch).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.07.015

a given cell. The two particles are combined into a single particle with mass-averaged position and momentum. The global charge, mass, and momentum are exactly conserved; however, small corrections for local charge conservation must be made in electromagnetic simulation. The energy conservation is only as good as the specified tolerance and necessarily results in cooling because the velocity space is compressed.

In this paper, we discuss an adaptive particle management (APM) algorithm built upon the formalism outlined by Assous et al. [4]. Here, basic quantities such as charge, momentum, and energy are globally and locally conserved without need of correction. Our modification and extension of the algorithm involves an efficient approach to the accurate reproduction of the initial spatial and velocity distributions within a cell. Based on a fixed orthogonal grid, the efficiency and accuracy of the method is optimized by characterizing the original particle phase space within a cell and placing new particles in the more densely populated regions of that space. In addition, momentum and energy conservation are enforced in such a way that the velocity distribution of the replacement particles closely approximates that of the original set. The method can be used to reduce or increase particle number in each cell. This can be useful in maintaining a uniform load balance for parallel simulation. Unlike the pairwise method, APM permits accurate conservation by making use of many particles.

In Section 2, we present the APM formalism which conserves particle quantities on an orthogonal grid. Grid (and total) particle weights, momenta, and energy are conserved while preserving the essential features of the velocity distributions. In Section 3, an illustrative example of particle coalescence in a single cell is presented. The algorithm is then tested on two stressing gas breakdown problems in which the macroparticle number grows exponentially, quickly becoming unmanageable. The first problem simulates streamer formation and propagation from an applied electric field in hydrogen gas and the second involves the breakdown of nitrogen gas by an intense relativistic electron beam. In both cases, the results reasonably reproduce avalanche rates when compared to simulations without APM, while the simulations using APM exhibit dramatically reduced computational run times. As an example of the utility of the APM fissioning feature, the simulation of an expanding plasma is examined in which particle statistics otherwise degrade with time. Here, the APM algorithm maintains a constant number of particles per cell as the plasma volume increases. The algorithm as described here has been implemented in the 3D PIC code LSP [5] and all sample calculations presented in Section 3 were carried out using this code. Conclusions are presented in Section 4.

2. APM method

The APM approach is based on the general method outlined by Assous et al. [4]. It is a local approximation in that particles in each computational cell are coalesced (or fissioned) into a smaller (or larger) number. The user chooses N, the maximum (or minimum) allowable number of particles in a single cell, and M, the target number of replacement particles after coalescence (after fissioning). The new set of particle weights or masses is calculated according to a prescription outlined in Ref. [4], and generalized here to 2D and 3D orthogonal grids. The method conserves the grid weights (4 and 8 corner weights in 2D and 3D, respectively), and also the total particle weights. Particle momenta are then calculated in a similar fashion. The final step achieves energy conservation, while at the same time preserving (in a statistical sense) the properties of the relativistic velocity distributions of the original N particles.

2.1. Particle weights

The method allows for arbitrary location of the M replacement particles within the cell. (The first attempts at implementation were done with the M particles placed on a uniform sub-cell grid, which proved to be highly inefficient and was later modified.) At each grid node the charge is given by

$$Q_i = \sum_{n=1}^{N} w_n \lambda_i(\boldsymbol{x}_n), \quad i = 1, 2, \ldots, 8, \tag{1.1}$$

where w_n and $\boldsymbol{x}_n$ are the original particle weights and positions. The bilinear particle interpolation functions λ_i, which correspond to cell corners with coordinates (x_i, y_i, z_i), are given by

$$\lambda_i = \left(1 - \frac{|x_n - x_i|}{\Delta x}\right)\left(1 - \frac{|y_n - y_i|}{\Delta y}\right)\left(1 - \frac{|z_n - z_i|}{\Delta z}\right). \tag{1.2}$$

We replace the original N particles with M particles at positions $\boldsymbol{x}_m$ and with weights w_m such that the grid weights are conserved, i.e.

$$\sum_{m=1}^{M} w_m \lambda_i(\boldsymbol{x}_m) = Q_i. \tag{1.3}$$

We note that the property $\Sigma\lambda_i = 1$ ensures conservation of total charge as well. We now require a function $G(\boldsymbol{x})$ to describe the distribution of weights such that $w_m = G(\boldsymbol{x}_m)$. Following Ref. [4] we choose (in 3D)

$$G(\boldsymbol{x}) = \sum_{j=1}^{8} g_j \lambda_j(\boldsymbol{x}), \tag{1.4}$$

although other weighting schemes are possible. Substituting Eq. (1.4) into Eq. (1.3), we obtain

$$\sum_{j=1}^{8} g_j \sum_{m=1}^{M} \lambda_j(\boldsymbol{x}_m)\lambda_i(\boldsymbol{x}_m) = Q_i. \tag{1.5}$$

We now solve the system of eight linear equations (1.5) for coefficients g_j, from which we obtain the particle weights for the M replacement particles

$$w_m = \sum_{j=1}^{8} g_j \lambda_j(\boldsymbol{x}_m). \tag{1.6}$$

The system of linear equations requires at least $M, N \geqslant 2^D$ for a solution where D is the dimensionality (e.g. $D = 3$ for 3D). We note here that the method does not ensure that all the M replacement particle weights will be greater than zero, a physical requirement. Depending on the distribution of the initial N particles within the cell, and the choice of locations for the replacement particles, cases can occur that give rise to weights <0, a clearly unphysical result, or to very small weights that result in particles with unreasonably large momentum and energy. We discuss this point further below.

2.2. Momentum conservation

Having chosen the particle locations $\boldsymbol{x}_m$ and calculated the M weights w_m, similar calculations are performed to assign each of the three components of the replacement particle momenta. The grid momenta in the x-direction (the two other directions are similarly obtained) are given by

$$P_{xi} = \sum_{n=1}^{N} w_n p_{xn} \lambda_i(\boldsymbol{x}_n), \quad i = 1, 2, \ldots, 8, \tag{1.7}$$

where p_{xn} are the original particle momentum in the x-direction and the bilinear interpolation functions λ_i are again used. As before, we choose a weight function

$$H(\boldsymbol{x}) = \sum_{j=1}^{8} h_j \lambda_j(\boldsymbol{x}), \tag{1.8}$$

and use the conservation of grid momenta to obtain the system of linear equations for the coefficients h_j

$$\sum_{j=1}^{8} h_j \sum_{m=1}^{M} \lambda_j(\boldsymbol{x}_m)\lambda_i(\boldsymbol{x}_m) = P_{xi}, \quad i = 1, 2, \ldots, 8. \tag{1.9}$$

Solving for h_j, we obtain the x-momentum for the mth particle

$$p_{xm}^{\text{ave}} = \frac{1}{w_m} \sum_{j=1}^{8} h_j \lambda_j(\boldsymbol{x}_m). \tag{1.10}$$

We use the superscript "ave" to denote the fact that to this point we have conserved momentum only in the average sense. The random (or thermal) components of the p_{xm} are lost in the sums of Eq. (1.7). They are reintroduced into the replacement particle momenta p_{xm} as described in the next section. Here, there is no requirement on the sign of the particle momentum. It is actually the weighted momentum of the macroparticles that is solved for, and the particle p_{xm}^{ave} determined by dividing by w_m. Thus, a very small particle weight can result in undesirably large momentum and energy. Such cases were seen in numerical tests and prompted a modification of the algorithm described in the following sections.

2.3. Energy conservation

To this point, the calculated momentum distributions of the replacement particles only preserve the average properties accounting for local gradients. These distributions are thus strongly peaked about some average value. No account of possible random or thermal energy in the original distribution has been taken, and energy has not yet been conserved. The momentum (again taking the direction x as an example) of each replacement particle is perturbed by adding a random contribution p_x^{ran} (proportional to a parameter β)

$$w_m p_{xm} = w_m p_{xm}^{\text{ave}} + \beta p_x^{\text{ran}}(\boldsymbol{x}_m), \tag{1.11}$$

where in order to continue to preserve conservation of grid and total particle momenta we require

$$\sum_{m=1}^{M} \lambda_i(\boldsymbol{x}_m) p_x^{\text{ran}}(\boldsymbol{x}_m) = 0. \tag{1.12}$$

We construct the $p_x^{\text{ran}}(\boldsymbol{x}_m)$ according to

$$p_x^{\text{ran}}(\boldsymbol{x}_m) = p_{xm}^{\text{sam}} + \sum_{j=1}^{8} b_j \lambda_j(\boldsymbol{x}_m), \tag{1.13}$$

where the p_{xm}^{sam} are random contributions sampled from the appropriate distribution as described below. The additional sum involving the coefficients b_j is added in order to preserve momentum conservation. Substitution of Eq. (1.13) into (1.12) gives

$$\sum_{j=1}^{8} b_j \sum_{m=1}^{M} \lambda_j(\boldsymbol{x}_m)\lambda_i(\boldsymbol{x}_m) = -\sum_{m=1}^{M} \lambda_i(\boldsymbol{x}_m) p_{xm}^{\text{sam}}. \tag{1.14}$$

This process is repeated for the y- and z-momenta. We now equate the relativistic energy of the original N particles to the energy of the M replacement particles

$$\sum_{n=1}^{N} w_n \sqrt{1 + p_{xn}^2 + p_{yn}^2 + p_{zn}^2} = \sum_{m=1}^{M} w_m \sqrt{1 + p_{xm}^2 + p_{ym}^2 + p_{zm}^2}, \tag{1.15}$$

which gives a single non-linear equation for β through Eq. (1.11). A simple Newton–Raphson solve is used to find β. There is no guarantee of a real solution; however, in numerical tests to date, convergence failures have been extremely rare and a relative energy conservation error $<10^{-5}$ is usually obtained in <10 iterations. A convenient choice for the initial guess for β, based on a non-relativistic energy calculation in Ref. [4], expedites the convergence.

2.4. Distribution functions

A key aspect of our APM algorithm is the method of choosing the momentum perturbations p_{xm} which are added to the original "average" momenta in the energy conservation calculations. These are chosen to closely approximate the original momentum distribution. It should be noted that any method for the reconstruction of a distribution requires a sufficient statistical sample. While highly problem dependent, we have found empirically that $N \geqslant 100$ and $M \geqslant 50$ give reasonable accuracy for gas breakdown simulations. One

approach we have taken involves the use of a set of analytic functions to approximate the momentum distributions about the mean, $f(u)$, of the set of particles to be coalesced. We make use of a uniform distribution

$$f^{\text{unif}}(u) = \frac{1}{2\sqrt{3}}, \quad -\sqrt{3} \leqslant u \leqslant \sqrt{3}, \tag{1.16}$$

Gaussian distribution

$$f^{\text{G}}(u) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{u^2}{2}\right], \tag{1.17}$$

and distributions with large wings of the form

$$f^{k}(u) = \frac{c_k}{\pi} \frac{1}{\left(1 + \frac{u^2}{2k-3}\right)^k}, \tag{1.18}$$

for $k = 2, 3, \ldots, K$. To date, we have considered only symmetric distributions and have used momentum cutoffs of $\pm 7\sigma$ in the Gaussian and large wing cases. In applications thus far, we have used $k = 2, 3, 4, 6, 8$ and 10 which gives us a set of eight possible distributions. The $f(u)$ appropriate in a particular case is identified as follows. For each distribution in the set, we can associate a unique value of the ratio R of the fourth to the square of the second velocity moments

$$R = \frac{\int u^4 f(u) \mathrm{d}u}{\left[\int u^2 f(u) \mathrm{d}u\right]^2}, \tag{1.19}$$

where it is understood that the variable u is measured from the mean of the distribution. The ratio R is a measure of the "peakedness" of the probability distribution. (In probability theory, R-3 is called the "Kurtosis" or "excess Kurtosis" of the distribution.) From the original set of N particles we calculate

$$R_N = \frac{\sum_{n=1}^{N} w_n (p_{xn} - \langle p_x \rangle)^4}{\left[\sum_{n=1}^{N} w_n (p_{xn} - \langle p_x \rangle)^2\right]^2}, \tag{1.20}$$

and select a single distribution for which R is closest to R_N from the set of possible distributions defined above.

This process establishes a distribution shape which is then sampled to determine the individual momenta. The original distribution shape is thus preserved in a statistical sense. Our algorithm at present handles distributions symmetric about the mean, but it is straightforward to generalize for asymmetric distributions. This "fitted" function approach is useful for problems where fine detail in the distributions is either not expected or inconsequential.

For problems where finer details in the distribution and correlations in the three directions are important, we employ a method of sampling the original N particle momentum distribution. One such problem involves gas interaction and breakdown where energy losses due to ionization and excitation are important. In these problems, we construct a 3D velocity space from the original particles that is sampled M times as

$$\int_0^{p_{n'}} f^{\text{dis}} \, \mathrm{d}\boldsymbol{p}' = \frac{\sum_{n=1}^{n'} w_n}{\sum_{n=1}^{N} w_n} = r, \tag{1.21}$$

where r is a random number from 0 to 1 giving the random momentum vector $\boldsymbol{p}_m^{\text{ran}} = (p_{xn'} - p_{xm}^{\text{ave}}, p_{yn'} - p_{ym}^{\text{ave}}, p_{zn'} - p_{zm}^{\text{ave}})$, where the local mean momentum has been subtracted out. This "discrete" method, to a large extent, preserves the original phase space correlations and asymmetries.

2.5. Treatment of fluid particles

The hybrid code LSP has the capability of describing the equations of motion and energy via a PIC fluid [7]. In this case, the particles are assumed to move as an ensemble with local mean velocity. There is no random momentum about this mean, but an internal energy or temperature is carried by each particle. We can treat

the temperature in a similar way as the charge since they are both scalar quantities which are interpolated to the grid corners

$$T_i = \sum_{n=1}^{N} w_n T_n \lambda_i(\boldsymbol{x}_n). \tag{1.22}$$

Using the identical treatment found in Eqs. (1.1)–(1.6), the temperature T_m can be found for each replacement particle.

3. Numerical tests

Several numerical tests have been carried out with simulations involving neutral gas ionization resulting in rapid growth of the number of simulation particles. In these cases, the grid was swept periodically and those cells containing more than 100 particles were coalesced to typically $M = 25$. The original choice of a uniform sub-cell grid for placement of the M particles (noted at the beginning of Section 2) resulted in a large number of cases with either negative or small weights such that the corresponding particle momentum and energy were unphysically large. We determined that the source of the problem was cells in which the original distribution of weights was highly non-uniform, and could not accurately be represented by a uniform replacement particle distribution. To address this problem, we numerically binned the original particle weights on a sub-grid and then placed the replacement particle positions to more accurately reflect this spatial distribution. This is achieved as follows. We first establish a "target" mean replacement weight according to $w_{\text{tmean}} = M^{-1}\Sigma w_n = M^{-1}W_{\text{total}}$. Then the weight w_{sc} is calculated for each sub-cell, and j replacement particles are laid down randomly within that sub-cell such that $(j-0.5)w_{\text{tmean}} < w_{\text{sc}} < (j+0.5)w_{\text{tmean}}$. To within the resolution of the sub-grid, the replacement particle spatial distribution approximates the original. With this modification, test simulations have achieved cell coalescence at a 98% success rate. With these improvements, the dominant computational expense is that of sorting particles to each grid cell. Because the APM calculation typically requires the computational time associated with a simulation time step, an interval for the APM of 100–500 does not significantly slow the simulation.

3.1. Single-cell algorithm test

As an illustration of the APM method, presented here are detailed results from coalescence on a single cell originally containing 200 particles. The particles were given random weights and laid down non-uniformly within the cell (see Fig. 1) in order to test the method of determining replacement particle positions described above. A 7×7 sub-grid was used and the method resulted in the distribution of positions shown in Fig. 1(a). (The number of replacement particles approximates the number of sub-grid cells, but can be greater or less than that number by a few. In the particular example shown in Fig. 1(b), $M = 47$.) The non-uniformity of the initial distribution is reflected in the positions of the replacement particles, resulting in a reasonably uniform set of replacement particle weights. The M weights varied from 0.69 to 1.16 times the average weight, and the potential problem of unphysically large momenta was thus avoided.

The reconstruction of the x-momentum distribution of the initial 200 particles is shown in Fig. 1(c). The initial momenta were selected randomly from a Gaussian distribution with average value 0.01 and standard deviation 0.002. Following the application of grid-momentum conservation (Eqs. (1.7)–(1.10)), the x-momenta of the M replacement particles were distributed as shown in Fig. 1(c). At this point, it is clear that momentum has been conserved only in the average sense—the replacement particle distribution is strongly peaked near the average and bears no resemblance to the original Gaussian. The final step in the process is the energy conservation procedure of Eqs. (1.11)–(1.15) and (1.20). This procedure results in the x-momentum distribution that reproduces the original distribution at least in a statistical sense. This was checked by performing the coalescence process 100 times with only random variation between trials. When this process was repeated with increasing particle number, the original and replacement distributions approach each other in both magnitude and shape as expected. Below we present results from three applications of the APM algorithm in 2D PIC plasma simulation.

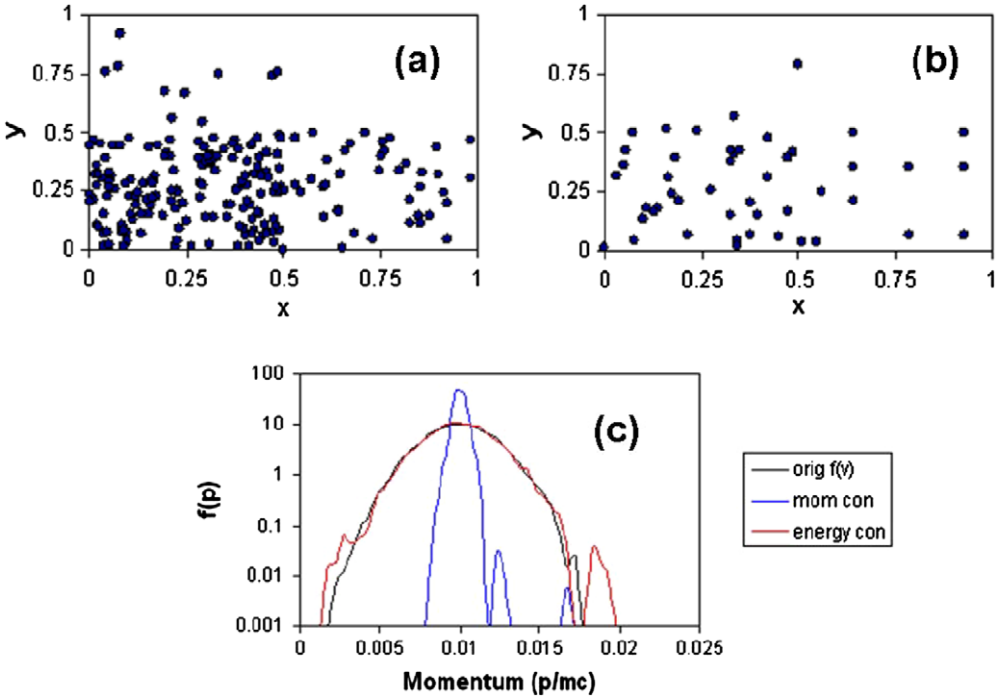

Fig. 1. The APM test is illustrated for a single cell of particles with most residing the lower left quadrant. The initial (a) and replacement (b) particle positions are plotted. In (c), the initial momentum distribution, the replacement particle distribution after the momentum conservation (mom con) phase and the replacement distribution after the final energy conservation (energy con) phase are plotted.

3.2. Streamer propagation problem

In this section, we test the utility of the APM algorithm by simulating the evolution and dynamics of a plasma streamer [6] in 1 atm. of hydrogen. The Lsp implicit collisional electromagnetic particle method [7,8] is used in 2D cylindrical (r, z) coordinates to model the streamer propagation. Energy-dependent cross sections for electron interaction (treated in Monte Carlo fashion) with neutral hydrogen include impact ionization, elastic scattering and inelastic energy loss. Coulomb collisions between all charged particles assuming Spitzer collision frequencies are also followed. This model follows the streamer propagation and evolution from an initially low density localized electron swarm (seed population) to high density (many orders-of-magnitude in number density). Because electrons with energies above the ionization potential of the neutral gas molecules have the greatest energy loss due to inelastic collisions, the electron distribution is non-Maxwellian. Thus, it is a good test case for studying the sensitivity of the streamer propagation to the choice of APM energy distribution.

In these calculations, a coaxial wave guide section is driven by a voltage wave giving electric fields of roughly 30 kV/cm across an 8-mm anode–cathode (AK) gap. A 3×10^8 cm^{-3} plasma was initialized 1 mm from the cathode (at $z = 8$ mm) on axis as shown in Fig. 2(a). Typically, a weak streamer initially formed a small distance from the seed location. As the streamer propagated and the density increased, eventually the electrons in the streamer tail became anchored in space by the increasing space charge. At this point the density of the plasma electrons and ions became comparable and the plasma began to shield out the electric field within the streamer body. The electric field is enhanced at the streamer edges to roughly twice the initial field in the AK gap. The electron temperature at the edges is roughly 4 eV which enables a weak avalanche that sustains the streamer. The streamer front velocity was roughly 20 cm/μs.

Because the problem involves avalanche ionization with each electron impact that results in an ionization event producing a new electron–ion pair, an increase in plasma density produces a corresponding increase in particle number. Given the 3×10^8 cm^{-3} initial seed density, following the streamer density to greater than

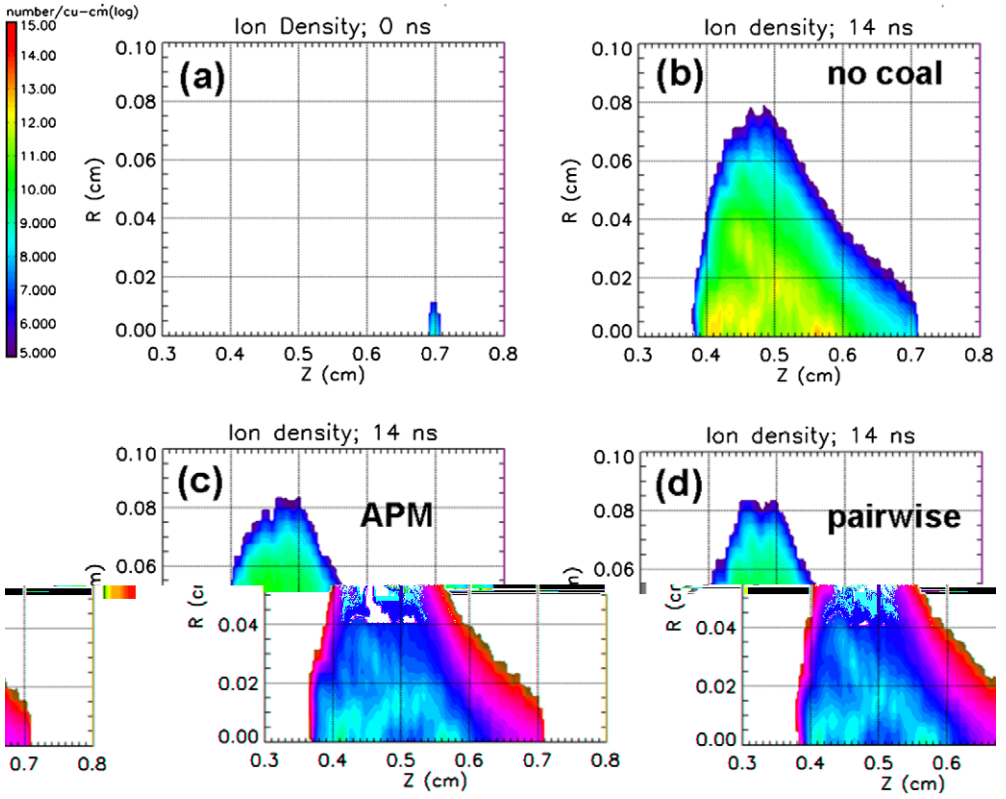

Fig. 2. The plasma ion density at (a) $t = 0$, (b) $t = 14$ ns for the simulation without coalescence, (c) $t = 14$ ns with APM, and (d) $t = 14$ ns with pairwise coalescence.

3×10^{11} cm^{-3} is problematic. In these simulations, we reduce the rate of new particle production by stipulating that an ionization probability $P = f n_{\text{gas}} \sigma_{\text{ion}} v_e \Delta t$ (n_{gas} is the neutral density, σ_{ion} is the ionization cross section for the electron of speed v_e and Δt is the simulation time step) and that the charge of the ionization pair is $q_{\text{ion}} = |q_e|/f$, where q_e is the impacting macroparticle electron charge. This is a reasonable approximation if $n_e \sigma_{\text{ion}} v_e \Delta t \ll 1$. For these calculations we take $f = 0.5$. In this problem, however, the particle number which begins at 10^4 reaches $>10^6$ after 14 ns. The initial ion density and density at 14 ns are plotted in Fig. 2 for three simulations: no coalescence, APM, and pairwise coalescence. The streamer propagates towards the anode (from right to left) due to the electron drift. We find that by 14 ns the streamer front in all three cases has moved 3 mm and the density has exceeded 10^{14} cm^{-3}. Thus, the two coalescence methods give satisfactory agreement with the "no coalescence" simulation and result in minimal differences to the underlying physical processes.

The basic timing information of the three simulations is summarized in Table 1. The target replacement numbers per cell, $M = 225$ in the APM and $M = 200$ in the pairwise replacement, were chosen to give roughly the same time-dependence of the macroparticle number over the entire simulation volume as seen in Fig. 3. For a parallel simulation with four processors, both coalescence schemes dramatically reduced computational runtime. The overall runtime to 14.6 ns for the APM simulations was 10% of the runtime of the simulation

Table 1
Timing information of the streamer propagation simulations performed on four processors

Coalescence method	Target number	Success rate	Run time to 14.6 ns (h)
None	NA	NA	11.90
APM (discretized)	225	0.978	1.10
Pairwise (0.005 energy tolerance)	200	0.39	1.78

with no coalescence. The pairwise method was 16% that of the no-coalescence simulation. The pairwise method is somewhat slower because each particle is compared with all other particles in the cell to find suitable pairs. The computation time thus scales nearly as the square of the macroparticle number. In Table 1, the coalescence success rate is defined as the ratio of the number of successful particle coalescences to the number of particles attempted. Note the much lower success rate of the pairwise method despite the higher 0.005 energy tolerance on matched particles. The tolerance for the APM method was 10^{-6}. In the APM method, the lowest acceptable replacement particle charge was 0.2 of the mean (this test caused 67% of the failures) and the highest acceptable particle energy was 1.5 times the maximum of the original set (this test caused 33% of the failures).

3.3. Relativistic electron-beam driven gas breakdown

In this test, we look at electron-beam driven gas breakdown. Paraxial electron diodes with gas transport cells have been used to focus intense electron beams onto a high-atomic-number target producing bremsstrahlung radiation [9]. Ideally, the self fields of the electron beam are completely neutralized by the gas conductivity. In practice, direct impact ionization by the beam and avalanche from the electron secondaries drive a breakdown of the gas that rapidly increases the plasma conductivity. Because of the delay in the conductivity rise and incompleteness of the breakdown, gas-filled focusing cells typically operate with a small but finite and slowly increasing net current (sum of the plasma and beam currents) [8,10]. This non-ideal effect results in an axial sweep of the beam focal position away from the bremsstrahlung converter at the end of the gas cell.

In this test, a 35-kA, 10-MeV (peak parameters) electron beam is produced in a diode with a 1-cm radius spherical cathode with a 5-cm AK gap as shown in Fig. 4. The voltage in the diode rises in 20 ns to the 10-MV flat top. Due to the electromagnetic fields in the diode AK gap, the beam at peak voltage enters the gas cell with a nearly uniform density, a 17-mm outer radius and a 10-cm focal length (the gas cell is thus also 10 cm in

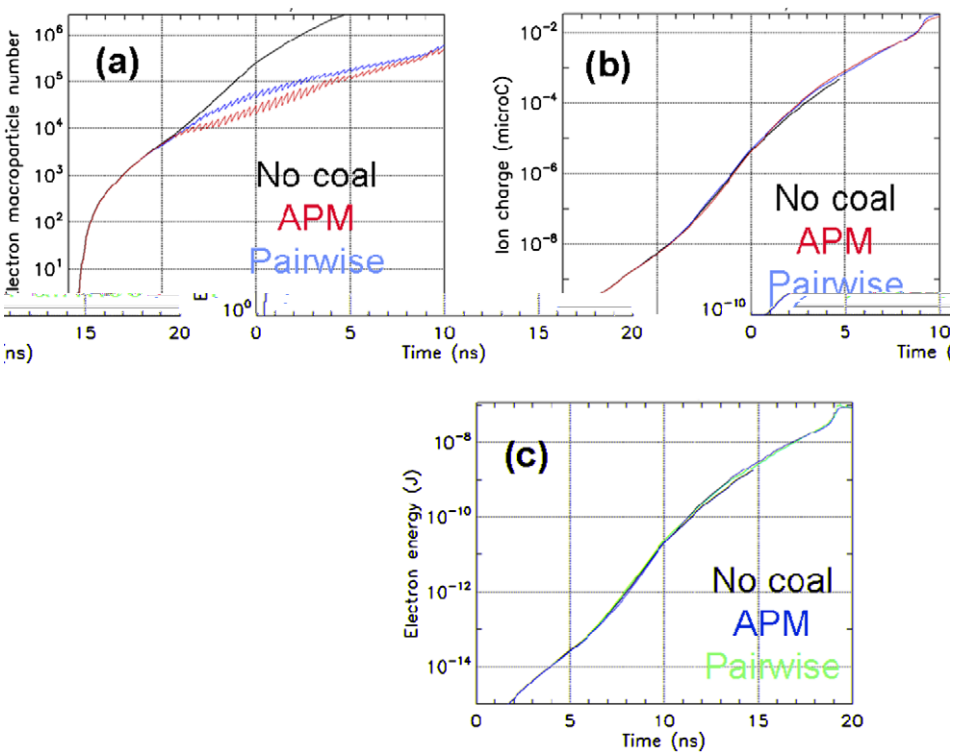

Fig. 3. The (a) electron macroparticle number, (b) total ion charge, and (c) total electron energy in the simulation are plotted for the three streamer propagation simulations.

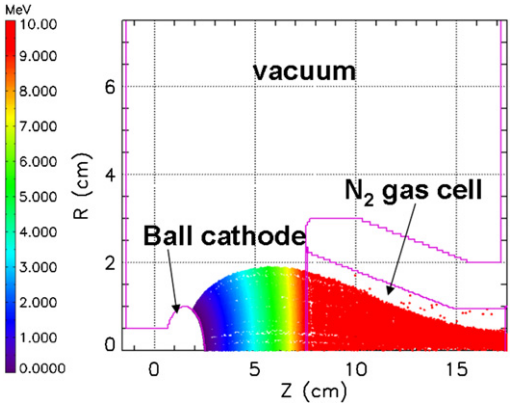

Fig. 4. The diode and gas cell configuration for 1-torr nitrogen is shown for the electron beam focusing test at $t = 20$ ns. The diode voltage rises in 20 ns to a 10-MV peak producing a 35-kA converging electron beam. Individual macroparticles are color-coded to their total energy in MeV.

length). The beam first passes through a 250-μm-thick Al anode foil which elastically scatters and slows the electrons.

The gas breakdown evolution is again studied with implicit kinetic PIC simulation that includes energy-dependent cross sections for electrons interacting with an initially neutral nitrogen gas at 1-torr pressure.

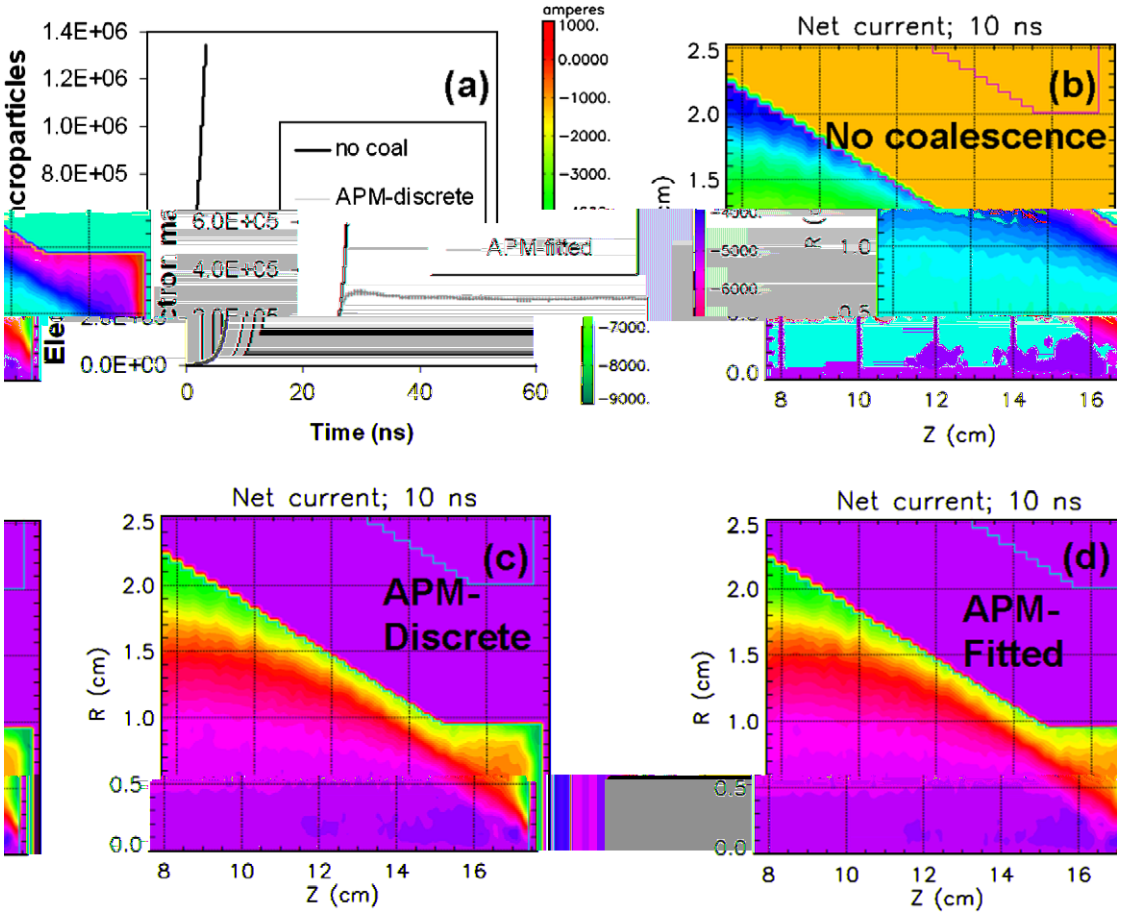

Fig. 5. The net current after 10 ns for the (b) no coalescence, (c) APM-discrete momentum distributions, and (d) APM-fitted distributions are plotted. In (a), the electron macroparticle number is plotted for the three simulations.

Because the charge neutralization is nearly perfect, minimizing the net current (sum of the beam and plasma return current) is key to the performance of the gas cell. We compare three simulations: a case with no coalescence, one with APM but using the discrete particle distribution (Eq. (1.21)) for the replacement momentum, and one with APM but using the fitted momentum distribution (Eqs. (1.16)–(1.20)). In both APM simulations, the coalescence was triggered by particle-per-cell number $N > 80$ with $M = 49$ target number per cell. In Fig. 5(a), we see that the electron macroparticle number exceeds 10^6 within 10 ns without coalescence. In contrast, the two APM simulations limit the number to $<3 \times 10^5$ for the entire simulation. The net currents after 10 ns are plotted in Fig. 5 showing good agreement between all three simulations after 10 ns (good agreement was found between the two APM simulations for all times indicating the fitted distributions were sufficiently accurate). The net current structure in the gas cell center as well as the unneutralized sheath at the wall is reproduced. The success rate for coalescence was 80% for both APM simulations.

We attempted to use the pairwise method on this problem as well. Within 10 ns, the simulation became numerically unstable. The instability was triggered by the charge correction calculation required by the pairwise method that introduced significant noise into the simulation and disrupted the physics. We were able to avoid this problem by turning off the correction and ignoring the charge conservation errors. The simulation results in this case were comparable to the previous test in both accuracy and efficiency (APM method again being more efficient).

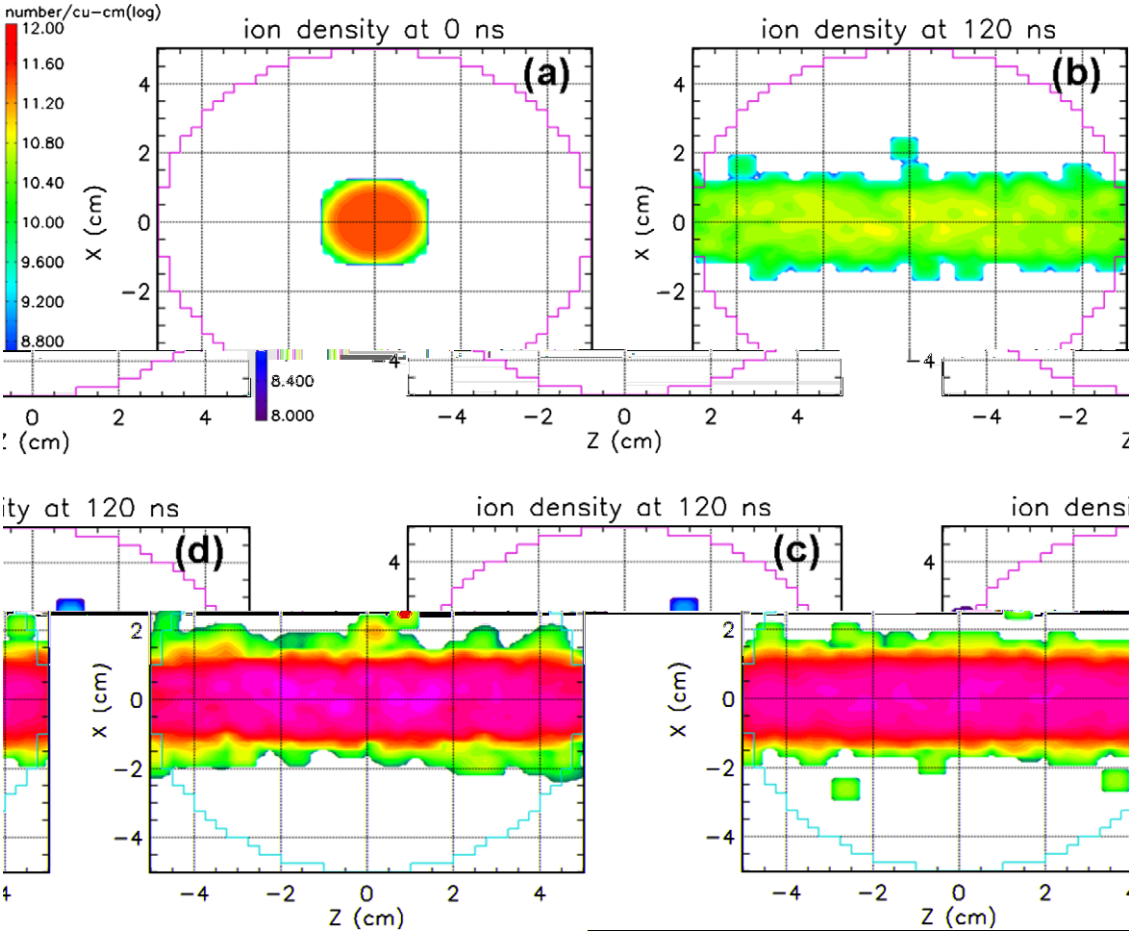

Fig. 6. The initial proton density for the three simulations of an expanding plasma in a 1-torr axial magnetic field is shown in (a). The proton density after 120 ns for the (b) "no fission-small", (c) the "no fission-large", and (d) the "fission" simulations are also plotted.

3.4. Magnetically confined plasma with APM particle fission

In the final test, we examine a low-β plasma (here, β is the usual ratio of thermal to magnetic pressure) confined in a 1-T axial magnetic field. Shown in Fig. 6(a), the 1-cm radius, 1-keV, 10^{12} cm^{-3} density plasma is loaded at rest immersed in the B_z field. We simulate the plasma in a 5-cm cylindrical tube with 2D Cartesian coordinates with an energy conserving but explicit particle push. The plasma skin depth (5 mm) is coarsely resolved with a 2.5-mm grid size in both directions. The simulations have time step $\omega_p \Delta t = 0.2$ and were run for 120 ns. Because the plasma expands into vacuum along the field lines, the number of particles per cell decreases without APM. This case is ideal for testing the fission aspect of APM. We compare the results of three simulations. The first ("no fission-small") has no APM and initially 16 particles-per-cell for both electrons and protons. The second ("no fission-large") also has no APM but 144 particles-per-cell per plasma species. In the third simulation, labeled "fission", the APM algorithm attempts to maintain the $M = 14$ particles-per-cell per species and has total particle number comparable to the "no fission-large" simulation by 120 ns.

A comparison of the proton densities in the three simulations is shown in Fig. 6. We can see the "no fission-small" simulation results in a thinner spatial distribution because the tail of the proton energy distribution is underpopulated. The simulation with larger particle number more closely resembles the fission simulation. In Fig. 7(a), the proton macroparticle number versus time shows the increase in the macroparticles for the fission simulation. Also shown is the total simulation energy error (out of 0.0014 J) for the three simulations. Although all three have energy error of order 0.1% after 120 ns, the fission case and the "no fission-large" are again more closely paired indicating a roughly 3× smaller error than the "no fission-small" simulation.

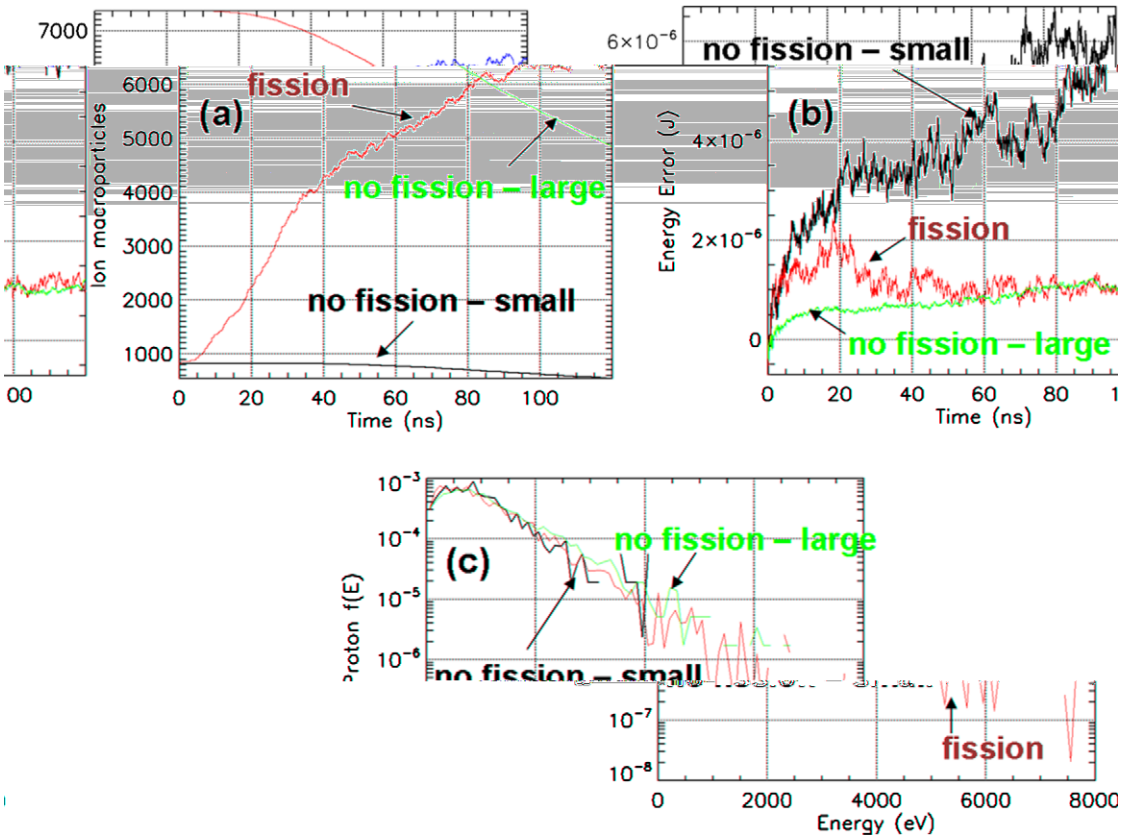

Fig. 7. For the immersed expanding plasma simulations, (a) the macroparticle number, (b) simulation energy error out of a total 0.0014 J, and (c) the proton energy distribution after 120 ns are shown. The three cases refer to no APM fission with a small initial particle number (830), no fission with a large number (7380), and APM fission beginning with the small number.

Finally, in Fig. 7(c), the proton energy distribution after 120 ns is shown for the three simulations. The distribution in the fission simulation evolves a higher energy tail similar to the "no fission-large" simulation consistent with the broader spatial wings shown in Fig. 6. Thus, the APM algorithm when used to produce more particles can be effective at maintaining good particle statistics. One caveat is that when applying this technique to the electrons in this particular problem, the electron velocity distribution perpendicular to B_z does skew from the other two simulations. This is due to the disparity in the electron cyclotron radius (0.007 cm) compared with the simulation cell size. Thus, applying the APM to particle species with spatially under-resolved cyclotron orbits can lead to errors.

4. Conclusions

We have demonstrated a practical particle management scheme that is useful for problems where the momentum distribution function shape is important. The APM algorithm, which builds on the framework of Ref. [4], was tested in two stressing gas breakdown problems and a plasma expansion problem and proved to be efficient and to accurately reproduce the momentum distribution functions given adequate spatial resolution. To the times when the simulation results could be compared with and without coalescence, the APM enabled $>10\times$ faster computation speed without sacrificing accuracy. For decreasing particle number, the APM method is more efficient and reliable than the pairwise method in the two test problems considered here. The method can also be used to increase particle number for problems that require better statistics such as in the expanding plasma simulation. The ability to keep the particle number-per-cell nearly constant provides for more efficient parallel computation.

Acknowledgments

Work supported by Sandia National Laboratories through the US Department of Energy. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, Adam Hilger, New York, 1991.
[2] R.W. Hockney, J.W. Eastwood, Computer Simulation using Particles, Adam Hilger, New York, 1988.
[3] G. Lapenta, J.U. Brackbill, J. Comput. Phys. 115 (1994) 213–217.
[4] F. Assous, T. Pougeard Dulimbert, J. Segre, J. Comput. Phys. 187 (2003) 550.
[5] LSP is a software product of ATK-Mission Research, Albuquerque, NM 87110.
[6] Yu.D. Korolev, G.A. Mesyats, Physics of Pulsed Breakdown in Gases, Ural Division of the Russian Academy of Sciences, Yekaterinbutg, 1998.
[7] D.R. Welch, D.V. Rose, M.E. Cuneo, R.B. Campbell, T.A. Mehlhorn, Phys. Plasmas 13 (2006) 063105.
[8] D.R. Welch, D.V. Rose, B.V. Oliver, E. Schamiloglu, K. Hahn, J.E. Maenchen, Phys. Plasmas 11 (2004) 751.
[9] T.J. Goldsack et al., IEEE Trans. Plasma Sci. 30 (2002) 239.
[10] B.V. Oliver, D. Short, G. Cooper, J. McLean, J. O'Malley, IEEE Trans. Plasma Sci. 33 (2) (2004) 704–711.