Journal of Computational Physics **176,** 402–429 (2002)
doi:10.1006/jcph.2001.6994, available online at http://www.idealibrary.com on IDEAL®

A Numerical Method to Simulate Radio-Frequency Plasma Discharges

E. P. Hammond,* K. Mahesh,† and P. Moin*

Department of Mechanical Engineering, Stanford University, Stanford, California 94305; and †Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455

Received January 2, 2001; revised December 10, 2001

A fully conservative and efficient numerical algorithm is developed for fluid simulations of radio-frequency plasma discharges. Results are presented in one and multiple dimensions for a helium discharge. The algorithm produces accurate results even on fairly coarse grids without the use of numerical dissipation. The proposed electron flux discretization is more accurate and efficient than two of the most commonly used discretizations: low-order upwinding (M. S. Barnes, T. J. Colter, and M. E. Elta, 1987, *J. Appl. Phys.* **61**, 81) and Scharfetter–Gummel (D. L. Scharfetter and H. K. Gummel, 1969, *IEEE Trans. Electron Devices* **ED-16**, 64).

Key Words: plasma; numerical method; conservative; nondissipative.

1. INTRODUCTION

The development of numerical models to simulate processing plasmas has been ongoing for over two decades, with the most rapid growth occurring during the past decade. Prior to 1990, simulations were typically one dimensional, either with a fluid model or a particle model of the plasma; since then, two- and three-dimensional models with detailed chemistry have been developed [1]. Some examples of two-dimensional simulations can be found in [2]–[7]; three-dimensional simulations are more rare, but [8]–[10] contain representative calculations. Numerical tools are actually being used by industry as part of the design process [1].

Our objective is to develop a robust, accurate method to simulate plasma discharges. We have included conservative and nondissipative concepts from the numerical methods used to solve the Navier–Stokes equations, which have been much more extensively studied. Nitschke and Graves developed a one-dimensional fluid model for a helium glow discharge [11]. They compared results from particle and fluid simulations for the same model system based on helium. The rate coefficients for the collision terms in the fluid equations were based on numerical integration of the collision cross sections with an assumed Maxwellian velocity distribution. They found that the two methods agree well for pressures greater than or equal to 100 mTorr. Using their relatively simple fluid model, we sought an efficient numerical method for the robust and rapid solution of the equations.

As with Nitschke and Graves [11], this paper examines a low-pressure, low-temperature, glow discharge helium plasma with a very small ionization fraction ($\approx 10^{-8}$). The plasma is maintained in a capacitively coupled reactor; this type of reactor is frequently used for materials processing purposes. As will be seen, even this one-dimensional problem poses significant challenges: It has a wide range of time scales, steep unsteady gradients in charged particle densities, and near-zero electron densities in the sheaths. These problems exist today, and work continues to be published that presents one-dimensional, radio-frequency plasma simulation methods [12].

Robustness is commonly introduced into the numerical solution of the plasma equations by upwinding the spatial discretization, in particular, the discretization of the electron flux. One method of upwinding the electron flux is described by Barnes *et al.* in [13]. This technique is used by Nitschke and Graves [11]. Another method used to alleviate the stiffness due to the electron flux is discussed by Scharfetter and Gummel in [14].

As will be shown, low-order upwinding on coarse grids can provide results that are quite far from the grid converged solution, almost a factor of two. Accuracy on coarse grids is crucial for advancement into multiple dimensions, particularly when complex chemistries are used. When large errors accrue due to the numerical method, modelers run the risk of having the numerical errors exceed the errors from the approximations in the model. Although Scharfetter–Gummel is reasonably accurate on coarse grids, it introduces another approximation into the fluid model. Later, this additional approximation will be shown to have nearly a 20% impact on the plasma density. Furthermore, the Scharfetter–Gummel discretization leads to more expensive computations, which will also be shown.

The equations used are described in Section 2, and the approximations used in developing the equations are explained. A conservative, time-accurate algorithm that provides rapid solutions to the plasma fluid equations was developed; care was taken to ensure that the algorithm is not numerically dissipative. This discretization of the governing equations is described in Section 3. The derivation of the conservative discretization of the electron flux and the solution it provides, and the results of alternative discretizations, are described in Section 4. The validation against previously published simulations by Nitschke and Graves [11] is also discussed in Section 4. The extension of the algorithm to multiple dimensions and a sample two-dimensional result are presented.

2. MODEL EQUATIONS

Generally speaking, processing plasmas are sustained by the highly energetic electrons which collide with the background gas to generate ions and more electrons. Power is input to the electrons in a capacitive discharge via the applied radio frequency voltage. Since the characteristic frequency of the helium ions ($\omega_i/2\pi = \sqrt{e^2 n_i/\epsilon_0 m_i}/2\pi \approx 3$ MHz for $n_i = 10^{15}$ m^{-3}) is lower than the radio frequency (on the order of 10 MHz), the ion density and velocity do not change appreciably during each period. In contrast, the characteristic frequency of the electrons ($\omega_e/2\pi = \sqrt{e^2 n_e/\epsilon_0 m_e}/2\pi \approx 300$ MHz for $n_e = 10^{15}$ m^{-3}) is much higher than the radio frequency. As a result, the electrons are subject to significant changes in each period. On the whole, the plasma is electrically neutral, with thin sheaths forming near the boundaries. These sheaths form because the lighter electrons quickly diffuse away, which creates a net positive charge and an electric field that acts to prevent additional electrons from escaping and to accelerate the ions. Sheaths in a capacitive radio-frequency discharge are extremely unsteady, and their maximum thickness is generally a few tens of Debye lengths ($\lambda_D = \sqrt{\epsilon_0 k_B T_0 / e^2 n_0} \approx 0.5$ mm for $n_0 = 10^{15}$ m^{-3} and $T_0 =$ 45,000 K) A more complete description of the physics of a plasma discharge may be found in references [15] and [16].

The fluid equations for a plasma are derived from the Boltzmann equation for each species by specifying the velocity distribution function and by taking velocity moments of the Boltzmann equation. This yields conservation equations for the mass, momentum, and energy of each species. The collisional terms are typically modeled by assuming a Maxwellian velocity distribution function. Viscosity plays a negligible role, and its effect is usually not included in the momentum equations. Maxwell's equations are solved to determine the electric and magnetic fields generated by the charged particles and their motion. In the absence of an externally imposed magnetic field, magnetic effects are generally quite small, and the plasma is usually treated as electrostatic, where the electric field is the gradient of a potential, $\vec{E} = -\nabla\Phi$.

In this paper, three species are considered: the background gas (He), the ions (He^+), and the electrons. The background gas is assumed to be stagnant and at constant temperature and pressure. Since the ion mass is very nearly the same as the mass of the background gas, the ions are assumed to be in thermal equilibrium with the background, and no energy equation is solved for the ions. Due to their lower mass, the electrons exchange kinetic energy very poorly with the other particles, and their temperature can be significantly higher than the other species. Thus, it is important to solve for the electron energy. A commonly used simplification to the electron momentum equation is the "drift–diffusion approximation." The electron inertial terms, i.e., those in the material derivative, are neglected. This results in a balance among the Lorentz force, the pressure gradient, and the drag from collisions with the background gas which yields an algebraic expression for the electron flux.

Justification for the particular form of the fluid conservation equations used is provided by [17]. Surendra and Dalvie [17] take velocity moments of a particle-in-cell Monte Carlo collision calculation of a model plasma to determine the variation in space and time of each term in the conservation equations. For sufficiently high pressures (i.e., high collision frequency), they find that the inertial terms in the electron momentum equation are negligible compared to the pressure gradient, Lorentz force, and collisional drag, except in the sheath regions. For the ion momentum equation, the inertial terms, Lorentz force, and collisional drag are all significant while the ion pressure gradient is essentially negligible. They also find that at high enough pressures, the ion temperature is roughly equivalent to the background gas temperature, except in the sheaths. The implication of the fluid model approximations on accuracy in the sheath regions is not certain, but densities are generally low in the sheaths (particularly for the electrons), so the impact is not expected to be large.

2.1. Governing Equations

The governing equations are taken from those used by Nitschke and Graves [11], and they are nondimensionalized prior to solution. The plasma bulk is used to define reference values for the charged particle densities (n_0) and the electron temperature (T_0); for the simulations discussed in this paper, these values are 1.0×10^{15} m^{-3} and 45,000 K. The Debye length and plasma frequency of the electrons are used as reference length and time scales, respectively. The variables n_e and n_i denote the dimensionless electron and ion number densities, respectively, and $\vec{v}_i$ and $\frac{3}{2}n_eT_e$ represent the dimensionless ion velocity and the electron thermal energy density. The electron number density flux and thermal flux are represented by $\vec{j}_e$ and $\vec{q}_e$. The density of the neutrals is denoted by N, and Φ and $\vec{E}$ denote the electric potential and electric field, respectively. Values for the constants E_a, σ_{cx}, k_{i0}, k_{mt}, and h_{iz} can found in [11]. Although the pressureless form of the ion equations used here can cause concentration effects, problems have not been observed. Simulations performed with the ion pressure gradient were negligibly different from those without. The nondimensional form of the equations, written for three dimensions, is presented below along with the nondimensionalized parameters:

$$\frac{\partial n_e}{\partial t} + \nabla \cdot \vec{j}_e = p_1 n_e e^{-p_2/T_e} \tag{1}$$

$$\vec{j}_e = -p_7[n_e\vec{E} + \nabla(n_eT_e)] \tag{2}$$

$$\frac{\partial n_i}{\partial t} + \nabla \cdot (n_i\vec{v}_i) = p_1 n_e e^{-p_2/T_e} \tag{3}$$

$$\frac{\partial \vec{v}_i}{\partial t} + \vec{v}_i \cdot \nabla\vec{v}_i = p_3\vec{E} - p_4|\vec{v}_i|\vec{v}_i \tag{4}$$

$$\frac{\partial\left(\frac{3}{2}n_eT_e\right)}{\partial t} + \nabla \cdot \vec{q}_e = -\vec{j}_e \cdot \vec{E} - p_5 n_e e^{-p_2/T_e} - p_6 n_e (T_e - T_{\text{neut}}) \tag{5}$$

$$\vec{q}_e = \frac{5}{2}\vec{j}_eT_e - p_8 n_e T_e \nabla T_e \tag{6}$$

$$\nabla^2\Phi = -(n_i - n_e) \tag{7}$$

$$\vec{E} = -\nabla\Phi \tag{8}$$

$$\begin{aligned} p_1 &= Nk_{i0}2\pi/\omega_e & p_2 &= E_a/k_BT_0 \\ p_3 &= 4\pi^2 m_e/m_i & p_4 &= N\sigma_{cx}\lambda_D\pi/2 \\ p_5 &= h_{iz}Nk_{i0}2\pi/k_BT_0\omega_e & p_6 &= 3m_eNk_{mt}2\pi/m_{\text{neut}}\omega_e \\ p_7 &= 2\pi\omega_e/k_{mt}N & p_8 &= 5p_7/2. \end{aligned}$$

2.2. Boundary Conditions

A characteristic analysis of the one-dimensional form of the governing equations was performed to determine the number of boundary conditions required. For the electron equations, the analysis showed that boundary conditions should be specified on both sides of the computational domain. The boundary conditions used in this paper are the same as those used by Nitschke and Graves [11]; they are applied on both sides of the domain:

$$j_e = \mp p_9 n_e \sqrt{T_e}\ (-\text{ at } x = 0, + \text{ at } x = L) \tag{9}$$

$$p_9 = \frac{n_0}{4}\sqrt{\frac{8k_BT_0}{\pi m_e}}(1 - \Theta)$$

$$q_e = \frac{5}{2}j_ek_BT_e\ (\text{at } x = 0 \text{ and } L). \tag{10}$$

The boundary fluxes for the electron continuity equation (1) are assumed to correspond to the one-way flux for particles with a Maxwellian velocity distribution. Reflection is accounted for with Θ; a value of 0.25 was used by Nitschke and Graves [11]. The thermal flux at the boundaries is equated to the enthalpy flux. A boundary condition for the energy equation (5) that is more consistent with the boundary condition for the continuity equation is to equate q_e to the one-way flux of kinetic energy for a Maxwellian distribution, $\frac{4}{3} \times \frac{3}{2} k_B T \times j_e$, or $q_e = 2 j_e k_B T_e$. Our simulations showed little difference between the two thermal boundary conditions; however, for consistency with Nitschke and Graves, the enthalpy flux boundary condition (10) is used in this paper. Secondary electron emission is neglected.

Characteristic analysis of the ion equations indicates that no boundary conditions should be specified. The characteristics point in the direction of v_i, and, since E is generally directed out of the plasma close to the walls, the positive ions always flow out of the computational domain. Thus, no boundary conditions are specified for the ion equations. Nitschke and Graves applied $\partial n_i / \partial x = 0$ on both boundaries, and set $\partial v_i / \partial x = 0$ on one boundary [11]. When these boundary conditions were used, small oscillations in the ion number density were observed at both boundaries. When the boundary conditions were removed, so were the oscillations. Results demonstrating this will be shown later.

The boundary conditions for the Poisson equation for Φ are a sinusoidal radio-frequency voltage at $x = 0$ and ground at $x = L$. The amplitude of the voltage is 500 V, and the frequency is 12 MHz. The spacing between the electrodes varied with the pressure of the background gas and was between 4 and 12 cm. These conditions were chosen to correspond with the calculations of Nitschke and Graves [11].

3. NUMERICAL METHOD

The dependent variables are discretized on a nonuniform, staggered grid. A staggered grid was chosen because of the conservative properties it provides for the nonlinear fluid equations [18]. Figure 1 illustrates the staggered positioning of variables. The equations governing n_e and $n_e T_e$ are quite stiff, and an implicit time advancement scheme is therefore necessary. Furthermore, the accuracy and stability of the electron equations depend critically on the discretization of the flux, j_e. The ion equations do not exhibit such stiffness, and an explicit time advancement scheme is adequate. The spatial and temporal discretizations are described in detail below.

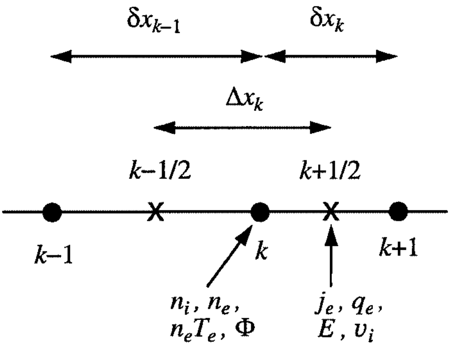

FIG. 1. Storage locations of variables on the nonuniform, staggered grid; the ×'s are located halfway between the ●'s.

3.1. Spatial Discretization

The discrete continuity equations and the electron energy equation were derived using the finite volume method. Since the ion momentum equation (4) was not written in conservative form, the discrete ion momentum equation was derived using second-order finite differences on a nonuniform grid. Simulations were performed using the conservative form of the ion momentum equation, and the results were in agreement. For consistency with Nitschke and Graves [11], the primitive form is used throughout. The following semidiscretized equations were used to advance the dependent variables.

Ion continuity:

$$\frac{dn_{i,k}}{dt} = -\frac{1}{\Delta x_k}\left[\left(\frac{n_{i,k}+n_{i,k+1}}{2}\right)v_{i,k+1/2} - \left(\frac{n_{i,k-1}+n_{i,k}}{2}\right)v_{i,k-1/2}\right] + p_1 n_{e,k} e^{-p_2/T_{e,k}}. \tag{11}$$

Ion momentum:

$$\begin{aligned}\frac{dv_{i,k+1/2}}{dt} = -\frac{v_{i,k+1/2}}{\Delta x_{k-1}+\Delta x_k}&\left[\frac{\Delta x_{k-1}}{\Delta x_k}v_{i,k+3/2} - \left(\frac{\Delta x_{k-1}}{\Delta x_k} - \frac{\Delta x_k}{\Delta x_{k-1}}\right)v_{i,k+1/2}\right.\\ &\left.-\frac{\Delta x_k}{\Delta x_{k-1}}v_{i,k-1/2}\right] + p_3 E_{k+1/2} - p_4\left|v_{i,k+1/2}\right|v_{i,k+1/2}.\end{aligned} \tag{12}$$

Electron continuity:

$$\frac{dn_{e,k}}{dt} = -\frac{1}{\Delta x_k}\left(j_{e,k+1/2} - j_{e,k-1/2}\right) + p_1 n_{e,k} e^{-p_2/T_{e,k}}. \tag{13}$$

The discretization of j_e significantly impacts the robustness and accuracy of the solution and is discussed in detail later.

Electron energy:

$$\begin{aligned}\frac{3}{2}\frac{dn_eT_{e,k}}{dt} &= -\frac{1}{\Delta x_k}\left(q_{e,k+1/2} - q_{e,k-1/2}\right) - \frac{1}{2}\left(j_{e,k-1/2}E_{k-1/2} + j_{e,k+1/2}E_{k+1/2}\right)\\ &\quad - p_5 n_{e,k} e^{-p_2/T_{e,k}} - p_6 n_{e,k}(T_{e,k} - T_{\text{neut}})\\ q_{e,k+1/2} &= \frac{5}{2} j_{e,k+1/2}T_{e,k+1/2} - p_8\left(\frac{n_eT_{e,k+1} + n_eT_{e,k}}{2}\right)\left(\frac{T_{e,k+1} - T_{e,k}}{\delta x_k}\right)\\ T_{e,k+1/2} &= \left(\frac{T_{e,k} + T_{e,k+1}}{2}\right).\end{aligned} \tag{14}$$

Poisson's equation:

$$\frac{\Phi_{k+1} - \Phi_k}{\delta x_k} - \frac{\Phi_k - \Phi_{k-1}}{\delta x_{k-1}} = -\Delta x_k(n_{i,k} - n_{e,k}) \tag{15}$$

$$E_{k+1/2} = -\frac{\Phi_{k+1} - \Phi_k}{\delta x_k}. \tag{16}$$

3.2. Discrete Boundary Conditions

At the boundaries, the domain is discretized as illustrated in Fig. 2. Since the electron density is not known at the edge, the electrons are assumed to obey the Boltzmann relation [15],

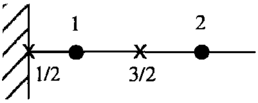

FIG. 2. The discretization of the computational domain at the left edge; the discretization at the right edge is similar. The storage of variables is the same as in Fig. 1.

and the electron density at the edge is given in nondimensional terms by

$$n_{e,1/2} = n_{e,1} \exp\left[\frac{\left(\Phi_{1/2} - \Phi_1\right)}{T_{e,1}}\right].$$

Thus, the electron flux at the left edge, for example, is

$$j_{e,1/2} = -p_9 n_{e,1} \exp\left[\frac{\left(\Phi_{1/2} - \Phi_1\right)}{T_{e,1}}\right]\sqrt{T_{e,1}}. \tag{17}$$

The ion flux at the wall is found by linearly extrapolating the ion density from the two grid points nearest the wall and then multiplying by the velocity at the wall. Since the ion momentum equation is evaluated at the wall, one-sided differences are taken for the spatial derivatives. If the ion velocity at the wall points into the domain, it is set to zero.

An earlier discretization of the domain at the boundary resembled Fig. 2 except that an additional node (●) was located at the "1/2." This provided the electron number density immediately at the edge for the flux boundary conditions, and it also gave the ion number density and velocity at the same location so that the ion flux out of the domain could be calculated directly. This earlier scheme provided the same results as the scheme described in Fig. 2 but was more cumbersome to implement (particularly in multiple dimensions).

3.3. Temporal Discretization

A standard, fourth-order Runge–Kutta scheme was used to advance the ion equations; implicit time advancement was used for the electrons. Both fully nonlinear and linearized time advancement methods for the electron equations were considered. The implicit Euler, trapezoidal, and second-order implicit Runge–Kutta methods were evaluated. The implicit Euler method was found to perform adequately. The trapezoidal method was unsatisfactory since oscillatory solutions were obtained unless very small time steps were used. The implicit Runge–Kutta scheme offered the greatest accuracy.

The implicit Runge–Kutta method is described below; it was adapted from the second-order scheme given by Zhong [19]. Some of the implicit Runge–Kutta methods discussed by Zhong are based on the Rosenbrock linearized methods in [20]. An ordinary differential equation $dy/dt = f(t, y)$ is discretized with the second-order implicit Runge–Kutta method as $y^{n+1} = y^n + \omega_1 \delta y_1 + \omega_2 \delta y_2$, where $\omega_1 = \omega_2 = 0.5$ and δy_1 and δy_2 are computed as

$$\begin{aligned}\delta y_1 &= \delta t f(t^n + \alpha_1 \delta t, y^n + a_{11}\delta y_1)\\ \delta y_2 &= \delta t f(t^n + \alpha_2 \delta t, y^n + a_{21}\delta y_1 + a_{22}\delta y_2),\end{aligned}$$

with $a_{11} = 1 - \sqrt{2}/2$, $a_{21} = \sqrt{2} - 1$, $a_{22} = 1 - \sqrt{2}/2$, $\alpha_1 = a_{11}$, and $\alpha_2 = a_{21} + a_{22}$. The time advancement can be linearized, yielding the equations for δy_1 and δy_2,

$$\delta y_1 = \delta t \left[f(t^n, y^n) + a_{11} \left.\frac{\partial f}{\partial y}\right|_{t^n, y^n} \delta y_1 \right]$$

$$\delta y_2 = \delta t \left[f(t^n + \alpha_2 \delta t, y^n + a_{21}\delta y_1) + a_{22} \left.\frac{\partial f}{\partial y}\right|_{t^n+\alpha_2\delta t, y^n+a_{21}\delta y_1} \delta y_2 \right],$$

with $a_{11} = 1 - \sqrt{2}/2$, $a_{21} = \sqrt{2} - 1$, $a_{22} = 1 - \sqrt{2}/2$, and $\alpha_2 = a_{21}$.

Some of the terms in the ion density and velocity equations depend on the electron density or energy either explicitly (the ionization source term) or implicity (the electric field). To simplify the ion time advancement, the ion equations used the values of electron density and energy and the electric field at the beginning of the time step, and the ions were advanced explicitly to the next time level needed for the implicit solution of the electrons. For example, for the simulations using implicit Euler, the ions were advanced from time t^n to t^{n+1} using n_e^n, T_e^n, and E^n. With the ions now given at t^{n+1}, the electrons were implicitly advanced to t^{n+1}. The electron and ion equations were fully coupled using the explicit/implicit advancement described in [19], and the results were equivalent to within the accuracy of the iterative solution of the implicit electron equations.

A Newton–Raphson iterative technique was used to solve the nonlinear equations resulting from the fully implicit formulations. Strictly speaking, the Jacobian for this system is nearly a full matrix, since both the electron continuity and energy equations involve the electric field, which is a function of the net charge density throughout the domain. Thus, for example, when starting from time level n and solving the nonlinear equations for the electron variables at time level $n+1$, the equations for $n_{e,k}^{n+1}$ and $n_e T_{e,k}^{n+1}$ are functions of $n_{e,1}^{n+1}, n_{e,2}^{n+1}, \ldots, n_{e,N-1}^{n+1}, n_{e,N}^{n+1}$. However, close examination of the Jacobian matrix reveals that the submatrices which are more than one block off the main diagonal have terms that are much smaller in magnitude than the submatrices on the main diagonal and those immediately adjacent to it. This indicates that, in the iterative solution of the system of equations, blocks not adjacent to the main diagonal have little impact on the solution at subiterations. Thus, the full Jacobian can be approximated by a block tri-diagonal Jacobian. The results obtained with the reduced Jacobian are virtually the same as the results with the full Jacobian, and the time savings is significant ($O(N)$ operations to invert the Jacobian matrix versus $O(N^2)$ with an iterative solver).

The integration time step is chosen so that the radio-frequency period is adequately resolved; this is the basic physical limit. With approximately 200 steps per period, which is generally what is required to achieve time step independence for the base case presented here, the CFL's for the ions are usually very low on a coarse 20-point grid, O(0.01), which indicates that ion convection is not limiting. The only real numerical limitation on the stability of the system is in the implicit advancement of the electron equations, ironically. Although the advancement is implicit, the root must be found to a very nonlinear system of equations, which is not always an easy task, much less guaranteed to be successful [21]. Robustness can be added to the root finding by using a modified Newton method; see [21] and [22]. But the computational overhead generally exceeds the time saved by using larger time steps. A more effective way to add robustness to the root finding is to advance equations for $\ln n_e$ and $\ln n_e T_e$, which can be derived by dividing Eqs. (13) and (14) by $n_{e,k}$ and $n_e T_{e,k}$, respectively; the root is then sought in terms of $\ln n_e$ and $\ln n_e T_e$.

The electrons can also be advanced in time with a linearized implicit algorithm. As before, the Euler method is more stable than the linearized trapezoidal method. The second-order linearized implicit Runge–Kutta scheme seems to offer the same stability as the linearized Euler, but with higher order accuracy. While cheaper per time step than the fully nonlinear formulation, the linearized algorithms require a smaller time step for accuracy. Further discussion of the merits of fully implicit versus linearized implicit time advancement when applied to a highly nonlinear ordinary differential equation can be found in Appendix A.

The equations were advanced in time until the relative change in the maximum ion density from one period to the next was less than 10^{-5}. No acceleration was used to speed the evolution to steady state. Typically, 500 periods were needed to reach steady state when starting with a uniform profile for the densities and zero velocity for the ions.

4. RESULTS

The stiff nature of the electron equations makes even one-dimensional simulations challenging. The spatial and temporal variation in n_e and n_eT_e depend on the exact conditions imposed, but the electron densities within the sheaths can change by dozens of orders of magnitude over the course of a radio-frequency period. The ion density and velocity are much better behaved, and they generally change by no more than an order of magnitude throughout the domain during a period.

4.1. Resolution of Numerical Methods

In this section, the consistency and accuracy of the various methods for discretizing the electron flux will be shown. Also, the results of the different time advancement methods will be discussed. All the computations are run at the same condition as Case I in reference [11]: helium gas at 250 mTorr and 300 K, applied radio-frequency voltage of 500 V at 12 MHz, and a gap spacing of 4 cm.

4.1.1. *Standard, Central-Difference Flux*

The most straightforward discretization of j_e using central differences is as follows:

$$j_{e,k+1/2} = -p_7\left[\left(\frac{n_{e,k}+n_{e,k+1}}{2}\right)E_{k+1/2} + \frac{n_eT_{e,k+1} - n_eT_{e,k}}{\delta x_k}\right]. \tag{18}$$

While this is a natural approach, it requires very fine grids to capture the wide spatial variations in n_e and n_eT_e. Even with a tanh-stretched grid, about 400 grid points are required. At this resolution, the minimum grid spacing (at the edges of the domain) is about 3% of a Debye length, and the maximum (in the center of the domain) is nearly half a Debye length. Using fewer grid points results in unphysical negative values for n_e or n_eT_e in the sheath regions (where these two variables become extremely small), which makes the computation unstable.

Simulation results using 400 points on a stretched grid and 250 time steps per radio-frequency period (T) with the fully implicit second-order Runge–Kutta method are shown in Fig. 3. This computation required approximately 2.5 h to run for 500 periods on a Silicon Graphics Iris Indigo 2 workstation (195 MHz processor). Calculations with a refined grid (800 points) and with a refined time step (500 steps per period) are also shown in Fig. 3.

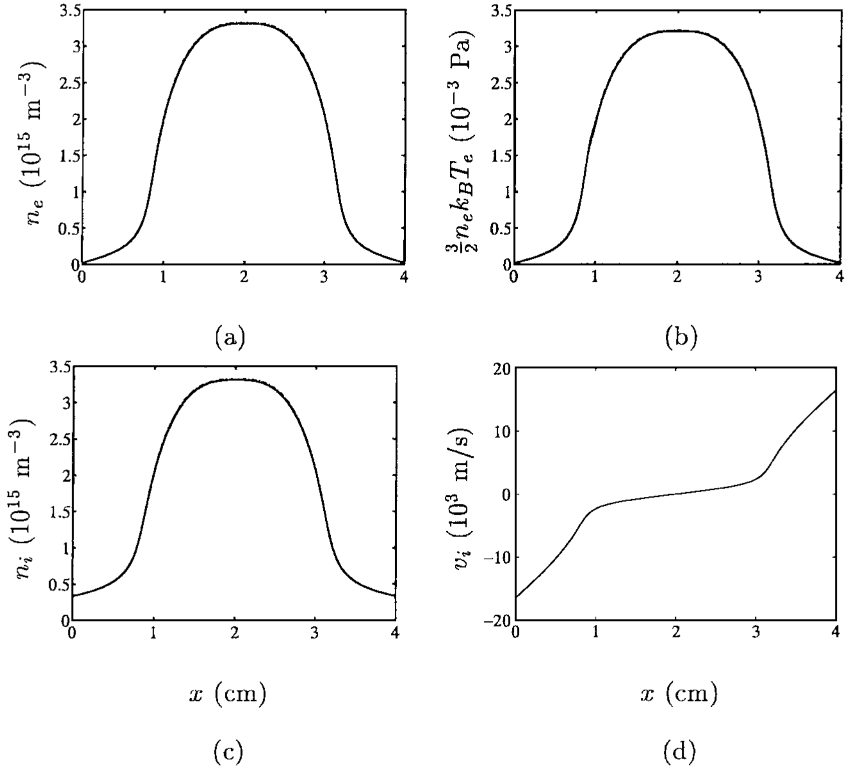

FIG. 3. The period-averaged profiles of: (a) electron density, (b) electron energy density, (c) ion density, and (d) ion velocity. Three different simulations at the conditions of Case I in reference [11] were performed to establish convergence with respect to grid and time step: 800 points, $\delta t = 0.004$ T (—); 400 points, $\delta t = 0.004$ T (— —); 400 points, $\delta t = 0.002$ T (—·).

The change in the peak plasma density due to the change in grid or time step is less than a percent. The result with 400 points and $\delta t = 0.004$ T will be referred to as the "exact" result. The solution at these conditions will be used as a benchmark to evaluate the other discretizations discussed below. The drastic variation of the electron density during the radio-frequency period is shown in Fig. 4.

The need for approximately 400 grid points and the runtime of over 2 h make three-dimensional solutions of this type of problem impractical with this discretization. Assuming that roughly 100 grid points would be needed for the other two dimensions means that a three-dimensional calculation would take 10,000 times longer, or almost three years, on the same computer, assuming (optimistically) that the computation time would still depend linearly on the number of grid points.

4.1.2. *Upwinded Flux*

Upwind biasing the spatial discretization of the electron flux (and sometimes the ion flux) is used almost universally to circumvent the stability problems posed by steep gradients at the sheath edges and very low electron densities (see, for example, Barnes *et al.* [13], Sommerer and Kushner [23], Gogolides *et al.* [24], Kushner *et al.* [8], and Colella *et al.* [25]). However, some simulations have been published with a nondissipative discretization for the

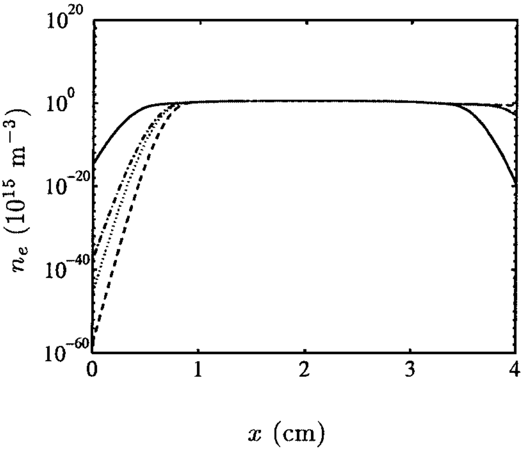

FIG. 4. The variation in the instantaneous electron number density at four different times in the radio-frequency period for the "exact" solution shown in Fig. 3: t = 0.5 T (—), t = 0.625 T (-·-), t = 0.75 T (- - -), and t = 0.875 T (· · ·).

electron and ion equations. Huppert *et al.* [26] used Gauss–Lobatto–Legendre polynomials as the basis functions for the finite elements in their spatial discretization and found that upwinding affected the peak density by approximately 30%.

We evaluated a first-order upwinded approach similar to that used in Barnes *et al.* [13]. The sign of the electric field at $x_{k+1/2}$ determines the discretization of the drift component of the electron flux. All the other variables are calculated in the same fashion as before:

$$j_{e,k+1/2} = \begin{cases} -p_7\left[n_{e,k}E_{k+1/2} + \frac{n_eT_{e,k+1} - n_eT_{e,k}}{\delta x_k}\right], & \text{if } E_{k+1/2} < 0 \\ -p_7\left[n_{e,k+1}E_{k+1/2} + \frac{n_eT_{e,k+1} - n_eT_{e,k}}{\delta x_k}\right], & \text{if } E_{k+1/2} > 0. \end{cases} \tag{19}$$

Although more robust, this approach is not as accurate as the central-difference technique discussed previously. For the same stretched grid as the "exact" calculation, the upwinded calculation still shows grid dependence; see Fig. 5. Not surprisingly, the effects of upwinding are most noticeable in the electron quantities; the upwinding only indirectly affects the ions. The period-averaged ion density simply reflects the increased average electron density, and the ion velocity is hardly affected.

4.1.3. *Scharfetter–Gummel*

The Scharfetter–Gummel discretization [14] is also a popular method to overcome the stiffness in the drift-diffusion form of the electron flux. Some examples of its use can be found in [2, 4, 27–29]. The nondimensional form of the flux discretization using this method is

$$j_{e,k+1/2} = -p_7E_{k+1/2}\left[\frac{n_{e,k}}{1 - \exp\left(\frac{E_{k+1/2}\delta x_k}{T_{e,k+1/2}}\right)} + \frac{n_{e,k+1}}{1 - \exp\left(\frac{-E_{k+1/2}\delta x_k}{T_{e,k+1/2}}\right)}\right]. \tag{20}$$

All the other variables are calculated in the same fashion as before, such as with the evaluation of the upwinded method.

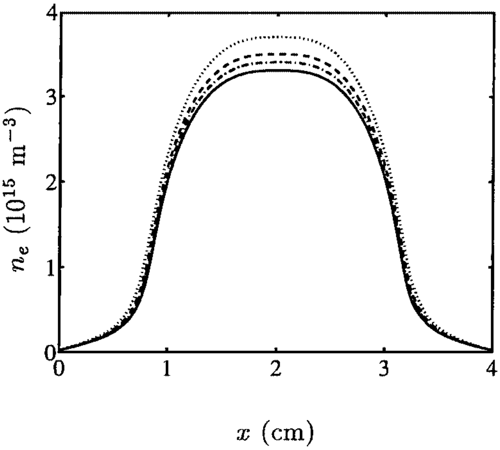

FIG. 5. The period-averaged electron number density at the conditions of Case I in [11] for four different calculations: the “exact” solution (—); 400 points, upwinded (-··-); 200 points, upwinded (- - -); 100 points, upwinded (· · ·). All the simulations used the second-order implicit Runge–Kutta with $\delta t = 0.004$ T, and the grids were stretched using a tanh function.

The accuracy of the Scharfetter–Gummel method is reasonably good, as shown in Fig. 6 with the period-averaged electron number density. However, the grid-converged result differs noticeably from that of the central-difference or upwinded techniques; the plasma density is nearly 20% higher. As with the upwinded calculations, the increase in ion density reflects the increase in electron density, and the ion velocity is hardly affected. The results using the Scharfetter–Gummel discretization differ because in the limit as δx_k tends to zero, the modified equation for $j_{e,k+1/2}$ does not reduce to Eq. (2):

$$j_e\left(x_{k+1/2}\right) = -p_7\left[n_e\left(x_{k+1/2}\right)E\left(x_{k+1/2}\right) + T_e\left(x_{k+1/2}\right)\left.\frac{\partial n_e}{\partial x}\right|_{x_{k+1/2}} + O\left(\delta x_k^2\right)\right]. \qquad (21)$$

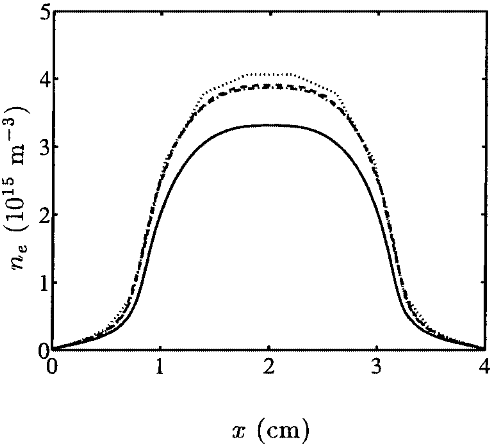

FIG. 6. The period-averaged electron number density at the conditions of Case I in [11] for four different calculations: the “exact” solution (—); 80 points, Scharfetter–Gummel (-··-); 40 points, Scharfetter–Gummel (- - -); 20 points, Scharfetter–Gummel (· · ·). All the simulations used the second-order implicit Runge–Kutta with $\delta t = 0.004$ T, and the grids were stretched using a tanh function.

The electron temperature has been taken out of the pressure gradient, and an additional approximation has been introduced into the fluid model. The Scharfetter–Gummel discretization is inconsistent with the originally posed problem.

4.1.4. *Mean Velocity Flux*

It is evident from the linear variation of n_e in the sheath regions of Fig. 4 that the electron density varies exponentially in the sheath region. Thus, the derivative of the logarithm of n_e would be more accurately calculated numerically than the derivative of an exponentially varying quantity such as n_e. An alternative approach that exploits this behavior and is nondissipative, yet robust, is derived below. The robustness of the method is proven for a simple model problem in Appendix B.

The drift–diffusion approximation can be rearranged and then integrated to determine the mean electron velocity. Since the electron flux, j_e, equals $n_e v_e$, where v_e is the electron velocity, the drift–diffusion Eq. (2) can be divided by n_e and rearranged via the chain and product rules to yield

$$v_e = -p_7\left(E + T_e\frac{\partial \ln n_e}{\partial x} + \frac{\partial T_e}{\partial x}\right). \tag{22}$$

Here, quantities that vary more linearly are being differentiated. Since n_e varies exponentially in x, $\ln n_e$ varies "linearly" in x, and T_e is observed to have only a weak dependence on x.

With the electric field expressed as the gradient of the potential, Eq. (22) can be integrated from x_k to x_{k+1}. Using the midpoint rule for the $T_e\partial \ln n_e/\partial x$ term we obtain the following expression for the mean electron velocity:

$$\bar{v}_{e,k+1/2} = -p_7\left[-\frac{\Phi_{k+1}-\Phi_k}{\delta x_k} + \frac{T_{e,k+1/2}}{\delta x_k}\ln\frac{n_{e,k+1}}{n_{e,k}} + \frac{T_{e,k+1}-T_{e,k}}{\delta x_k}\right]. \tag{23}$$

The electron flux at the midpoint is now simply the product of the average velocity and the average number density:

$$j_{e,k+1/2} = \left(\frac{n_{e,k}+n_{e,k+1}}{2}\right)\bar{v}_{e,k+1/2}. \tag{24}$$

The solutions obtained with this new method (henceforth referred to as the mean velocity method) on several different stretched grids are compared against the "exact" solution in Fig. 7. The solution using 80 points with the mean velocity flux discretization is virtually indistinguishable from the "exact" solution in Fig. 7. The peak density in the 80-point solution differs from the "exact" solution by less than 1%; the 40- and 20-point solutions differ by 1 to 2 percentage points.

The results from a 20-point upwinded calculation and a 20-point calculation with the new method are compared with the "exact" solution in Fig. 8–10. The electron velocity technique is close to the "exact" answer while the upwinded method is noticeably inaccurate. The discrepancy between the two methods is apparent not only in the bulk plasma parameters, but also in the sheath region where the electron densities can be extremely low. Figures 9 and 10 show the large errors with the upwinded calculation for the electron energy ($\frac{3}{2}k_B T_e$) and for the electron density in the sheath regions. The electron energy from the mean

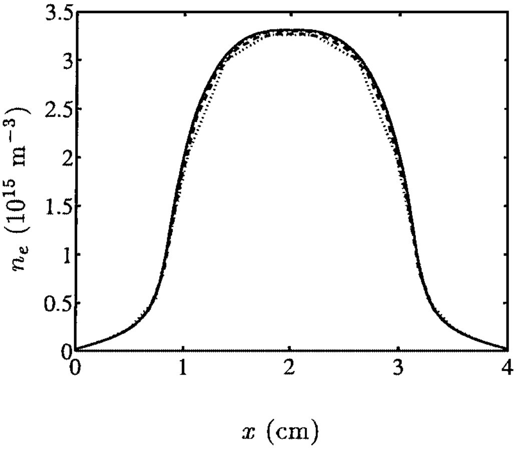

FIG. 7. The period-averaged electron number density at the conditions of Case I in reference [11] is shown for four different simulations: the "exact" solution (—); 80 points, mean velocity (- - -); 40 points, mean velocity (- - -); 20 points, mean velocity (· · ·). All the simulations used the same time advancement and tanh-stretching for the grids.

velocity method differs slightly from that for "exact" solution in the sheaths; this difference disappears as more points are added to the sheath regions. The variables actually being solved for are n_e and n_eT_e, both of which are very nearly zero in the sheaths; the electron energy is found from the ratio of the two.

Besides being more accurate than the common low-order upwinding, the mean velocity method provides two advantages over the Scharfetter–Gummel discretization. Not only does it retain the electron temperature in the pressure gradient, but also it is computationally less burdensome. The run times and floating point operation counts are compared in Table I for three similar runs that used the upwinded, Scharfetter–Gummel and mean velocity methods. The Scharfetter–Gummel technique is clearly much more expensive than either upwinding or the mean velocity method. This is because the electric field appears nonlinearly in the

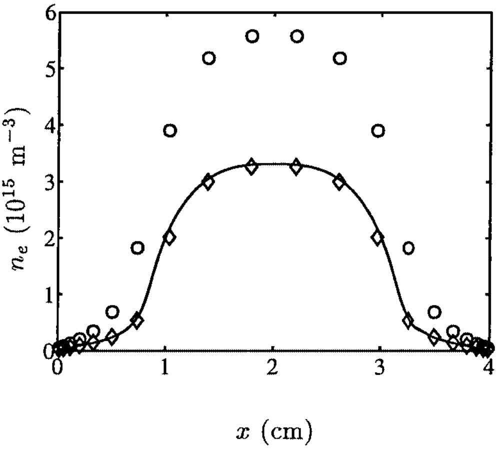

FIG. 8. The period-averaged electron number density at the conditions of Case I in [11] is shown for three different simulations: "exact" solution (—); 20 points, mean velocity (◇); 20 points, upwinded (○). All the simulations used the same time advancement and tanh-stretching for the grids.

TABLE I
Comparison of Computational Cost

Method	Time (sec)	Floating point operations
Scharfetter–Gummel	630	2.70×10^{10}
Upwinding	353	1.30×10^{10}
Mean velocity	338	1.19×10^{10}

Note. All three simulations were run on the same computer (Silicon Graphics Iris Indigo 2) for 500 radio-frequency periods (T) on identical 20-point, tanh-stretched grids with second-order implicit Runge–Kutta time advancement and $\delta t = 0.004$ T.

electron flux discretization, which makes the calculation of the Jacobian for implicit time advancement more expensive. Both the upwinded and the mean velocity methods maintain the linear nature of the electric field in the flux discretization.

4.1.5. *Time Advancement*

Other time advancement techniques were explored besides the fully implicit second-order Runge–Kutta. The implicit Euler method's cost is approximately half the cost per time step of the implicit Runge–Kutta scheme since only one iterative solution is required per time

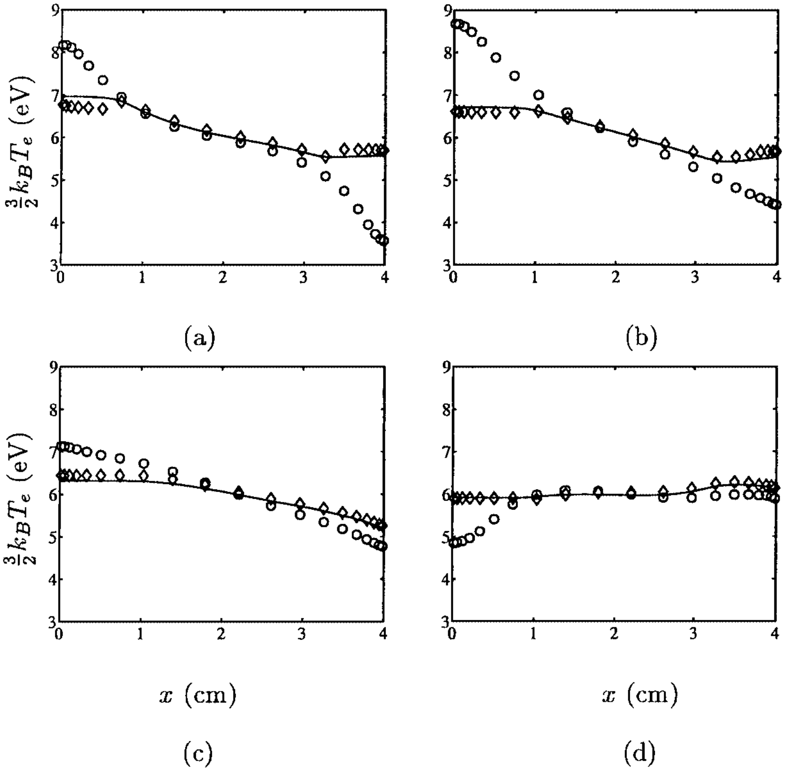

FIG. 9. The instantaneous electron energy profiles at four times during the period for simulations run at the conditions of Case I in [11]: (a), t = 0.5 T; (b), t = 0.625 T; (c), t = 0.75 T; and (d), t = 0.875 T. The three different calculations are the "exact" solution (—); 20 points, mean velocity (◇); 20 points, upwinded (○). All the simulations used identical time advancement and tanh-stretching for the grids.

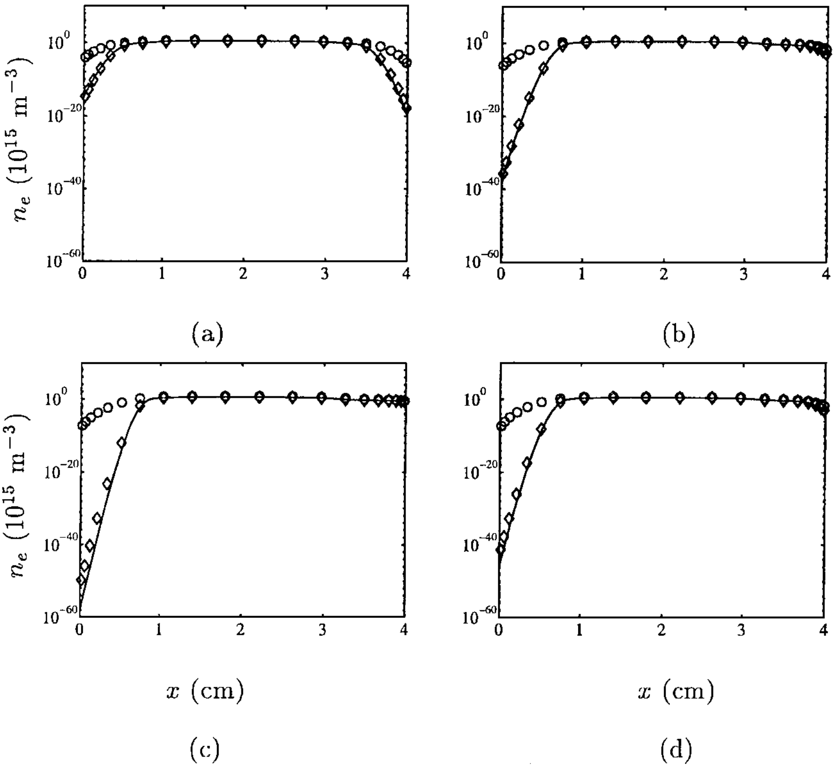

FIG. 10. The instantaneous electron density profiles at four times during the period for the conditions of Case I in [11]: (a), t = 0.5 T; (b), t = 0.625 T; (c), t = 0.75 T; and (d), t = 0.875 T. The three different calculations are the "exact" solution (—); 20 points, mean velocity (◇); 20 points, upwinded (○). All the simulations used identical time advancement and tanh-stretching for the grids.

step (instead of two). Also, with suitable relaxation of the Newton–Raphson iterations, as large a time step as desired can be taken with the implicit Euler, provided the ion CFL's do not become limiting. However, the implicit Euler is less accurate than the second-order Runge–Kutta and requires smaller time steps for the same level of accuracy, which causes longer runtimes. Results from both the fully implicit methods using the same 20-point stretched grid and the mean velocity method are shown in Table II. While the fully implicit Runge–Kutta is time step independent at 250 steps per radio-frequency period, the implicit Euler still shows time step dependence at 1,000 steps per period.

Linearized implicit versions of the fully implicit methods were also examined. Although the runtimes were dramatically shortened (by factors of 4 or more, depending on how many Newton–Raphson iterations were needed for the fully implicit solution), the accuracy suffered. The smaller time steps required for greater accuracy result in runtimes longer than the fully implicit calculations. Results from both the linearized methods using the same 20-point grid discretized with the mean velocity method are shown in Table II. The linearized implicit Euler is quite inaccurate, and even at 4,000 steps per period—a time step 16 times smaller than the time step for the converged fully implicit Runge–Kutta—it still shows time step dependence. Similarly, the linearized implicit Runge–Kutta shows some error at 4,000 steps per period. For suitably small time steps, all the time advancement methods converge to the same result. These results are discussed in a more general context in Appendix A.

TABLE II
Comparison of the Different Time Advancement Methods

Time advancement	Steps per period	Peak density (10^{15} m^{-3})
Fully implicit	1,000	3.26
Runge–Kutta	500	3.26
	250	3.26
Fully implicit	1,000	3.33
Euler	500	3.41
	250	3.55
Linearized implicit	4,000	3.36
Runge–Kutta	2,000	3.45
	1,000	3.67
Linearized implicit	4,000	3.62
Euler	2,000	4.04
	1,000	5.21

Note. All simulations were performed at the conditions of Case I in Nitschke and Graves [11], and they used the mean velocity method on a 20-point, stretched grid.

4.1.6. *Ion Boundary Conditions*

A characteristic analysis of the ion equations indicates that no boundary conditions should be enforced, and the results of simulations with and without boundary conditions bear this out. The effect of imposing boundary conditions is shown in Fig. 11. The $\partial n_i/\partial x = 0$ boundary condition used by Nitschke and Graves [11] is enforced by setting the ion density at the wall equal to the ion density at the nearest adjacent grid point in the interior. Thus, the flux of ions out the left side of the domain (refer to Fig. 2), for example, is determined by $n_{i,1}v_{i,1/2}$. The $\partial v_i/\partial x = 0$ boundary condition on the left side is enforced by setting $v_{i,1/2} = v_{i,3/2}$. With the boundary conditions imposed, oscillations in n_i are observed close to the boundaries. When the boundary conditions are removed (see Section 3.2), so are the oscillations.

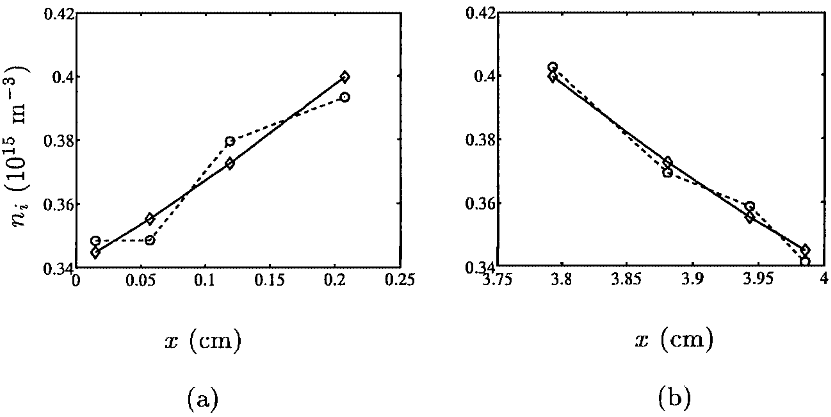

FIG. 11. The period-averaged ion number density at the left edge of the domain (a) and at the right edge (b) for calculations done on 20-point, stretched grids at the conditions of Case I in [11]. The simulations were performed without ion boundary conditions (◇—◇) and with boundary conditions (○- - -○).

Nitschke and Graves upwind their discretizations of the ion continuity and momentum equations, and they do not note any point-to-point oscillations in the ion variables [11].

4.2. Validation

The mean velocity discretization of the electron flux is validated by comparing the results against the previously published results of Nitschke and Graves [11]. Nitschke and Graves reported using 100 to 400 grid points to obtain their solutions. Thus, as shown in Fig. 5, the overall impact of the upwinding is likely not large. Validation is not performed against experimental data at this time as this model was developed to compare fluid and particle simulations. Comparisons with experimental data should be made with a model that represents more of the plasma chemistry, for example, metastable species and their reactions.

The fluid simulations presented in reference [11] are repeated here for the three conditions simulated: 50 mTorr with a 12-cm gap, 100 mTorr with a 6-cm gap, and 250 mTorr with a 4-cm gap. All the cases were run with an applied voltage of 500 V at a frequency of 12 MHz. The overall plasma properties predicted by the mean velocity method are compared in Table III against the "exact" 400-point simulations. The most noticeable difference between the results here and those of Nitschke and Graves is in the electron energy at 100 mTorr; Nitschke and Graves reported a value of 5.9 eV, while a value of 7.0 eV is obtained with the mean velocity method (and also with the standard flux discretization on a 400-point grid). This discrepancy is unusual since all the other parameters in the table, including those at different pressures, agree to within several percentage points. Profiles of various parameters from a 20-point mean velocity simulation at the conditions of Nitschke

TABLE III
Comparison of Overall Plasma Properties

	400 points, standard	40 points, mean velocity	20 points, mean velocity
50 mTorr			
Power (W/m^2)	207	207	206
Current (A/m^2)	14.9	15.0	15.4
Plasma density (10^{15} m^{-3})	1.33	1.33	1.33
Electron energy (eV)	6.52	6.54	6.56
100 mTorr			
Power (W/m^2)	287	285	290
Current (A/m^2)	19.0	19.0	19.5
Plasma density (10^{15} m^{-3})	1.47	1.45	1.44
Electron energy (eV)	6.97	7.00	7.04
250 mTorr			
Power (W/m^2)	522	521	532
Current (A/m^2)	28.2	28.2	29.3
Plasma density (10^{15} m^{-3})	3.31	3.27	3.26
Electron energy (eV)	6.05	6.07	6.11

Note. Plasma properties were computed with the mean velocity method using 20 and 40 grid points, and they are compared to the converged, 400-point "exact" solutions for the different cases of Nitschke and Graves [11]. For the calculation of the power, a small time step (1,000 steps per period) was used since the time integration of the product of the voltage and current is sensitive to the time step size; the error in the power calculation is less than a few percent.

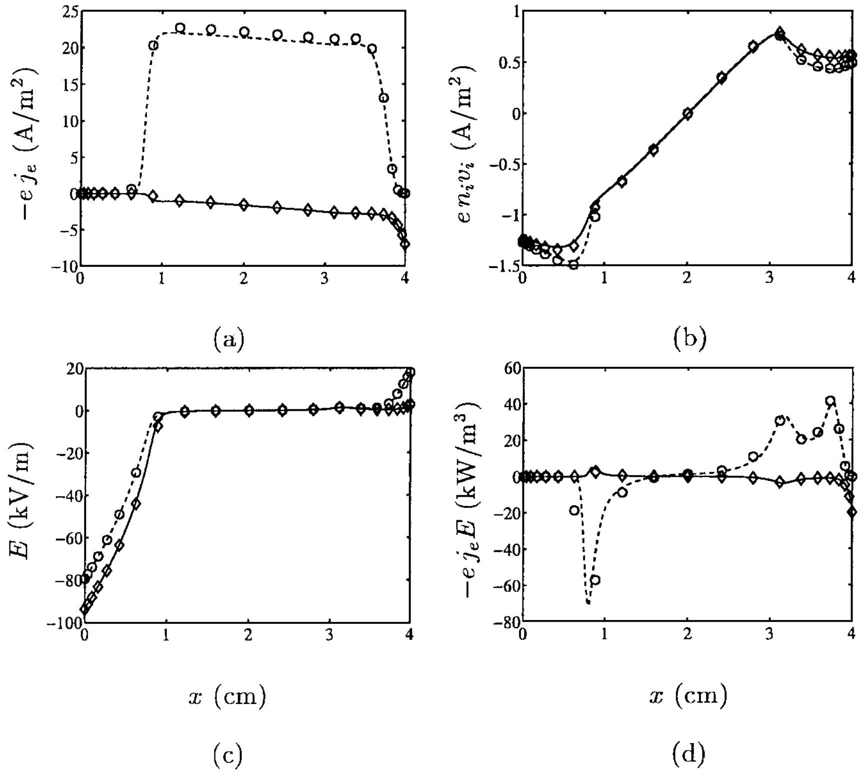

FIG. 12. Instantaneous plasma parameters for Case I in reference [11]: (a) electron current, (b) ion current, (c) electric field, and (d) joule heating ($j_e E$). The line styles and symbols represent two different times in the radio-frequency period for the "exact" solution, 0.75 T (—) and 0.875 T (- - -), and for the 20-point mean velocity solution, 0.75 T (◇) and 0.875 T (○). The coarse 20-point simulation agrees quite well with the resolved 400-point simulation.

and Graves' Case I are shown in Fig. 12 and compared against the "exact" 400-point solution. The parameters plotted are not the dependent variables that are advanced in time but are "secondary" variables that are generally nonlinear combinations of the principal dependent variables. The agreement is quite good. Other comparisons between the mean velocity method and the "exact" solution were shown in Figs. 8 to 10.

4.3. Extension to Multiple Dimensions

The one-dimensional method described above can be implemented in multiple dimensions. The results from a common reactor geometry in a cylindrical coordinate system are shown to demonstrate the method.

4.3.1. *Multidimensional Spatial Discretization*

A structured, staggered grid is used, and scalar parameters, such as n_e or T_e, are located in the cell interior. Vector quantities, such as $\vec{j}_e$ or $\vec{v}_i$, are stored on the cell faces, and the component normal to the face is stored at the face. The staggered grid scheme and the variable locations are illustrated in Fig. 13, and the grid parameters are shown in Fig. 14. The grid is stretched in the z- and r-directions, but it is uniform in the θ-direction.

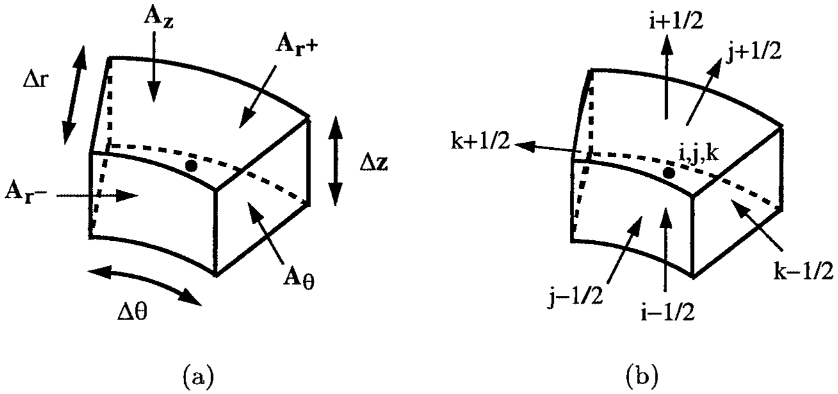

FIG. 13. A generic computational cell is illustrated in (a); a ● denotes the "central" node. The cell volume and the area of the faces are used in the finite volume discretization. In (b), the principle locations of the variables are sketched. Scalar quantities such as n_e, n_i, n_eT_e, T_e, and Φ are stored at the i, j, k node roughly in the cell center. The vector quantities are stored at the cell surfaces and are located halfway between the nodes; the vector component at each face is the component normal to the cell surface. The vector components stored at the cell faces are those for $\vec{j}_e, \vec{q}_e, \vec{v}_i$, and $\vec{E}$.

The finite volume method is used to discretize the electron and ion continuity equations and also the electron energy equation. The ion momentum equation is kept in its primitive form, and it is discretized with finite differences. For Poisson's equation for the electrostatic potential, a discrete Fourier transform is performed in the periodic azimuthal direction for three-dimensional simulations. The semidiscretized equations are presented below in two dimensions for conciseness:

Ion continuity:

$$V_{\text{cell}}\frac{dn_{i_{i,j}}}{dt} = -A_z\left(n_{i_{i+1/2,j}}v_{z_{i+1/2,j}} - n_{i_{i-1/2,j}}v_{z_{i-1/2,j}}\right) - A_{r+}n_{i_{i,j+1/2}}v_{r_{i,j+1/2}} + A_{r-}n_{i_{i,j-1/2}}v_{r_{i,j-1/2}} + V_{\text{cell}}p_1\, n_{e_{i,j}}e^{-p_2/T_{e_{i,j}}}. \tag{25}$$

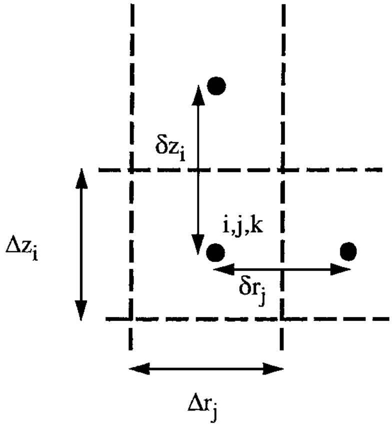

FIG. 14. The locations of the different grid parameters used in the finite difference discretizations and in the flux calculations.

Where nodal values are required at the flux locations, simple averaging is done since the flux locations are midway between the nodes, for example, $n_{i_{i+1/2,j}} = (n_{i_{i,j}} + n_{i_{i+1,j}})/2$.

Ion momentum:

$$\begin{aligned}\frac{dv_{z_{i+1/2,j}}}{dt} = &-\left(\frac{v_{r_{i+1/2,j}}}{\delta r_j + \delta r_{j-1}}\right)\left[\frac{\delta r_{j-1}}{\delta r_j} v_{z_{i+1/2,j+1}} - \left(\frac{\delta r_{j-1}}{\delta r_j} - \frac{\delta r_j}{\delta r_{j-1}}\right) v_{z_{i+1/2,j}} - \frac{\delta r_j}{\delta r_{j-1}} v_{z_{i+1/2,j-1}}\right] \\ &-\left(\frac{v_{z_{i+1/2,j}}}{\Delta z_{i-1} + \Delta z_i}\right)\left[\frac{\Delta z_{i-1}}{\Delta z_i} v_{z_{i+3/2,j}} - \left(\frac{\Delta z_{i-1}}{\Delta z_i} - \frac{\Delta z_i}{\Delta z_{i-1}}\right) v_{z_{i+1/2,j}}\right. \\ &\left. - \frac{\Delta z_i}{\Delta z_{i-1}} v_{z_{i-1/2,j}}\right] + p_3 E_{z_{i+1/2,j}} - p_4 \sqrt{v^2_{z_{i+1/2,j}} + v^2_{r_{i+1/2,j}}}\, v_{z_{i+1/2,j}} \end{aligned} \tag{26}$$

$$\begin{aligned}\frac{dv_{r_{i,j+1/2}}}{dt} = &-\left(\frac{v_{r_{i,j+1/2}}}{\Delta r_j + \Delta r_{j-1}}\right)\left[\frac{\Delta r_{j-1}}{\Delta r_j} v_{r_{i,j+3/2}} - \left(\frac{\Delta r_{j-1}}{\Delta r_j} - \frac{\Delta r_j}{\Delta r_{j-1}}\right) v_{r_{i,j+1/2}}\right. \\ &\left. - \frac{\Delta r_j}{\Delta r_{j-1}} v_{r_{i,j-1/2}}\right] - \left(\frac{v_{z_{i,j+1/2}}}{\delta z_{i-1} + \delta z_i}\right)\left[\frac{\delta z_{i-1}}{\delta z_i} v_{r_{i+1,j+1/2}} - \left(\frac{\delta z_{i-1}}{\delta z_i} - \frac{\delta z_i}{\delta z_{i-1}}\right) v_{r_{i,j+1/2}}\right. \\ &\left. - \frac{\delta z_i}{\delta z_{i-1}} v_{r_{i-1,j+1/2}}\right] + p_3 E_{r_{i,j+1/2}} - p_4 \sqrt{v^2_{z_{i,j+1/2}} + v^2_{r_{i,j+1/2}}}\, v_{r_{i,j+1/2}}. \end{aligned} \tag{27}$$

Some of the velocity components are required at locations where they are not normally stored (e.g., $v_{r_{i+1/2,j}}$); these values are computed by linear interpolation and averaging. This example is illustrated in Fig. 15.

Electron continuity:

$$V_{\text{cell}} \frac{dn_{e_{i,j}}}{dt} = -A_z\left(j_{z_{i+1/2,j}} - j_{z_{i-1/2,j}}\right) - A_{r^+} j_{r_{i,j+1/2}} + A_{r^-} j_{r_{i,j-1/2}} + V_{\text{cell}} p_1\, n_{e_{i,j}} e^{-p_2/T_{e_{i,j}}}. \tag{28}$$

The electron fluxes are found as follows:

$$\begin{aligned} j_{z_{i+1/2,j}} &= -p_7\, n_{e_{i+1/2,j}}\left[E_{z_{i+1/2,j}} + \frac{T_{e_{i+1/2,j}}}{\delta z_i} \ln \frac{n_{e_{i+1,j}}}{n_{e_{i,j}}} + \left(\frac{T_{e_{i+1,j}} - T_{e_{i,j}}}{\delta z_i}\right)\right] \\ j_{r_{i,j+1/2}} &= -p_7\, n_{e_{i,j+1/2}}\left[E_{r_{i,j+1/2}} + \frac{T_{e_{i,j+1/2}}}{\delta r_j} \ln \frac{n_{e_{i,j+1}}}{n_{e_{i,j}}} + \left(\frac{T_{e_{i,j+1}} - T_{e_{i,j}}}{\delta r_j}\right)\right]. \end{aligned}$$

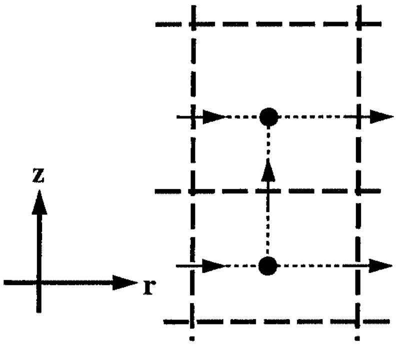

FIG. 15. To compute the value of the radial velocity (v_r) at the axial velocity (v_z) location, first the radial velocities are interpolated onto the cell nodes (indicated by ●'s). These interpolated values at the two nodes, which are equidistant from the axial velocity location, are then averaged to obtain the radial velocity at the axial velocity location. One-sided extrapolation is used adjacent to boundaries.

Electron energy:

$$
\begin{aligned}
V_{\text{cell}}\frac{3}{2}\frac{dn_eT_{e_{i,j}}}{dt} &= -A_z\left(q_{z_{i+1/2,j}} - q_{z_{i-1/2,j}}\right) - A_{r^+}q_{r_{i,j+1/2}} + A_{r^-}q_{r_{i,j-1/2}} \\
&\quad - \frac{1}{2}V_{\text{cell}}\left(j_{z_{i-1/2,j}}E_{z_{i-1/2,j}} + j_{z_{i+1/2,j}}E_{z_{i+1/2,j}} + j_{r_{i,j-1/2}}E_{r_{i,j-1/2}} + j_{r_{i,j+1/2}}E_{r_{i,j+1/2}}\right) \\
&\quad - V_{\text{cell}}p_5\, n_{e_{i,j}}e^{-p_2/T_{e_{i,j}}} - V_{\text{cell}}p_6\, n_{e_{i,j}}\left(T_{e_{i,j}} - T_{\text{neut}}\right). \qquad (29)
\end{aligned}
$$

The electron energy fluxes are defined as

$$
\begin{aligned}
q_{z_{i+1/2,j}} &= \frac{5}{2}j_{z_{i+1/2,j}}T_{e_{i+1/2,j}} - p_8\, n_eT_{e_{i+1/2,j}}\left(\frac{T_{e_{i+1,j}} - T_{e_{i,j}}}{\delta z_i}\right) \\
q_{r_{i,j+1/2}} &= \frac{5}{2}j_{r_{i,j+1/2}}T_{e_{i,j+1/2}} - p_8\, n_eT_{e_{i,j+1/2}}\left(\frac{T_{e_{i,j+1}} - T_{e_{i,j}}}{\delta r_j}\right).
\end{aligned}
$$

Poisson's equation:

$$
\begin{aligned}
&A_z\left[\left(\frac{\Phi_{i+1,j} - \Phi_{i,j}}{\delta z_i}\right) - \left(\frac{\Phi_{i,j} - \Phi_{i-1,j}}{\delta z_{i-1}}\right)\right] + A_{r^+}\left(\frac{\Phi_{i,j+1} - \Phi_{i,j}}{\delta r_j}\right) \\
&\quad - A_{r^-}\left(\frac{\Phi_{i,j} - \Phi_{i,j-1}}{\delta r_j}\right) = -V_{\text{cell}}\left(n_{i_{i,j}} - n_{e_{i,j}}\right). \qquad (30)
\end{aligned}
$$

The matrix resulting from the discretization of Poisson's equation is solved with a conjugate gradient method that is preconditioned with a relaxed, modified incomplete Cholesky decomposition [30].

The time advancement is done with the fully coupled, second-order, semi-implicit, Runge–Kutta scheme of Zhong [19]. As in one dimension, the Jacobian matrix for the multidimensional case is, strictly speaking, nearly a full matrix due to the global effect of the electric field. Frequent inversion of such a matrix would make a multidimensional simulation impossible. Similar to the one-dimensional simulation, the Jacobian calculation ignores the effect of a change in local electric field due to a change in charge density in a nonadjacent (distant) cell. The neglect of nonadjacent terms is even more justified in multiple dimensions since now the change in electric field due to a change in charge density diminishes with distance, which was not the case in one dimension. The Jacobian matrix is inverted by alternating sweeps through the z- and r-directions.

4.3.2. *Two-Dimensional Simulation*

The geometry corresponds to that for the Gaseous Electronics Conference (GEC) Reference Cell [31] and is illustrated in Fig. 16. A 1-mm-thick ground shield is separated from the upper and lower 5-cm-radius electrodes by a 1.5-mm-thick alumina insulator with a dielectric constant of 9. Experimental measurements are often made in the GEC Reference Cell configuration; see [31]. In practice, plasma discharges frequently have electrodes with unequal areas, and the potential on the powered electrode floats. This can lead to a DC self-bias on the powered electrode to enforce a zero time-averaged current through the electrode, i.e., charge is not allowed to build up continuously on the powered electrode. To represent this effect in the simulation, a blocking capacitor is inserted between the voltage-driven power supply and the powered electrode. Although the GEC Cell commonly uses a

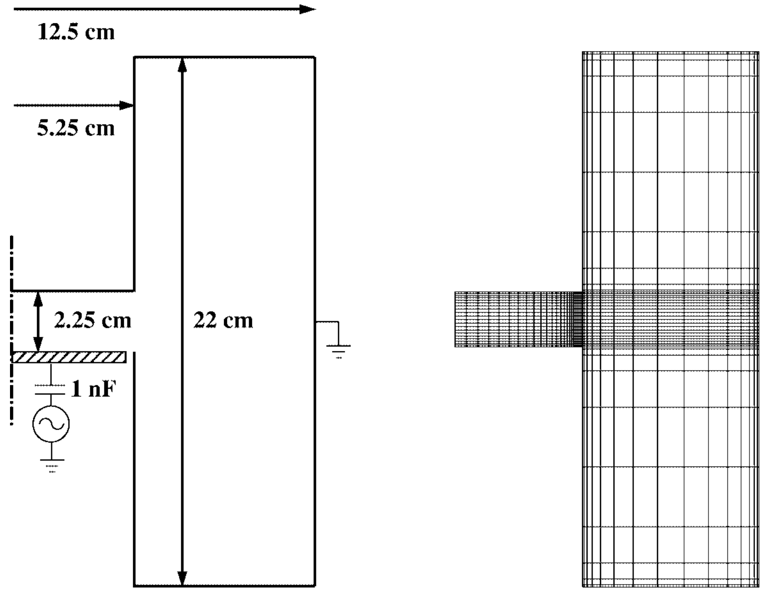

FIG. 16. A sketch of the simulation geometry. The powered electrode is indicated by the hatched box, and the blocking capacitor is shown between the powered electrode and the power supply. The other walls of the chamber are assumed to be grounded. The reactor has azimuthal symmetry; the centerline is indicated by the dash–dot line. The computational cells used to discretize this geometry are shown to the right; the axial and radial directions were each subdivided into 40 computational cells.

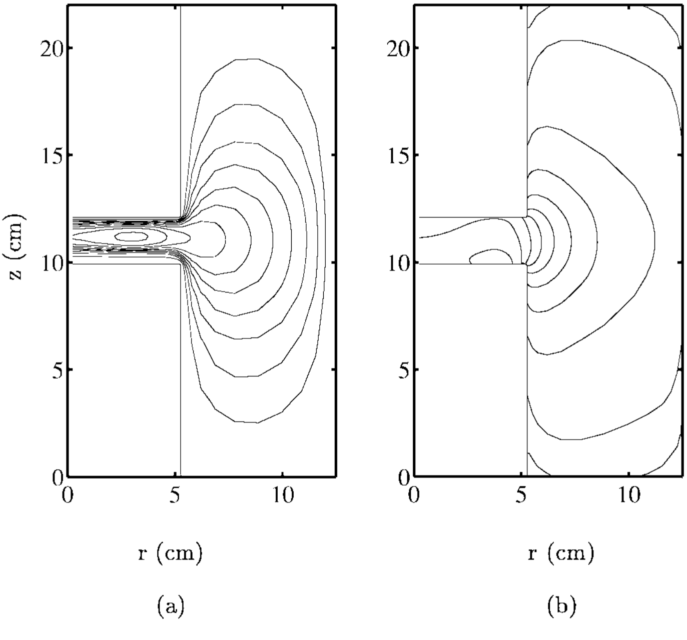

FIG. 17. The period-averaged electron number density, (a), is shown in units of 10^{15} m^{-3}; the contours begin at 0.13×10^{15} m^{-3} and are equally spaced by 0.13×10^{15} m^{-3}. The peak electron density is 1.33×10^{15} m^{-3}. The period-averaged electron energy, (b), is shown in eV; the contours begin at 3.33 eV and are spaced by 0.33 eV. The peak electron energy is 0.39 eV.

100-nF blocking capacitor [31], a 1-nF capacitor was used in the simulation, as indicated in Fig. 16, to reduce the computational time to achieve harmonic steady state. The differential equation governing the voltage drop across the capacitor, $dV_c(t)/dt = I(t)/C$, is advanced explicitly in time using the total current that flows from the powered electrode through the plasma and the alumina insulator.

A simulation was run in this GEC geometry on a 40×40 grid to demonstrate the viability of the mean velocity method for simulations in multiple dimensions. The pressure of the helium background gas was 500 mTorr at a temperature of 300 K; the radio-frequency voltage on the lower electrode was at 13.56 MHz with an amplitude of 110 V (at the power supply the amplitude was 297 V). The powered electrode developed a DC self-bias of −80 V. The period-averaged electron number density and electron energy are shown in Fig. 17. Using 100 steps per period, approximately 1,000 radio-frequency periods were needed to achieve steady state; that is, the relative change in the peak ion number density from period to period was less than 10^{-5} after 1,000 periods. The total CPU time, on the same Silicon Graphics workstation used for the one-dimensional simulations, was approximately 67 hours.

5. CONCLUSION

A fully conservative, efficient numerical algorithm for plasma simulation was derived and extended to multiple dimensions. The method was validated by reproducing the published simulation results for a one-dimensional, radio-frequency, helium discharge of Nitschke and Graves [11]. The method is robust and accurate even on fairly coarse grids. Low-order upwinding of the electron flux discretization on the same coarse grid was found to have a significant impact on the accuracy of the solution. Also, the Scharfetter–Gummel discretization was found to be more expensive computationally than the proposed method and found to introduce an additional approximation that significantly affected the plasma density.

APPENDIX A

Fully Implicit Versus Linearized Implicit Time Advancement

Stiff systems of differential equations are usually solved with implicit methods. The implicit discretization can be linearized to increase the efficiency of the numerical algorithm; however, it is possible that the linearization can significantly impact the accuracy of the solution method. An example of such a situation is the following ODE:

$$\frac{dy}{dt} = f(t, y) = e^{y-t} \quad \text{with} \quad y(t) = -\ln(e^{-y_0} + e^{-t} - 1), \quad y_0 = y(t = 0).$$

The analytical solution can be used to compute exact values for the leading error terms in the implicit Euler method: the time discretization error ($\frac{d^2y}{dt^2}\frac{\delta t}{2}$), which is the only source of error for the fully implicit treatment, and the linearization error ($-\frac{\partial^2 f}{\partial y^2}\frac{(\partial y)^2}{2}$), which is a result of linearizing in the unknown coordinate, y. Typically, the linearization error term is thought of as being $O(\delta t^2)$, since δy can be approximated by $f\delta t$. Thus, one might assume that the error introduced by the linearization is less than the error due to the time discretization. However, this is not always the case. These error terms from the fully nonlinear solution and the linearized solution are shown as a function of time along with the exact solution in Fig. 18 for $y_0 = -1 \times 10^{-5}$ and $\delta t = 0.1$. The linearization error is significantly larger than the time discretization error at the start of the calculation, and this causes the linearized implicit calculation to be much less accurate than the fully implicit calculation.

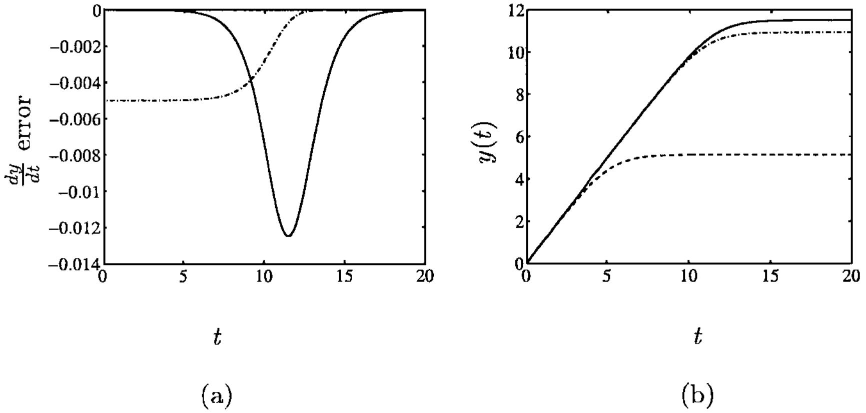

FIG. 18. In (a), the time discretization (—) and the linearization (-··-) error terms are plotted as a function of time. In (b), the exact solution (—), the fully implicit Euler solution (-··-), and the linearized implicit Euler solution (- - -) are plotted as functions of time.

APPENDIX B

Robustness of the Mean Velocity Method for Simple Drift–Diffusion

The steady-state, drift–diffusion problem $d/dx(bu - a\,du/dx) = 0$, with Dirichlet boundary conditions $u(x = 0) = u_0$ and $u(x = L) = u_L$ and $a > 0$, makes a good model problem to estimate the behavior of the electrons, especially in the sheath region where both n_e and $\partial n_e/\partial t$ are going to zero. This model can be used to compare the classic central difference discretization with the mean velocity method. The computational burden can be reduced by considering a crude three-point discretization that has two (known) boundary points, u_0 and u_L, with one central point, u, to be solved for with grid spacing $h = L/2$, illustrated in Fig. 19. The exact solution is

$$u(x = L/2) = \frac{u_0\left(e^{bL/a} - e^{bL/2a}\right) + u_L\left(e^{bL/2a} - 1\right)}{e^{bL/a} - 1},$$

which yields a "physical" positive number when "physical" boundary conditions are specified, $u_0, u_L > 0$.

On a staggered grid with the u "flux" (given by $bu - a\,du/dx$) calculated halfway between the grid points, the classic central difference discretization provides the following

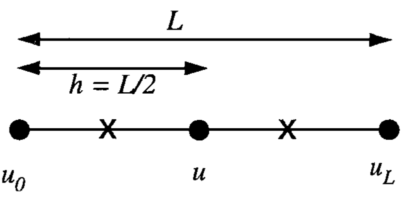

FIG. 19. Discretization for the model drift–diffusion problem; u is the only unknown, and the "flux" of u is calculated at the ×'s, which are halfway between the ●'s.

equation for u:

$$\frac{1}{h}\left\{\left[b\left(\frac{u_L+u}{2}\right)-a\left(\frac{u_L-u}{h}\right)\right]-\left[b\left(\frac{u+u_0}{2}\right)-a\left(\frac{u-u_0}{h}\right)\right]\right\}=0.$$

This can easily be solved to find u in terms of the boundary conditions and the grid Peclet number, $\mathrm{Pe}=bh/2a$:

$$u=\frac{1}{2}(1-\mathrm{Pe})u_L+\frac{1}{2}(1+\mathrm{Pe})u_0.$$

This discretization can fail by predicting a "nonphysical" negative value for u when the magnitude of the grid Peclet number is greater than 1 and appropriate boundary conditions are chosen (e.g., $u_L \gg u_0$ or $u_0 \gg u_L$).

The mean velocity discretization on the same staggered grid provides the following equation for u:

$$\frac{1}{h}\left[\left(\frac{u_L+u}{2}\right)\left(b-\frac{a}{h}\ln\frac{u_L}{u}\right)-\left(\frac{u+u_0}{2}\right)\left(b-\frac{a}{h}\ln\frac{u}{u_0}\right)\right]=0.$$

This equation can be rearranged so that the left-hand side, $f(u)$, depends only on the boundary conditions and the grid Peclet number:

$$f(u)=(u_L+u)\left(\mathrm{Pe}-\frac{1}{2}\ln\frac{u_L}{u}\right)-(u+u_0)\left(\mathrm{Pe}-\frac{1}{2}\ln\frac{u}{u_0}\right)=0.$$

It can be shown that only one root exists to $f(u)=0$, regardless of the grid Peclet number or strictly positive boundary conditions. First, at least one root exists because $f(u)$ is a smooth, continuous function of u for $u \in (0,\infty)$, and it changes sign for $u \in (0,\infty)$. As $u \to 0$, $f(u) \to 1/2(u_L+u_0)\ln u$, which is negative for $u_0, u_L > 0$, and as $u \to \infty$, $f(u) \to u \ln u$, which is positive. Second, only one root exists because df/du is strictly positive. The slope of $f(u)$ is

$$\frac{df}{du}=\ln\frac{u}{\sqrt{u_0u_L}}+\frac{u_0+u_L}{2u}+1.$$

The problem can be simplified with no loss of generality by rescaling the u variables so that $u_L=1$. The condition for $f(u)$ to be a strictly increasing function can be rewritten as

$$u\exp\left(\frac{1+u_0+2u}{2u}\right)>\sqrt{u_0}$$

$$\text{let } y=\sqrt{u_0}$$

$$u\left[1+\left(\frac{1+y^2+2u}{2u}\right)+\frac{1}{2}\left(\frac{1+y^2+2u}{2u}\right)^2+\cdots\right]>y$$

$$u+\frac{1+y^2+2u}{2}+\frac{1}{2}\frac{(1+y^2+2u)^2}{4u}+\cdots>y.$$

It is clear that $(1+y^2)/2 \geq y$ for all y; thus, since $u>0$, the above inequality is true and $f(u)$ is a strictly increasing function for $u, u_0, u_L > 0$.

ACKNOWLEDGMENT

The authors acknowledge helpful discussions with M. A. Capelli. E. P. Hammond was supported by the Brit and Alex d'Arbeloff Stanford Graduate Fellowship.

REFERENCES

1. M. J. Kushner, Advances in plasma equipment modeling, *Solid State Technol*. **39**(6), 135 (1996).
2. J. D. P. Passchier and W. J. Goedheer, A two-dimensional fluid model for an argon rf discharge, *J. Appl. Phys.* **74**, 3744 (1993).
3. P. L. G. Ventzek, R. J. Hoekstra, and M. J. Kushner, Two-dimensional modeling of high plasma density inductively coupled sources for materials processing, *J. Vac. Sci. Technol. B* **12**, 461 (1994).
4. J. P. Boeuf and L. C. Pitchford, Two-dimensional model of a capacitively coupled rf discharge and comparisons with experiments in the Gaseous Electronics Conference reference reactor, *Phys. Rev. E* **51**, 1376 (1995).
5. D. P. Lymberopoulos and D. J. Economou, Two-dimensional self-consistent radio frequency plasma simulations relevant to the gaseous electronics conference RF reference cell, *J. Res. Natl. Inst. Stand. Technol.* **100**, 473 (1995).
6. D. P. Lymberopoulos and D. J. Economou, Two-dimensional simulation of polysilicon etching with chlorine in a high density plasma reactor, *IEEE Trans. Plasma Sci.* **23**, 573 (1995).
7. J. D. Bukowski, D. B. Graves, and P. Vitello, Two-dimensional fluid model of an inductively coupled plasma with comparison to experimental spatial profiles, *J. Appl. Phys.* **80**, 2614 (1996).
8. M. J. Kushner, W. Z. Collison, M. J. Grapperhaus, J. P. Holland, and M. S. Barnes, A. three-dimensional model for inductively coupled plasma etching reactors: Azimuthal symmetry, coil properties, and comparison to experiments, *J. Appl. Phys.* **80**, 1337 (1996).
9. M. J. Kushner, Consequences of asymmetric pumping in low pressure plasma processing reactors: A three-dimensional modeling study, *J. Appl. Phys.* **82**, 5312 (1997).
10. H. H. Hwang, E. R. Keiter, and M. J. Kushner, Consequences of three-dimensional physical and electromagnetic structures on dust particle trapping in high plasma density material processing discharges, *J. Vac. Sci. Technol. A* **16**, 2454 (1998).
11. T. E. Nitschke and D. B. Graves, A comparison of particle in cell and fluid model simulations of low-pressure radio frequency discharges, *J. Appl. Phys.* **76**, 5646 (1994).
12. Y. Lin and R. A. Adomaitis, Simulation and model reduction methods for an RF plasma glow discharge, *J. Comput. Phys.* **171**, 731 (2001).
13. M. S. Barnes, T. J. Colter, and M. E. Elta, Large-signal time-domain modeling of low-pressure rf glow discharges, *J. Appl. Phys.* **61**, 81 (1987).
14. D. L. Scharfetter and H. K. Gummel, Large-signal analysis of a silicon read diode oscillator, *IEEE Trans. Electron Devices* **ED-16**, 64 (1969).
15. M. A. Lieberman and A. J. Lichtenberg, *Principles of Plasma Discharges and Materials Processing* (Wiley, New York, 1994).
16. Y. P. Raizer, *Gas Discharge Physics* (Springer-Verlag, Berlin, 1991).
17. M. Surendra and M. Dalvie, Moment analysis of rf parallel-plate-discharge simulations using the particle-in-cell with Monte Carlo collisions technique, *Phys. Rev. E* **48**, 3914 (1993).
18. F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* **8**, 2182 (1965).
19. X. Zhong, Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows, *J. Comput. Phys.* **128**, 19 (1996).
20. H. H. Rosenbrock, Some general implicit processes for the numerical solution of differential equations, *Comput. J.* **5**, 329 (1963).
21. W. H. Press, S. A. Teukolsky, W. V. Vetterling, and B. P. Flannery, *Numerical Recipes* (Cambridge Univ. Press, Cambridge, 1992).
22. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, New York, 1993).

23. T. J. Sommerer and M. J. Kushner, Numerical investigation of the kinetics and chemistry of rf glow discharge plasmas sustained in He, N_2, O_2, He/N_2/O_2, He/CF_4/O_2, and SiH_4/NH_3 using a Monte Carlo–fluid hybrid model, *J. Appl. Phys.* **71**, 1654 (1992).

24. E. Gogolides, H. H. Sawin, and R. A. Brown, Direct calculation of time-periodic states of continuum models of radio-frequency plasmas, *Chem. Eng. Sci.* **47**, 3839 (1992).

25. P. Colella, M. R. Dorr, and D. D. Wake, A Conservative Finite Difference Method for the Numerical Solution of Plasma Fluid Equations, *J. Comput. Phys.* **149**, 168 (1999).

26. G. L. Huppert, H. H. Sawin, and R. A. Brown, Spectral element analysis of radio-frequency glow discharges, *Chem. Eng. Sci.* **49**, 1601 (1994).

27. J. P. Boeuf, Numerical model of rf glow discharges, *Phys. Rev. A* **36**, 2782 (1987).

28. G. J. Nienhuis, W. J. Goedheer, E. A. G. Hamers, W. G. J. H. M. van Sark, and J. Bezemer, A self-consistent fluid model for radio-frequency discharges in SiH_4–H_2 compared to experiments, *J. Appl. Phys.* **82**, 2060 (1997).

29. G. J. M. Hagelaar and G. M. W. Kroesen, Speeding up fluid models for gas discharges by implicit treatment of the electron energy source term, *J. Comput. Phys.* **159**, 1 (2000).

30. J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Numerical Linear Algebra for High-Performance Computers* (Soc. for Industr. & Appl. Math., Philadelphia, 1998).

31. J. K. Olthoff and K. E. Greenberg, The gaseous electronics conference RF reference cell—An introduction, *J. Res. Natl. Inst. Stand. Technol.* **100**, 327 (1995).