Journal of Computational Physics **152,** 550–583 (1999)
Article ID jcph.1999.6245, available online at http://www.idealibrary.com on IDEAL®

Numerical Solution of Plasma Fluid Equations Using Locally Refined Grids[1]

Phillip Colella, Milo R. Dorr, and Daniel D. Wake

Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, California 94551

Received April 14, 1998; revised February 17, 1999

This paper describes a numerical method for the solution of plasma fluid equations on block-structured, locally refined grids. The plasmas under consideration are typical of those used for the processing of semiconductors. The governing equations consist of a drift–diffusion model of the electrons, together with an energy equation, coupled via Poisson's equation to a system of Euler equations for each ion species augmented with electric field, collisional, and source/sink terms. A discretization previously developed for a uniform spatial grid is generalized to enable local grid refinement. This extension involves the time integration of the discrete system on a hierarchy of levels, each of which represents a degree of refinement, together with synchronization steps to ensure consistency across levels. This approach represents an advancement of methodologies developed for neutral flows using block-structured adaptive mesh refinement (AMR) to include the significant additional effect of the electrostatic forces that couple the ion and electron fluid components. Numerical results that assess the accuracy and efficiency of the method and illustrate the importance of using adequate resolution are also presented.

1. INTRODUCTION

Many of the process steps performed in the manufacture of very large scale integrated (VLSI) circuits involve plasmas [18, 19]. Inductively coupled plasma (ICP) reactors represent one type of processing tool that utilizes high-density, low-pressure plasmas to satisfy the demanding process criteria resulting from the desire to create increasingly smaller device features on large wafers. Computational models of ICP reactors can help equipment manufacturers and process engineers understand the complex relationships among reactor

[1] This work was supported by the Applied Mathematical Sciences Program of the Office of Mathematics, Information and Computational Sciences, of the U.S. Department of Energy under Contract DE-AC03-76SF00098 with Lawrence Berkeley National Laboratory and by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

0021-9991/99

and plasma parameters (e.g., reactor geometry, radiofrequency (RF) power, gas pressure, gas composition, electrode bias, densities, velocities) and process performance (e.g., etch rate, anisotropy, uniformity, selectivity, damage).

A number of simulators have been developed to computationally model process plasmas. The Hybrid Plasma Equipment Model (HPEM) [30] at the University of Illinois consists of many interacting physics modules (e.g., electromagnetics, chemistry, fluids, Monte Carlo collisions) and has been used extensively [12, 13, 17, 23, 24, 28, 29]. The INDUCT code [26, 27] developed at Lawrence Livermore National Laboratory implements a two-dimensional, axisymmetric plasma fluid model with a self-consistent treatment of the inductive fields generated by the RF coils. INDUCT includes neutral flow, multiple ion species, and variable ion temperature. It has been benchmarked against experiment [4, 31] and has been used extensively within the semiconductor industry. Work at the University of California at Berkeley has extended the code to include complex chemistry, and the effect of these reactions has been investigated with INDUCT [5]. The commercial package CFD-PLASMA (ICP), utilizing unstructured grids, has been developed by CFD Corporation for ICP simulations [33].

Among the primary difficulties encountered in the development of computational models of plasma processes is the need to address problems associated with a wide range of temporal and spatial scales. The high-density, low-pressure plasmas employed in ICP reactors are particularly challenging in this regard. Here, the difference between the dielectric relaxation time of the plasma and the reactor transit time for an ion can be many orders of magnitude. Moreover, due to the small Debye length of such plasmas, the sheath and presheath regions over which much of the potential drop occurs can be three or four orders of magnitude smaller than the reactor dimensions. Numerical methods must therefore accommodate large scale variations. For example, all of the plasma simulators listed above use some form of "sheath model" to handle the rapid variation of the underresolved state variables near the reactor boundary.

One way to handle multiple scales in a numerical model is through the use of adaptive mesh refinement (AMR). AMR algorithms permit the underlying computational mesh to be modified in space and time to follow changing solution features. To date, the application of AMR techniques to plasma simulation has been mainly limited to fusion plasmas (e.g., [16] and references therein), and magnetohydrodynamics [25]. Relatively little attention has been given to the use of AMR in the simulation of high-density, low-pressure process plasmas. In [9], we described and analyzed a numerical algorithm for the solution of a system of plasma fluid equations on a uniform spatial grid. In the present companion paper, we begin an investigation of AMR for high-density, low-pressure plasma simulation by generalizing this algorithm to enable the solution of the same system of plasma fluid equations on locally refined grids. We focus our attention on grid structures with multiple levels of resolution that are prescribed *a priori*, which is the first step toward a fully adaptive algorithm.

The approach we have taken in this work utilizes and extends a number of ideas that have been developed for the application of AMR in other contexts. One of the first block-structured AMR algorithms was introduced in [3] for hyperbolic conservation laws using finite difference methods on a hierarchy of regular Cartesian grids. Early applications included problems arising in gas dynamics and shock physics [2, 11]. In this approach, each level of the grid hierarchy corresponds to a degree of spatial refinement, where the location and topology of the grids is determined by Richardson extrapolation estimates of the truncation error combined with cell tagging/clustering algorithms. The hyperbolic system is

integrated on each refinement level separately using a time step appropriate to the CFL stability requirement of the grid on that particular level. Communication among levels occurs through the use of temporally and spatially interpolated coarse grid data to define boundary conditions for the integrations on finer levels, as well as so-called "refluxing" operations performed to restore flux continuity and local conservation at coarse–fine boundaries.

The generalization of AMR for systems that are not purely hyperbolic, such as the plasma fluid model we consider here, presents a number of additional challenges. Elliptic, parabolic, and mixed systems lack the real characteristic structure and transit time isolation that make hyperbolic AMR straightforward in comparison. Although the specific algorithms are problem-dependent, some general strategies have been developed and applied to problems in incompressible flow [1], combustion [22], and radiation transport [14]. A common theme is the continued use of nested levelwise integrations as in the purely hyperbolic case, with the addition of "composite synchronization" steps to restore global constraints (e.g., incompressibility or charge conservation) as well as any compatibility conditions at the interface between coarse and fine grids that may have been violated during the level integrations. These synchronization steps typically require the solution of linear systems on a portion or all of the grid hierarchy. The fact that such linear systems must be solved on a hierarchy of successively refined grids makes multigrid-based multilevel solution algorithms a natural choice. In fact, the availability and high efficiency of multigrid algorithms on Cartesian grids is one of the primary motivations for using a block-structured AMR approach. In addition to the vast literature on multigrid methods, a good overview of multilevel adaptive methods can be found in [21]. The particular approach incorporated in the present work was motivated by the ideas presented in [20].

In Section 2, we present the physical model and the system of equations to be solved. The plasma is regarded as a fluid consisting of charged components, ions and electrons, coupled by Poisson's equation. The motion of the electrons is described by a drift–diffusion model including temperature, while each ion species is modeled by a system of Euler equations augmented with electrostatic force, collisional, and source/sink terms. The discretization of the system described in [9] on a uniform spatial grid is summarized in Section 3. In Section 4, we extend the single grid algorithm to the case of block-structured locally refined grids. Beginning with a rectangular uniform grid, subregions are successively refined to create a hierarchy of refinement levels, each of which is a union of non-overlapping rectangles. The time evolution of the fluid systems is accomplished through the coordinated advancement of individual refinement levels, using the single grid algorithm of Section 3, together with composite synchronization steps to enforce charge conservation and continuity of particle fluxes and electrostatic field across levels. In Section 5, numerical results illustrating the accuracy and efficiency of the method are presented. The method is then applied to two problems of practical interest, both of which suggest the need for adequate resolution in simulations.

2. THE PLASMA MODEL

We consider the plasma fluid model described in [9] consisting of the electron equations

$$\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e u_e) = \sum_j R_{ej}, \tag{2.1}$$

$$n_e u_e = n_e \mu \nabla \phi - \eta \nabla (n_e k T_e), \tag{2.2}$$

$$\frac{\partial\left(\frac{3}{2}n_e kT_e\right)}{\partial t} + \nabla\cdot\left(n_e\frac{5}{2}kT_e u_e\right) = -en_e u_e\cdot E + \nabla\cdot\left(\frac{5}{2}n_e kT_e\eta\nabla kT_e\right) + P_{\text{ind}} - \sum_j \frac{3m_e}{m_j}kT_e\nu_{ej}n_e + \sum_j \epsilon_{ej}S_{ej}, \quad (2.3)$$

the ion equations, for each species i,

$$\frac{\partial n_i}{\partial t} + \nabla\cdot(n_i u_i) = \sum_j R_{ij}, \quad (2.4)$$

$$\frac{\partial(m_i n_i u_i)}{\partial t} + \nabla\cdot(m_i n_i u_i u_i) + \nabla(n_i kT_i) = q_i n_i E - \sum_j \frac{m_i m_j}{m_i+m_j}n_i\nu_{ij}(u_i-u_j) + \sum_{j|R_{ij}<0} m_i u_i R_{ij} + \sum_{j|R_{ij}>0} m_i u_j R_{ij}, \quad (2.5)$$

$$\frac{\partial}{\partial t}\left[n_i\left(\frac{m_i}{2}u_i\cdot u_i + \mathcal{E}_i\right)\right] + \nabla\cdot\left[n_i\left(\frac{m_i}{2}u_i\cdot u_i + \mathcal{E}_i\right)u_i\right] + \nabla\cdot(n_i kT_i u_i) = q_i n_i u_i\cdot E - \sum_j \frac{2m_i m_j}{(m_i+m_j)^2}n_i\nu_{ij}\left[\frac{1}{2}(m_i u_i\cdot u_i - m_j u_j\cdot u_j + (m_j-m_i)u_i\cdot u_j) + \mathcal{E}_i - \mathcal{E}_j\right] + \sum_{j|R_{ij}<0}\left(\frac{m_i}{2}u_i\cdot u_i + \mathcal{E}_i\right)R_{ij} + \sum_{j|R_{ij}>0}\left(\frac{m_j}{2}u_j\cdot u_j + \mathcal{E}_j\right)R_{ij} + \sum_j \epsilon_{ij}S_{ij}, \quad (2.6)$$

$$\mathcal{E}_i = \frac{kT_i}{\gamma_i - 1}, \quad (2.7)$$

and Poisson's equation

$$\epsilon_0\nabla\cdot E = \sum_i q_i n_i - q_e n_e, \qquad E = -\nabla\phi \quad (2.8)$$

with the variables defined in Table 2.1. In deriving the electron equations (2.1)–(2.3) from moments of the Boltzmann equation, we have neglected electron inertia in replacing the electron momentum moment equation with the drift–diffusion force balance equation (2.2), and we have replaced the electron total energy moment equation with the internal energy equation (2.3). The electron transport coefficients are given by

$$\nu_{en} \equiv \sum_j \nu_{ej}, \quad (2.9)$$

$$\mu \equiv e/m_e\nu_{en}, \quad (2.10)$$

$$\eta \equiv 1/m_e\nu_{en}. \quad (2.11)$$

Boundary conditions for the electron continuity equation (2.1) are derived from the approximation

$$n_e u_e \approx \Gamma_b \equiv n_e\bar{w}_e(kT_e,\phi) \quad (2.12)$$

for the total electron boundary flux, where $\bar{w}_e$ is the average velocity of electrons reaching the boundary given by

$$\bar{w}_e(kT_e,\phi) \equiv \frac{\bar{v}(kT_e)}{4}\exp\left(-\frac{e|\phi_b-\phi|}{kT_e}\right) \quad (2.13)$$

TABLE 2.1
Variables Used in the Plasma Fluid Model

n_j	Number density
u_j	Velocity
ϕ	Potential
E	Electric field
T_j	Temperature
$\mathcal{E}_j$	Internal energy
γ_j	Ratio of specific heats
P_{ind}	Input power density
μ, η	Transport coefficients
m_j	Mass
R_{ij}	Number density gain/loss rates from ionization, attachment, etc.
S_{ij}	Number density gain/loss rates from inelastic collisions
ϵ_{ej}	Energy transferred in inelastic collisions
ν_{ij}	Elastic collision frequencies
q_j	Charge
e	Elementary charge
ϵ_0	Permittivity of free space
k	Boltzmann's constant

and $\overline{v}(kT_e) = \sqrt{8kT_e/\pi m_e}$ is the average speed for a Maxwellian distribution. Using (2.12), one obtains boundary conditions for the electron temperature equation (2.3) from

$$Q_b \equiv \frac{5}{2} n_e k T_e u_e - \frac{5}{2} \frac{n_e k T_e}{m_e \nu_{en}} \nabla k T_e = 2 k T_e \Gamma_b, \tag{2.14}$$

where Q_b is the total energy boundary flux.

Hypersonic outflow boundary conditions are assumed for the ion equations (2.4)–(2.6) corresponding to positively charged species i ($q_i > 0$). As explained in [9], this choice is consistent with the usual Bohm sheath criterion in electropositive systems (i.e., all ion species are positively charged) as well as modifications of the Bohm condition for electronegative systems (i.e., at least one ion species is negatively charged). Zero flux boundary conditions are imposed for negative ion species i ($q_i < 0$).

Dirichlet boundary conditions are specified for the potential ϕ in (2.8) corresponding to an applied voltage ϕ_b at the boundary. The value of ϕ_b can vary spatially along the boundary and with each time step. For example, in the simulation of an ICP reactor, ϕ_b along the wafer boundary could be prescribed as a sinusoidally varying waveform representing the applied radiofrequency (RF) bias.

The charged species described by (2.1)–(2.8) are generated from, and collide with, a background neutral gas. Although the neutral flow is important, we restrict our attention in this paper to the charged species only and assume that the dynamics of the neutral species are already known.

3. SINGLE-LEVEL ALGORITHM

Assume that we have known values for all quantities at time t_n, and that we wish to integrate (2.1)–(2.8) to a new time $t_{n+1} = t_n + \Delta t$. The time step Δt is chosen such that

$$\Delta t \le \alpha \min\left(\frac{\Delta x}{\mu^n \left| E_x^n \right|}, \frac{\Delta y}{\mu^n \left| E_y^n \right|} \right), \tag{3.15}$$

which is the stability restriction for the explicit steps in the algorithm. Since (3.15) is based on data at the beginning of the time step, we incorporate the factor α (typically $\alpha = 1/2$) to account for velocity changes during the step. As described in [9], the integration is split into five main steps, which we now describe.

Step 1. Advance the ions using a second-order Godunov method. For each species i, the system of ion equations (2.4)–(2.6) can be written as a single vector equation

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x} F(U) + \frac{\partial}{\partial y} G(U) = H(U, E), \tag{3.16}$$

where we have the definitions

$$U \equiv \begin{bmatrix} n_i \\ \Gamma_x \\ \Gamma_y \\ \frac{1}{2} u_i \cdot u_i + \mathcal{E}_i / m_i \end{bmatrix}, \qquad \Gamma_x \equiv n_i u_{i,x}, \ \Gamma_y \equiv n_i u_{i,y} \tag{3.17}$$

$$c^2 \equiv \frac{\gamma_i k T_i + k T_e}{m_i} \tag{3.18}$$

$$F(U) \equiv \begin{bmatrix} \Gamma_x \\ \frac{\Gamma_x^2}{n_i} + c^2 n_i \\ \frac{\Gamma_x \Gamma_y}{n_i} \\ \left(\frac{1}{2} u_i \cdot u_i + \mathcal{E}_i / m_i + c^2 \right) \Gamma_x \end{bmatrix}, \qquad G(U) \equiv \begin{bmatrix} \Gamma_y \\ \frac{\Gamma_x \Gamma_y}{n_i} \\ \frac{\Gamma_y^2}{n_i} + c^2 n_i \\ \left(\frac{1}{2} u_i \cdot u_i + \mathcal{E}_i / m_i + c^2 \right) \Gamma_y \end{bmatrix} \tag{3.19}$$

$$H(U, E) \equiv \frac{1}{m_i} \begin{bmatrix} S_0 \\ k T_e \frac{\partial n_i}{\partial x} + q_i n_i E_x + S_{1x} \\ k T_e \frac{\partial n_i}{\partial y} + q_i n_i E_y + S_{1y} \\ \nabla \cdot (n_i k T_e u_i) + q_i n_i u_i \cdot E + S_2 \end{bmatrix}, \tag{3.20}$$

where S_0, S_1, and S_2 denote the collisional and gain/loss terms in the right-hand sides of (2.4)–(2.6). The system is slightly modified for the electronegative case as described in [9]. Note that (3.18) defines the *plasma* sound speed rather than the usual neutral gas speed $\sqrt{\gamma_i k T_i / m_i}$. The inclusion of the $k T_e / m_i$ term in (3.18) is balanced by the density derivative terms in the right-hand side (3.20) (i.e., the first term in the second, third, and fourth components of (3.20)). The asymptotic analysis presented in [9] shows that this

particular organization of the ion equations, when combined with the electron and Poisson equations, results in a slowly varying source term (3.20) that can be lagged in an explicit time integration.

We use a second-order, unsplit Godunov method [8] to solve (3.16). The update scheme in a typical cell (i, j) is

$$U_{i,j}^{n+1} = U_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left[F\left(\tilde{U}^{n}_{i+\frac{1}{2},j}\right) - F\left(\tilde{U}^{n}_{i-\frac{1}{2},j}\right)\right] - \frac{\Delta t}{\Delta y}\left[G\left(\tilde{U}^{n}_{i,j+\frac{1}{2}}\right) - G\left(\tilde{U}^{n}_{i,j-\frac{1}{2}}\right)\right] + \frac{\Delta t}{2}\left[H\left(U^{n}_{i,j}, E^{n}\right) + H\left(U^{*n+1}_{i,j}, E^{n+1}\right)\right], \tag{3.21}$$

where $\tilde{U}^{n}_{i\pm1/2,j}$ and $\tilde{U}^{n}_{i,j\pm1/2}$ are the Riemann problem solutions at the right, left, top, and bottom edges of cell (i, j), respectively. The second of the two terms discretizing the source term $H(U, E)$ depends upon $U^{*n+1}_{i,j}$, which is a provisional value at time t^{n+1} computed using approximate values for the state variables at t^{n+1}, as well as the field E^{n+1} at the new time. Since the latter quantity is not yet known, we compute only the provisional value $U^{*n+1}_{i,j}$ and the flux divergences $\nabla \cdot (n_i u_i)$, which are required in Step 3 below, at this stage of the algorithm. The final ion update (3.21) will be completed after the new field is computed.

Step 2. Calculate electron diffusive flux. We next compute a slope-limited, Taylor-series prediction $n_e^{n+1/2}$ of the edge electron density at $t_{n+1/2}$ upwinded relative to the drift velocity $\bar{u}_e^n$. We then combine (2.1) and (2.2) to solve

$$\frac{\hat{n}_e^{n+1} - n_e^n}{\Delta t} = -\nabla \cdot \left(n_e^{n+1/2}\bar{u}_e^n\right) + \nabla \cdot \eta^n \nabla\left(\hat{n}_e^{n+1} k T_e^n\right) + \sum_j R_{ej}^n \tag{3.22}$$

for $\hat{n}_e^{n+1}$, which is a prediction of the electron density at time t_{n+1} that we employ solely in order to compute the electron diffusive flux

$$\Gamma_{\text{diff}}^{n+1} \equiv -\eta^n \nabla\left(\hat{n}_e^{n+1} k T_e^n\right). \tag{3.23}$$

Step 3. Calculate new potential and electron density. Using the electron diffusive flux $\Gamma_{\text{diff}}^{n+1}$ and the electron density $n_e^{n+1/2}$, we advance the potential by solving

$$\nabla \cdot \left[\frac{\epsilon_0}{e}\nabla\phi^{n+1} + \Delta t F(\phi^{n+1})\right] = n_e^n - \sum_i \frac{q_i}{e} n_i^n + \Delta t \nabla \cdot \left(\sum_i \frac{q_i}{e}(n_i u_i)^{n+1/2} - \Gamma_{\text{diff}}^{n+1}\right), \tag{3.24}$$

where

$$F(\phi^{n+1}) \equiv \begin{cases} n_e^{n+1/2}\mu^n \nabla\phi^{n+1} & \text{on interior cell edges} \\ n_e^{n+1/2}\bar{w}_e\left(kT_e^n, \phi^{n+1}\right) & \text{on boundary cell edges.} \end{cases} \tag{3.25}$$

We then compute the new electron density via

$$n_e^{n+1} = n_e^n - \Delta t \nabla \cdot \Gamma_e^{n+1} + \Delta t \sum_j R_{ej}^n, \tag{3.26}$$

where

$$\Gamma_e^{n+1} \equiv F(\phi^{n+1}) + \Gamma_{\text{diff}}^{n+1} \tag{3.27}$$

and $\Gamma_{\text{diff}}^{n+1}$ is assumed to vanish at the problem boundary. The new field is

$$E^{n+1} \equiv -\nabla\phi^{n+1}. \tag{3.28}$$

Note that the implicit treatment of the electron drift flux in (3.24) transforms an otherwise easy-to-solve Poisson equation into a nonlinear (due to the electron flux boundary condition) equation with a variable-coefficient Jacobian. The scaling analysis presented in [9] shows, however, that this modification is essential to avoid a severe time step restriction. Fortunately, given time steps satisfying (3.15), the nonlinear problem (3.24) can be adequately solved using only a couple of Newton iterations after the first few time steps, during which the performance of the Newton iteration depends upon the prescribed initial conditions. The Jacobian system is solved in each Newton step using a multigrid algorithm [9].

Step 4. Calculate new electron internal energy. Using n_e^{n+1}, Γ_e^{n+1}, and E^{n+1}, we obtain the internal electron energy kT_e^{n+1} at the new time by solving

$$\begin{aligned}
&\frac{3}{2}\frac{n_e^{n+1}kT_e^{n+1} - n_e^n kT_e^n}{\Delta t} + \nabla\cdot\left(\frac{5}{2}\Gamma_e^{n+1}kT_e^{n+1/2}\right) \\
&\quad = -e\Gamma_e^{n+1}\cdot E^{n+1} + \frac{5}{2}\nabla\cdot\left(\eta^n n_e^{n+1/2}kT_e^{n+1/2}\nabla kT_e^{n+1}\right) + P_{\text{ind}}^n \\
&\qquad - \sum_j \frac{3m_e}{m_j}kT_e^n\nu_{ej}^n n_e^n + \sum_j \epsilon_{ej}^n S_{ej}^n,
\end{aligned} \tag{3.29}$$

where $T_e^{n+1/2}$ is a slope-limited, Taylor-series prediction of the edge temperature at $t^{n+1/2}$ upwinded relative to the velocity u_e^n. Collecting all terms involving T_e^{n+1} on the left-hand side and incorporating the boundary conditions (2.14), we observe that (3.29) is a linear system with a symmetric, positive definite coefficient matrix.

Step 5. Complete calculation of new ion data. The last step is the computation of the ion data at the new time step via (3.21).

4. SOLUTION OF THE FLUID EQUATIONS ON LOCALLY REFINED GRIDS

Let Ω_c denote a union of disjoint, uniform, rectangular grids. Let Ω_f also denote a union of disjoint, uniform, rectangular grids, each of which is obtained by refining a rectangular subgrid of Ω_c by a factor n_{ref}. Let $P(\Omega_f)$ denote the projection of the *fine grid* Ω_f onto the *coarse grid* Ω_c. Define the *composite grid* by

$$\Omega_{\text{comp}} \equiv \Omega_f \cup (\Omega_c \backslash P(\Omega_f)) \tag{4.30}$$

and let I denote the interface between the coarse grid component $\Omega_c \backslash P(\Omega_f)$ and the fine grid component Ω_f of Ω_{comp} (see Fig. 4.1).

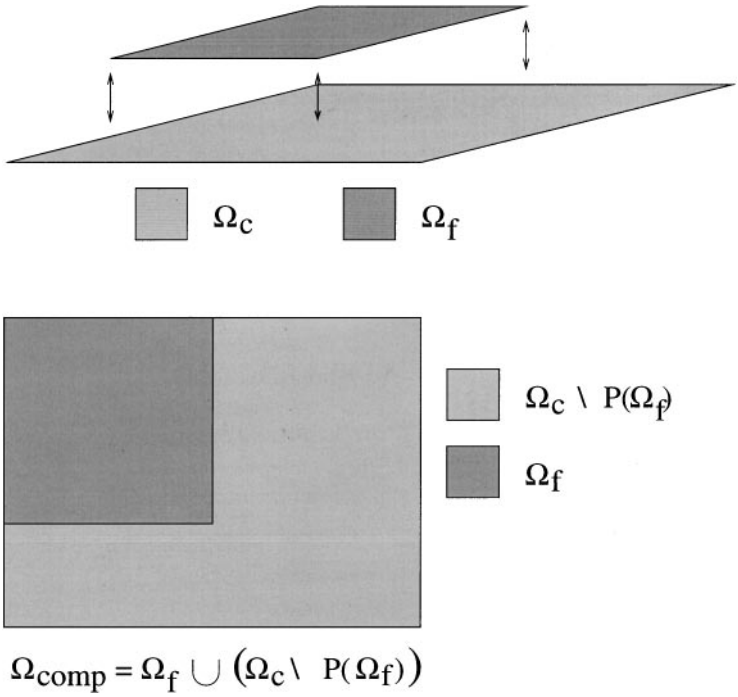

FIG. 4.1. Coarse, fine, and composite grids.

4.1. Solution of Poisson's Equation on a Composite Grid

We want to extend the algorithm described in the preceding section for use on the composite grid (4.30). A key requirement is the enforcement of Poisson's equation

$$-\epsilon_0 \Delta \phi = \rho \tag{4.31}$$

at each time step, so we begin by describing what it means to solve (4.31) on (4.30). As in the single grid case, we will not be solving (4.31) directly during the integration of the fluid system on the composite grid, except for initialization or diagnostic purposes. Nevertheless, the separate consideration of (4.31) here will introduce some important concepts and notation.

On Ω_f or Ω_c, the standard five-point cell-centered discretization of (4.31) can be expressed in terms of edge-centered fields as

$$\epsilon_0(-E_{i-1/2,j} + E_{i+1/2,j} - E_{i,j-1/2} + E_{i,j+1/2}) = h\rho_{i,j}, \tag{4.32}$$

where, e.g.,

$$E_{i+1/2,j} \equiv -(\phi_{i+1,j} - \phi_{i,j})/h, \tag{4.33}$$

$\phi_{i,j}$ and $\rho_{i,j}$ are the potential and charge at cell center (i, j), respectively, and h is the grid size. Dirichlet conditions are imposed on the physical boundary by extrapolating interior potential values and the prescribed boundary value to a "ghost" cell exterior to the problem, as is depicted in Fig. 4.2(a).

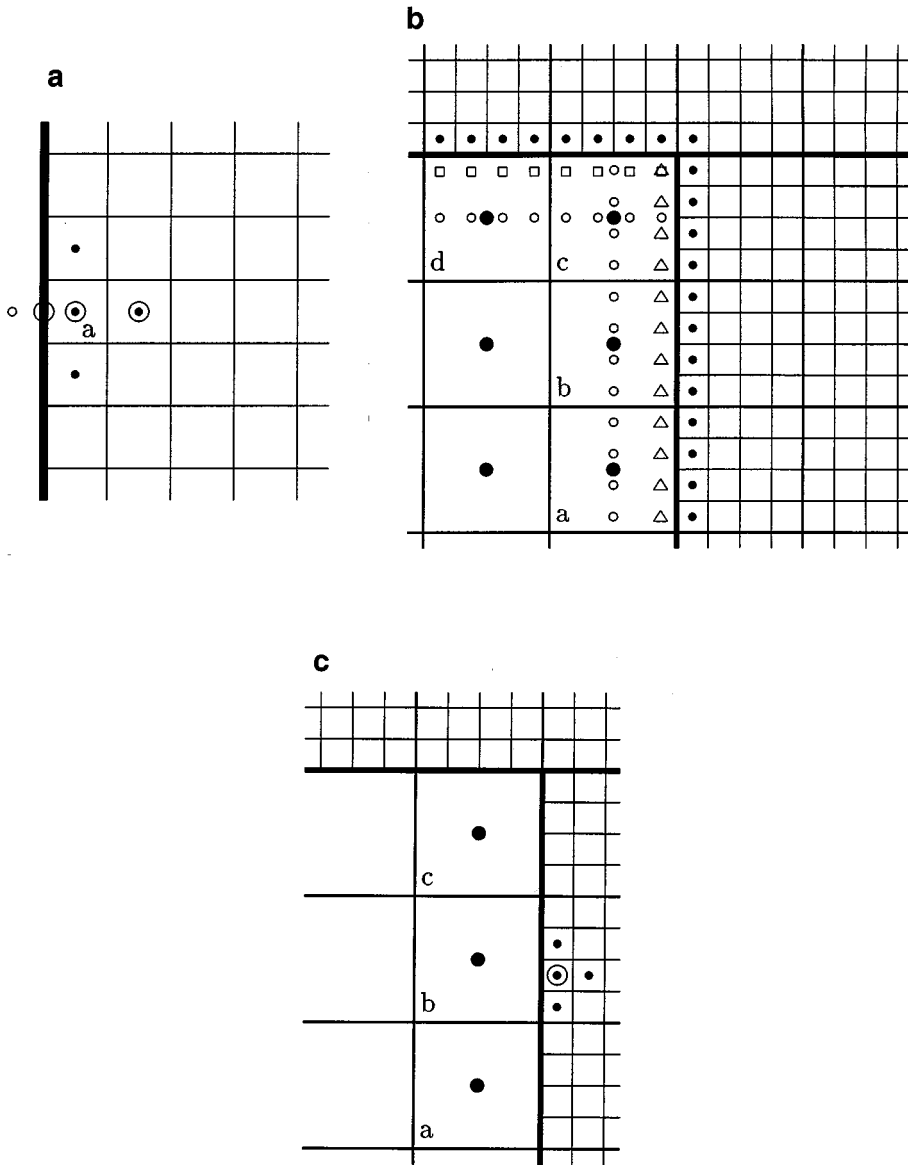

FIG. 4.2. (a) At a physical boundary, interior and boundary values (○) are used to extrapolate to the ghost cell (○); the ghost value and the other interior values (●) are used to construct the Laplacian at (a). (b) Locations of coarse grid boundary conditions (●), tangentially interpolated values (○), fine grid cells (●), and ghost cells (△ and □). (c) Domain of dependence (● and ●) of the Laplacian at a fine cell (○) adjacent to the coarse/fine interface.

A composite discretization of (4.31) on (4.30) is obtained by applying (4.32) separately on Ω_f and $\Omega_c \backslash P(\Omega_f)$ with the definition of the edge-centered fields on I suitably modified to enforce continuity of the potential and field across the interface. Since each coarse grid edge on I is the union of n_{ref} fine grid edges, field continuity requires that the coarse fields at such edges be obtained as the average of the fields on the corresponding fine edges.

To obtain the fine grid fields on I, we require potentials on $\Omega_c \backslash P(\Omega_f)$ interpolated to the fine grid resolution and centered in a single layer of ghost cells surrounding Ω_f, which

then enables (4.33) to also be applied on fine grid edges contained in I. We employ an interpolation scheme whose stencil is depicted in Fig. 4.2(b) for a sample grid structure with $n_{\text{ref}} = 4$. The potentials at coarse cell centers (large filled circles) are first quadratically interpolated tangentially to I to obtain values at the locations depicted by the unfilled circles. These values are then linearly interpolated normally to I using the potential values at the centers of the fine cells abutting I (small filled circles) to obtain potentials at the ghost cell centers depicted by the triangles and squares. As suggested by the overlapping triangle and rectangle in the upper right-hand corner of the coarse grid region in Fig. 4.2(b), the potentials interpolated in this way may not be unique, with the value denoted by the triangle being used to compute the field on the vertical fine edge to the right, and the value denoted by the square being used to compute the field on the horizonal fine edge immediately above.

The resulting discretization of (4.31) on Ω_{comp} is second-order accurate. The corresponding discrete Laplacian operator is not symmetric, however, as can be deduced from Fig. 4.2(c), which displays the operator stencil based at the circled fine grid cell abutting I. For example, the coarse cell labeled "a" participates in the stencil based at the circled fine grid cell, but the converse is not the case. Effective multilevel algorithms can nevertheless be employed to solve the discrete system, as will be described in Section 4.4 below. These multilevel algorithms rely heavily upon the observation that the solution ϕ of the composite Poisson equation is also obtainable as the solution of a coupled pair of Poisson equations on Ω_f and Ω_c,

$$-\epsilon_0 \Delta \phi_f = \rho_f \equiv \rho \text{ on } \Omega_f, \qquad \phi_f = Q(\phi_f, \phi_c) \text{ at } I \tag{4.34}$$

$$-\epsilon_0 \Delta \phi_c = \begin{cases} \rho - \epsilon_0 \nabla \cdot \delta E & \text{on } \Omega_c \backslash P(\Omega_f) \\ \langle \rho_f \rangle_{P(\Omega_f)} & \text{on } P(\Omega_f) \end{cases} \tag{4.35}$$

by setting

$$\phi = \begin{cases} \phi_c & \text{on } \Omega_c \backslash P(\Omega_f) \\ \phi_f & \text{on } \Omega_f. \end{cases} \tag{4.36}$$

Here, Q denotes the linear boundary operator at I implicitly defined by the previously described high-order tangential interpolation of coarse grid potentials ϕ_c combined with normal interpolation of fine grid potentials ϕ_f. $\langle \rho_f \rangle_{P(\Omega_f)}$ denotes the volume-weighted average of ρ onto $P(\Omega_f)$. In (4.35),

$$\delta E \equiv -\langle \nabla \phi_f \rangle_I + \nabla \phi_c, \tag{4.37}$$

where $\langle \cdot \rangle_I$ denotes the arithmetic averaging of fine edge data to coarse edge data on I. Here, and in the remainder of this section, we adopt the convention that δF denotes the difference on coarse edges in I of a generic quantity F computed on the coarse grid and fine grids. For such quantities, it is also convenient to define the divergence $\nabla \cdot \delta F$ as the divergence of the extension by zero of δF to the remaining edges of Ω_c. The term $\epsilon_0 \nabla \cdot \delta E$ has the form of an integrated surface charge, but in fact results solely from the change of grid resolution at I. In the solution of (4.35), this "coarse–fine" surface charge causes a jump in the coarse grid field at I such that continuity of the composite field is obtained.

Because only the coarse grid potential ϕ_c on $\Omega_c \backslash P(\Omega_f)$ is used in constructing the composite solution (4.36), it might at first seem inefficient to be solving (4.35) on the entire

coarse grid Ω_c versus solving just on $\Omega_c \backslash P(\Omega_f)$ and Ω_f with matching conditions at I. This is not the case, however, since we intend to solve (4.34)–(4.35) with a multilevel method in which the approximate solutions of (4.35) in residual-correction form computed on $P(\Omega_f)$ represent coarse grid corrections to smoothed approximate solutions of (4.34). This results in a very effective iterative solution method. This approach also avoids the need to compute solutions only on $\Omega_c \backslash P(\Omega_f)$, which may be far more geometrically complex than either grid Ω_c or grid Ω_f.

4.2. Solution of the Plasma Fluid System on a Composite Grid

With the composite solution of Poisson's equation as background, let us now consider the integration of the plasma fluid system. As in the single grid case, discrete time evolution equations on the composite grid can be obtained by combining a first-order difference of Poisson's equation over a time step $\Delta t = t^{n+1} - t^n$ with the continuity equations for the ions and electrons. Again with the electron drift flux treated implicitly, this yields

$$n_e^{n+1} = n_e^n - \Delta t \nabla \cdot \left(\Gamma_{\text{drift}}\left(n_e^{n+1/2}, \phi^{n+1}\right) + \Gamma_{\text{diff}}\right) + \Delta t S^n \tag{4.38}$$

$$n_i^{n+1} = n_i^n - \Delta t \nabla \cdot \Gamma_i + \Delta t S^n \tag{4.39}$$

$$\begin{aligned} &\nabla \cdot \left(\frac{\epsilon_0}{e} \nabla \phi^{n+1} + \Delta t \Gamma_{\text{drift}}\left(n_e^{n+1/2}, \phi^{n+1}\right) \right) \\ &\quad = n_e^n - \sum_i \frac{q_i}{e} n_i^n + \Delta t \left(\sum_i \frac{q_i}{e} \nabla \cdot \Gamma_i - \nabla \cdot \Gamma_{\text{diff}} \right) \end{aligned} \tag{4.40}$$

in addition to evolution equations for the ion momentum and electron temperature, which we omit for now. In (4.38) and (4.39), S^n denotes the source/sink term.

Implied in (4.38)–(4.40) are matching conditions at I that couple the dependent variables on Ω_f and $\Omega_c \backslash P(\Omega_f)$. Conservation of mass (i.e., hyperbolic matching conditions) requires that

$$\Gamma_{x,c} = \langle \Gamma_{x,f} \rangle_I, \qquad x = i, \text{drift}, \text{diff}, \tag{4.41}$$

where $\Gamma_{x,f}$ and $\Gamma_{x,c}$ are the fine and coarse fluxes on I, respectively. Continuity of the potential and field (i.e., second-order elliptic operator matching conditions) is achieved by enforcing

$$\phi_f^{n+1} = Q\left(\phi_f^{n+1}, \phi_c^{n+1}\right) \tag{4.42}$$

$$n \cdot \nabla \phi_c^{n+1} = \left\langle n \cdot \nabla \phi_f^{n+1} \right\rangle_I \tag{4.43}$$

on I, where n is a unit normal to I.

Although the composite system could be integrated directly, such an approach would be penalized by the need to use a single time step for both spatial grid resolutions. In particular, given comparable electron drift velocities, the time step will be limited by the stability requirements on Ω_f, which might represent only a very small fraction of the physical domain. To avoid the resulting unnecessary additional work on $\Omega_c \backslash P(\Omega_f)$, we instead employ a predictor–corrector approach comprising separate integrations on the coarse and fine grids coupled by a composite sychronization step. Specifically,

1. *Coarse predictor step*. Use the single grid algorithm of Section 3 to integrate the plasma fluid system on Ω_c over a time step $\Delta t_c = t^{n+1} - t^n$.

2. *Fine predictor step*. Integrate the system on Ω_f over the same time interval t^n to t^{n+1} using n_{cycle} uniform time steps $\Delta t_f \equiv \Delta t_c / n_{\text{cycle}}$, with boundary values interpolated spatially and temporally from $\Omega_c \backslash P(\Omega_f)$.

3. *Composite synchronization step*. Solve for a potential correction on the composite grid to restore matching conditions at the coarse–fine interface I that were violated during the separate coarse and fine integrations.

The corrected coarse and fine predictions are then combined to obtain a composite solution that closely approximates, with mild assumptions, the solution one would have obtained by integrating the composite system using the fine grid time steps everywhere. That is, letting $m = m(n), \ldots, m(n+1)$ index the time steps taken on the fine grid, we obtain approximations to the composite quantities $n_e^{m(n+1)}$, $n_i^{m(n+1)}$, and $\phi^{m(n+1)}$ satisfying

$$n_e^{m(n+1)} = n_e^{m(n)} - \sum_m \Delta t_f \left[\nabla \cdot \left(\Gamma_{\text{drift}}\left(n_e^{m+1/2}, \phi^{m+1}\right) + \Gamma_{\text{diff}}^{m+1/2} \right) - S^m \right] \tag{4.44}$$

$$n_i^{m(n+1)} = n_i^{m(n)} - \sum_m \Delta t_f \left(\nabla \cdot \Gamma_i^{m+1/2} - S^m \right) \tag{4.45}$$

$$\begin{aligned} &\nabla \cdot \left(\frac{\epsilon_0}{e} \nabla \phi^{m(n+1)} + \sum_m \Delta t_f \Gamma_{\text{drift}}\left(n_e^{m+1/2}, \phi^{m+1}\right) \right) \\ &\quad = n_e^{m(n)} - \sum_i \frac{q_i}{e} n_i^{m(n)} + \sum_m \Delta t_f \left(\sum_i \frac{q_i}{e} \nabla \cdot \Gamma_i^{m+1/2} - \nabla \cdot \Gamma_{\text{diff}}^{m+1/2} \right) \end{aligned} \tag{4.46}$$

with the conditions analogous to (4.41)–(4.43) on I. We now describe this algorithm in more detail.

4.2.1. *Coarse Predictor Step*

On Ω_c, select a time step $\Delta t_c = t^{n+1} - t^n$ and perform the integration

$$n_{e,c}^{n+1} = n_{e,c}^n - \Delta t_c \nabla \cdot \left(\Gamma_{\text{drift}}\left(n_{e,c}^{n+1/2}, \phi_c^{n+1}\right) + \Gamma_{\text{diff},c} \right) + \Delta t_c S_c^n \tag{4.47}$$

$$n_{i,c}^{n+1} = n_{i,c}^n - \Delta t_c \nabla \cdot \Gamma_{i,c} + \Delta t_c S_c^n \tag{4.48}$$

$$\begin{aligned} &\nabla \cdot \left(\frac{\epsilon_0}{e} \nabla \phi_c^{n+1} + \Delta t_c \Gamma_{\text{drift}}\left(n_{e,c}^{n+1/2}, \phi_c^{n+1}\right) \right) \\ &\quad = \frac{\epsilon_0}{e} \nabla \cdot \delta E^n + \Delta t_c \nabla \cdot \left(n_{e,c}^{n+1/2} \mu^n \delta E^n \right) + n_e^n - \sum_i \frac{q_i}{e} n_i^n \\ &\qquad + \Delta t_c \left(\sum_i \frac{q_i}{e} \nabla \cdot \Gamma_{i,c} - \nabla \cdot \Gamma_{\text{diff},c} \right). \end{aligned} \tag{4.49}$$

This is the same single-level integration described in Section 3 except for the first two terms in the right-hand side of (4.49) involving δE^n, which is computed as in (4.37) using ϕ_f^n and ϕ_c^n. We include the δE^n terms in (4.49) as approximations of the analogous quantities at the new time t^{n+1} to anticipate better the effect of the subsequent fine grid integrations. As

will be seen later, this modification of the coarse grid prediction reduces the magnitude of the correction required in the composite synchronization step.

4.2.2. *Fine Predictor Step*

Next, we integrate the system on Ω_f over the same time interval t^n to t^{n+1} using n_{cycle} uniform time steps $\Delta t_f \equiv \Delta t_c / n_{\text{cycle}}$. Let $m = m(n), \ldots, m(n+1)$ index the time steps taken on the fine grid. We have

$$n_{e,f}^{m(n+1)} = n_{e,f}^{m(n)} - \sum_m \Delta t_f \left[\nabla \cdot \left(\Gamma_{\text{drift}}\left(n_{e,f}^{m+1/2}, \phi_f^{m+1}\right) + \Gamma_{\text{diff},f}^{m+1/2}\right) - S_f^m\right] \tag{4.50}$$

$$n_{i,f}^{m(n+1)} = n_{i,f}^{m(n)} - \sum_m \Delta t_f \left(\nabla \cdot \Gamma_{i,f}^{m+1/2} - S_f^m\right) \tag{4.51}$$

$$\begin{aligned} &\nabla \cdot \left(\frac{\epsilon_0}{e} \nabla \phi_f^{m(n+1)} + \sum_m \Delta t_f \Gamma_{\text{drift}}\left(n_{e,f}^{m+1/2}, \phi_f^{m+1}\right)\right) \\ &\quad = n_e^{m(n)} - \sum_i \frac{q_i}{e} n_i^{m(n)} + \sum_m \Delta t_f \left(\sum_i \frac{q_i}{e} \nabla \cdot \Gamma_{i,f}^{m+1/2} - \nabla \cdot \Gamma_{\text{diff},f}^{m+1/2}\right). \end{aligned} \tag{4.52}$$

In performing this integration, boundary values are needed at the coarse–fine interface I. These values are obtained from the corresponding values on $\Omega_c \backslash P(\Omega_f)$ by linear interpolation in time and appropriate spatial interpolations. Specifically, potential boundary values at I are given by

$$\phi_f^m = Q\left(\phi_f^m, \frac{m(n+1) - m}{n_{\text{cycle}}} \phi_c^n + \frac{m - m(n)}{n_{\text{cycle}}} \phi_c^{n+1}\right), \qquad m = m(n)+1, \ldots, m(n+1). \tag{4.53}$$

Boundary values for the fine grid electron and ion data at I are obtained from conservative spatial interpolation of their corresponding values on $\Omega_c \backslash P(\Omega_f)$, linearly interpolated in time, to a layer of ghost cells surrounding Ω_f.

4.2.3. *Composite Synchronization Step*

Having integrated the fluid system on both Ω_f and $\Omega_c \backslash P(\Omega_f)$, we would like to compose a solution of (4.44)–(4.46) from the integrated solutions obtained from (4.47)–(4.49) and (4.50)–(4.52). However, it is not possible to do this immediately for two reasons. First, the interpolated Dirichlet boundary condition (4.53) only ensures that the potential is continuous at I, but nothing has been done to enforce field continuity there. In other words, Poisson's equation is not yet satisfied since the composite potential still needs to be adjusted to satisfy also the interface conditions (4.43). Second, since different fluxes were used on I during the integration of the coarse and fine grids, (4.41) does not necessarily hold and conservation must be restored by an appropriate "refluxing" of ions and electrons. Such a redistribution of charge again implies a modification of the composite potential.

One solution to this dilemma would be to iterate the coarse and fine grid integrations until we obtain a composite solution that satisfies the hyperbolic and elliptic matching conditions. This is clearly computationally unattractive. Another alternative would be to determine corrections to the fine and coarse solutions in order to obtain a solution satisfying

the matching conditions. If we were to formally subtract the coarse and fine solutions from the composite solution, we would hope to obtain an equation for these corrections. With very mild approximations, this is what we indeed obtain. The approximations we make are based on three principles.

The first principle is that the inertia of the ions makes them insensitive to changes in the potential over the relatively small time steps, which are determined by the significantly higher electron mobility. The ion density and momenta will not be immediately affected by small changes in the field. Consequently, we consider charge redistribution due to composite potential corrections as entirely due to a change in the electron density. It is also sufficient to include field corrections in the momentum source term (3.20) in subsequent time steps.

Second, for quasi-neutral plasmas, perturbations to the electron density (for a single coarse step) are very important as they affect the space charge, but they are not important for quantities involving the density alone. For example, a change of 1 part in 1000 for the electron density will have little effect on the calculated pressure gradient, but it may drastically change the space charge. This observation is used in three approximations. First, the edge centered electron densities used in the fine and coarse modified Poisson solves are approximately equal to those we would have if we were integrating the composite system itself. The same can be said regarding the diffusive flux terms used for the fine and coarse integrations. Third, the changes in ion and electron densities due to the rates R_{ej} and R_{ij} do not affect the net charge and therefore do not require correction after the separate integration of the coarse and fine grid systems.

Our final observation is that the electron temperature and the dependent rate coefficients change slowly over a time step. We may therefore neglect changes in the rate coefficients due to the composite corrections.

With these assumptions, we proceed in deriving equations for the potential and electron density corrections that allow us to satisfy the governing equations along with the hyperbolic and elliptic matching conditions.

We seek a potential correction $\phi^* = (\phi_f^*, \phi_c^*)$ satisfying the composite system obtained bysubstituting

$$\phi^m \equiv \begin{cases} \phi_f^m + \phi_f^* & \text{on } \Omega_f, \\ \phi_c^{n+1} + \phi_c^* & \text{on } \Omega_c \backslash P(\Omega_f), \end{cases} \qquad m = m(n)+1, \ldots, m(n+1) \quad (4.54)$$

into (4.46), then subtracting (4.49) on $\Omega_c \backslash P(\Omega_f)$ and (4.52) on Ω_f. The correction must also satisfy the continuity requirement

$$\phi_f^* = Q(\phi_f^*, \phi_c^*) \quad (4.55)$$

at the coarse–fine boundary I. Let us now determine the equation to be satisfied by ϕ^* on $\Omega_c \backslash P(\Omega_f)$. On edges of $\Omega_c \backslash P(\Omega_f)$ contained in I, define

$$\delta\Gamma_i \equiv \left\langle \frac{1}{\Delta t_c} \sum_m \Delta t_f \Gamma_{i,f}^{m+1/2} \right\rangle_I - \Gamma_{i,c} \quad (4.56)$$

$$\delta\Gamma_{\text{diff}} \equiv \left\langle \frac{1}{\Delta t_c} \sum_m \Delta t_f \Gamma_{\text{diff},f}^{m+1/2} \right\rangle_I - \Gamma_{\text{diff},c}. \quad (4.57)$$

Substituting (4.54) into (4.46), with $\Delta t = \Delta t_c$, then subtracting (4.49) on $\Omega_c \backslash P(\Omega_f)$, we obtain

$$\nabla \cdot \left[\frac{\epsilon_0}{e} \nabla \left(\phi^{m(n+1)} - \phi_c^{n+1} \right) + \Delta t \Gamma^*_{\text{drift}} \right] = -\frac{\epsilon_0}{e} \nabla \cdot \delta E^n - \Delta t \nabla \cdot \left(n_{e,c}^{n+1/2} \mu^n \delta E^n \right) + \Delta t \nabla \cdot \left(\sum_i \frac{q_i}{e} \delta \Gamma_i - \delta \Gamma_{\text{diff}} \right), \tag{4.58}$$

where

$$\Gamma^*_{\text{drift}} \equiv \begin{cases} \left\langle \frac{1}{\Delta t_c} \sum_m \Delta t_f \Gamma_{\text{drift}} \left(n_e^{m+1/2}, \phi^{m+1} \right) \right\rangle_I & \text{on edges of } \Omega_c \backslash P(\Omega_f) \\ \quad - \Gamma_{\text{drift}} \left(n_{e,c}^{n+1/2}, \phi_c^{n+1} \right) & \text{contained in } I \\ \frac{1}{\Delta t_c} \sum_m \Delta t_f \Gamma_{\text{drift}} \left(n_e^{m+1/2}, \phi^{m+1} \right) & \text{on edges of } \Omega_c \backslash P(\Omega_f) \\ \quad - \Gamma_{\text{drift}} \left(n_{e,c}^{n+1/2}, \phi_c^{n+1} \right) & \text{not contained in } I. \end{cases} \tag{4.59}$$

In the first term of (4.58), we observe that

$$\nabla \left(\phi^{m(n+1)} - \phi_c^{n+1} \right) = \begin{cases} \nabla \phi_c^* & \text{on edges of } \Omega_c \backslash P(\Omega_f) \text{ not contained in } I \\ \langle \nabla \phi_f^* \rangle_I & \text{on edges of } \Omega_c \backslash P(\Omega_f) \text{ contained in } I. \end{cases} \tag{4.60}$$

On edges of $\Omega_c \backslash P(\Omega_f)$ not contained in I and not part of the physical boundary, (4.59) yields

$$\Gamma^*_{\text{drift}} = n_{e,c}^{n+1/2} \mu^n \nabla \phi_c^*. \tag{4.61}$$

Using (4.59), on the edges of $\Omega_c \backslash P(\Omega_f)$ contained in I, we have

$$\Gamma^*_{\text{drift}} = \left\langle \frac{1}{\Delta t_c} \sum_m \Delta t_f n_e^{m+1/2} \mu^m \left(\nabla \phi_f^{m+1} + \nabla \phi^* \right) \right\rangle_I - n_{e,c}^{n+1/2} \mu^n \nabla \phi_c^{n+1} \tag{4.62}$$

$$\approx \left\langle \frac{1}{\Delta t_c} \sum_m \Delta t_f \Gamma_{\text{drift}} \left(n_{e,f}^{m+1/2}, \phi_f^{m+1} \right) \right\rangle_I - \Gamma_{\text{drift}} \left(n_{e,c}^{n+1/2}, \phi_c^{n+1} \right) + \left\langle n_e^{m(n+1)-1/2} \mu^{m(n+1)-1} \right\rangle_I \langle \nabla \phi^* \rangle_I, \tag{4.63}$$

in which we make the approximation

$$\left\langle n_e^{m+1/2} \mu^m \nabla \phi^* \right\rangle_I \approx \left\langle n_e^{m+1/2} \mu^m \right\rangle_I \langle \nabla \phi^* \rangle_I. \tag{4.64}$$

Therefore, if we let

$$\Gamma_{\text{drift}}(\phi_c^*) \equiv \begin{cases} n_e^{n+1/2} \mu^n \nabla \phi_c^* & \text{on edges of } \Omega_c \backslash P(\Omega_f) \text{ not contained in } I \text{ or the physical boundary} \\ \left\langle n_{e,f}^{m(n+1)-1/2} \mu^{m(n+1)-1} \right\rangle_I \nabla \phi_c^* & \text{on edges of } \Omega_c \backslash P(\Omega_f) \text{ contained in } I \end{cases} \tag{4.65}$$

on the edges of $\Omega_c \backslash P(\Omega_f)$ not contained in the physical boundary, and define

$$\delta E^{n+1} \equiv -\left\langle \nabla \phi_f^{n+1} \right\rangle_I + \nabla \phi_c^{n+1} \tag{4.66}$$

$$\begin{aligned} \delta\Gamma_e \equiv & \left\langle \frac{1}{\Delta t_c} \sum_m \Delta t_f \Gamma_{\text{drift}}\left(n_{e,f}^{m+1/2}, \phi_f^{m+1}\right) \right\rangle_I - \Gamma_{\text{drift}}\left(n_{e,c}^{n+1/2}, \phi_c^{n+1}\right) \\ & + n_{e,c}^{n+1/2} \mu^n \delta E^n + \delta\Gamma_{\text{diff}} \end{aligned} \tag{4.67}$$

on the edges of $\Omega_c \backslash P(\Omega_f)$ contained in I, (4.58) becomes

$$\begin{aligned} & \nabla \cdot \left(\frac{\epsilon_0}{e} \nabla \phi_c^* + \Delta t \Gamma_{\text{drift}}(\phi_c^*) \right) \\ & = -\nabla \cdot \left(\left(\frac{\epsilon_0}{e} + \left\langle \Delta t n_{e,f}^{m(n+1)-1/2} \mu^{m(n+1)-1} \right\rangle_I \right) (\langle \nabla \phi_f^* \rangle_I - \nabla \phi_c^*) \right) \\ & \quad + \frac{\epsilon_0}{e} \nabla \cdot (\delta E^{n+1} - \delta E^n) + \Delta t \nabla \cdot \left(\sum_i \frac{q_i}{e} \delta\Gamma_i - \delta\Gamma_e \right), \end{aligned}$$

$$\phi_c^* = 0 \text{ on the physical boundary.} \tag{4.68}$$

To obtain the equation determining ϕ^* on Ω_f, we subtract (4.52) from (4.46), which yields

$$\nabla \cdot \left(\frac{\epsilon_0}{e} \nabla \phi_f^* + \Delta t \Gamma_{\text{drift}}^* \right) = 0, \quad \phi_f^* = \begin{cases} Q(\phi_f^*, \phi_c^*) & \text{at } I, \\ 0 & \text{on the physical boundary,} \end{cases} \tag{4.69}$$

where

$$\Gamma_{\text{drift}}^* \equiv \frac{1}{\Delta t_c} \sum_m \Delta t_f \left[\Gamma_{\text{drift}}\left(n_e^{m+1/2}, \phi^{m+1}\right) - \Gamma_{\text{drift}}\left(n_{e,f}^{m+1/2}, \phi_f^{m+1}\right) \right]. \tag{4.70}$$

On the non-boundary edges of Ω_f, Γ_{drift}^* is approximated by

$$\Gamma_{\text{drift}}(\phi_f^*) \equiv n_{e,f}^{m(n+1)-1/2} \mu^{m(n+1)-1} \nabla \phi_f^*. \tag{4.71}$$

Hence, on Ω_f,

$$\nabla \cdot \left(\frac{\epsilon_0}{e} \nabla \phi_f^* + \Delta t \Gamma_{\text{drift}}(\phi_f^*) \right) = 0, \quad \phi_f^* = \begin{cases} Q(\phi_f^*, \phi_c^*) & \text{at } I, \\ 0 & \text{on the physical boundary.} \end{cases} \tag{4.72}$$

The pair of equations (4.68) and (4.72) thus form the system to be solved for the composite potential correction ϕ^*. It only remains to specify the modifications to be made on physical boundary edges due to (2.12). On the edges of $\Omega_c \backslash P(\Omega_f)$ on the physical boundary, (4.59)

implies that

$$
\begin{aligned}
&n\cdot\Gamma^{*}_{\text{drift}}\\
&\quad=\frac{1}{\Delta t_c}\sum_m \Delta t_f\, n_{e,c}^{m+1/2}\frac{\overline{v}_e^m}{4}\exp\left[-\frac{e\left(\phi^{m+1}-\phi_b^{m+1}\right)}{kT_e^m}\right]-n_{e,c}^{n+1/2}\frac{\overline{v}_e^n}{4}\exp\left[-\frac{e\left(\phi_c^{n+1}-\phi_b^{m+1}\right)}{kT_e^n}\right]\\
&\quad\approx n_{e,c}^{n+1/2}\frac{\overline{v}_e^n}{4}\left\{\frac{1}{\Delta t_c}\sum_m \Delta t_f\exp\left[-\frac{e\left(\phi^{m+1}-\phi_b^{m+1}\right)}{kT_e^n}\right]-\exp\left[-\frac{e\left(\phi_c^{n+1}-\phi_b^{m+1}\right)}{kT_e^n}\right]\right\}\\
&\quad=n_{e,c}^{n+1/2}\frac{\overline{v}_e^n}{4}\exp\left[-\frac{e\left(\phi_c^{n+1}-\phi_b^{m+1}\right)}{kT_e^n}\right]\left[\exp\left(-\frac{e\phi^{*}}{kT_e^n}\right)-1\right]\\
&\quad=n\cdot\Gamma_{\text{drift}}\left(n_{e,c}^{n+1/2},\phi_c^{n+1}\right)\left[\exp\left(-\frac{e\phi^{*}}{kT_e^n}\right)-1\right]
\end{aligned}
\tag{4.73}
$$

and similarly on the boundary edges in Ω_f. Thus, (4.65) and (4.71) are supplemented by

$$
n\cdot\Gamma_{\text{drift}}(\phi_c^{*})\equiv n\cdot\Gamma_{\text{drift}}\left(n_{e,c}^{n+1/2},\phi_c^{n+1}\right)\left[\exp\left(-\frac{e}{kT_e^n}\phi_c^{*}\right)-1\right] \tag{4.74}
$$

$$
n\cdot\Gamma_{\text{drift}}(\phi_f^{*})\equiv n\cdot\Gamma_{\text{drift}}\left(n_{e,f}^{m(n+1)-1/2},\phi_f^{m(n+1)}\right)\left[\exp\left(-\frac{e}{kT_e^n}\phi^{*}\right)-1\right], \tag{4.75}
$$

respectively.

Similar subtractions of the ion and electron continuity equations show that

$$
n_i^{m(n+1)}\equiv\begin{cases}n_{i,f}^{m(n+1)} & \text{on } \Omega_f\\ n_{i,c}^{n+1}-\Delta t\nabla\cdot\delta\Gamma_i & \text{on } \Omega_c\backslash P(\Omega_f)\end{cases} \tag{4.76}
$$

$$
n_e^{m(n+1)}\equiv\begin{cases}n_{e,f}^{m(n+1)}-\Delta t\nabla\cdot\Gamma_{\text{drift}}(\phi^{*}) & \text{on } \Omega_f\\ n_{e,c}^{n+1}-\Delta t\nabla\cdot\left(\Gamma_{\text{drift}}(\phi^{*})+\delta\Gamma_e\right) & \text{on } \Omega_c\backslash P(\Omega_f)\end{cases} \tag{4.77}
$$

satisfy (4.44)–(4.45).

In the single-level integration algorithm, the electron diffusive flux Γ_{diff} is computed using a prediction $\hat{n}_e^{n+1}$ of the electron density obtained by solving (3.22). In solving this linear system, it is convenient to regard $\hat{n}_e^{n+1}kT_e^n$ as the dependent variable. We therefore require a boundary value for this quantity for the corresponding linear solves occurring during the fine grid integration (4.50)–(4.52). For this, we again employ linear interpolation in time and the high-order spatial interpolation operator Q, i.e.,

$$
\hat{n}_{e,f}^{m+1}kT_{e,f}^{m}=Q\left(\hat{n}_{e,f}^{m+1}kT_{e,f}^{m},\frac{m(n+1)-m}{n_{\text{cycle}}}n_{e,c}^{n}kT_{e,c}^{n+1}+\frac{m-m(n)}{n_{\text{cycle}}}n_{e,c}^{n+1}kT_{e,c}^{n+1}\right) \tag{4.78}
$$

for $m=m(n)+1,\ldots,m(n+1)$. Note that, as in the single-level integration, we do not correct the diffusive flux following the composite correction step (4.77).

Restoration of ion momentum conservation is handled similarly to (4.76). Specifically, momentum flux differences are accumulated on I during the coarse and fine level integrations, then used to correct the coarse level ion momenta. This correction is independent of all other composite synchronization steps.

The composite synchronization of the coarse and fine level integrated electron temperatures is performed in a manner similar to that in which the potential correction is performed.

After the integration (4.47)–(4.49) of the potential and electron density over $\Delta t = \Delta t_c$ on $\Omega_c \backslash P(\Omega_f)$, the temperature equation is also integrated to obtain

$$\frac{3}{2} n_{e,c}^{n+1} k T_{e,c}^{n+1} - \frac{3}{2} n_e^n k T_e^n + \Delta t \nabla \cdot \left(\frac{5}{2} \Gamma_{e,c}^{n+1/2} k T_{e,c}^{n+1/2} \right) = \Delta t \left[\nabla \cdot \left(\frac{5 \eta_c^n}{2} n_{e,c}^{n+1/2} k T_{e,c}^{n+1/2} \nabla k T_{e,c}^{n+1} \right) + P_{\text{net}}^n \right], \tag{4.79}$$

where

$$P_{\text{net}}^n \equiv -e \Gamma_e^n \cdot E^n + P_{\text{ind}}^n - \sum_j \frac{3 m_e}{m_j} k T_e^n \nu_{ej}^n n_e^n + \sum_j \epsilon_{ej} S_{ej}^n. \tag{4.80}$$

After the integration (4.50)–(4.52) of the potential and electron densities on Ω_f, the temperature equation is integrated to obtain

$$\frac{3}{2} n_{e,f}^{m(n+1)} k T_{e,f}^{m(n+1)} - \frac{3}{2} n_e^{m(n)} k T_e^{m(n)} + \sum_m \Delta t_f \nabla \cdot \left(\frac{5}{2} \Gamma_{e,f}^{m+1/2} k T_{e,f}^{m+1/2} \right) = \sum_m \Delta t_f \left[\nabla \cdot \left(\frac{5 \eta_f^m}{2} n_e^{m+1/2} k T_{e,f}^{m+1/2} \nabla k T_{e,f}^{m+1} \right) + P_{\text{net}}^m \right], \tag{4.81}$$

where

$$P_{\text{net}}^m \equiv -e \Gamma_{e,f}^m \cdot E_f^m + P_{\text{ind}}^m - \sum_j \frac{3 m_e}{m_j} k T_e^m \nu_{ej}^m n_{e,f}^m + \sum_j \epsilon_{ej} S_{ej}^m \tag{4.82}$$

with boundary values at I interpolated from the fine grid for $m = m(n) + 1, \ldots, m(n+1)$:

$$T_{e,f}^m = Q\left(T_{e,f}^m, \frac{m(n+1) - m}{n_{\text{cycle}}} T_{e,c}^n + \frac{m - m(n)}{n_{\text{cycle}}} T_{e,c}^{n+1} \right). \tag{4.83}$$

The potential and electron densities are then corrected by solving (4.68) and (4.72) and performing the updates (4.54) and (4.77). Let n_e^* and Γ_e^* denote the corrections made to the electron density and flux, respectively, during this step (e.g., $n_{e,c}^* = n_e^{n+1} - n_{e,c}^{n+1}$). We then seek a composite temperature correction $T_e^* = (T_{e,f}^*, T_{e,c}^*)$ such that

$$T_e^m \equiv \begin{cases} T_{e,f}^m + T_{e,f}^* & \text{on } \Omega_f \\ T_{e,c}^{n+1} + T_{e,c}^* & \text{on } \Omega_c \backslash P(\Omega_f) \end{cases} \qquad m = m(n) + 1, \ldots, m(n+1) \tag{4.84}$$

with

$$\begin{aligned} T_{e,f}^* &= Q(T_{e,f}^*, T_{e,c}^*) && \text{at } I \\ \nabla T_e^* &= 0 && \text{at the physical boundary} \end{aligned} \tag{4.85}$$

approximates the solution of the composite system integrated over the same fine grid time steps,

$$\frac{3}{2} n_e^{m(n+1)} k T_e^{m(n+1)} - \frac{3}{2} n_e^{m(n)} k T_e^{m(n)} + \sum_m \Delta t_f \nabla \cdot \left(\frac{5}{2} \Gamma_e^{m+1/2} k T_e^{m+1/2} \right) = \sum_m \Delta t_f \left[\nabla \cdot \left(\frac{5 \eta^m}{2} n_e^{m+1/2} k T_e^{m+1/2} \nabla k T_e^{m+1} \right) + P_{\text{net}}^m \right], \tag{4.86}$$

where

$$P^m_{\text{net}} \equiv -e\Gamma^m \cdot E^m + P^m_{\text{ind}} - \sum_j \frac{3m_e}{m_j} kT_e^m \nu^m_{ej} n_e^m + \sum_j \epsilon_{ej} S^m_{ej}. \tag{4.87}$$

After the fact that

$$\begin{aligned} n_e^{n+1}T_e^{n+1} - n_{e,c}^{n+1}T_{e,c}^{n+1} &= n_e^{n+1}T_e^* + n_e^{n+1}T_{e,c}^{n+1} - n_{e,c}^{n+1}T_{e,c}^{n+1} \\ &= n_e^{n+1}T_e^* + n_e^* T_{e,c}^{n+1}, \end{aligned} \tag{4.88}$$

is used and approximations similar to those used in computing the potential correction are made, subtraction of (4.79) from (4.86) on $\Omega_c \backslash P(\Omega_f)$ yields

$$\begin{aligned} &\frac{3}{2} n_e^{n+1} kT_{e,c}^* - \Delta t \nabla \cdot \left(\frac{5\eta^n}{2} n_e^{n+1/2} kT_e^{n+1/2} \nabla kT_{e,c}^* \right) \\ &\quad = -\frac{3}{2} n_e^* kT_{e,c}^{n+1} + \Delta t \nabla \cdot \left[\left\langle \frac{5\eta^n}{2} n_e^{n+1/2} kT_e^{n+1/2} \right\rangle_I \right. \\ &\qquad \left. \times \left(\langle \nabla kT_{e,f}^* \rangle_I - \nabla kT_{e,c}^* \right) - \frac{5}{2} \Gamma_e^* kT_e^{n+1/2} + \delta Q \right], \end{aligned} \tag{4.89}$$

where

$$\begin{aligned} \delta Q = &\left\langle \frac{1}{\Delta t_c} \sum_m \Delta t_f \left[\nabla \cdot \left(\frac{5}{2} \Gamma_{e,f}^{m+1/2} kT_{e,f}^{m+1/2} \right) - \frac{5\eta_f^m}{2} n_{e,f}^{m+1/2} kT_{e,f}^{m+1/2} \nabla kT_{e,f}^{m+1} \right] \right\rangle_I \\ &- \nabla \cdot \left(\frac{5}{2} \Gamma_{e,c}^{n+1/2} kT_{e,c}^{n+1/2} \right) + \frac{5\eta_c^n}{2} n_{e,c}^{n+1/2} kT_{e,c}^{n+1/2} \nabla kT_{e,c}^{n+1}. \end{aligned} \tag{4.90}$$

Furthermore, the subtraction of (4.81) from (4.86) on Ω_f yields

$$\begin{aligned} &\frac{3}{2} n_e^{m(n+1)} kT_{e,f}^* - \Delta t \nabla \cdot \left(\frac{5\eta^{m(n+1)-1}}{2} n_e^{m(n+1)-1/2} kT_e^{m(n+1)-1/2} \nabla kT_{e,f}^* \right) \\ &\quad = -\frac{3}{2} n_e^* kT_{e,f}^{m(n+1)} - \Delta t \nabla \cdot \left(\frac{5}{2} \Gamma_e^* kT_e^{m(n+1)-1/2} \right). \end{aligned} \tag{4.91}$$

The pair (4.89) and (4.91) together with (4.85) is therefore the composite system to be solved for the correction T_e^*.

4.3. Generalization to an Arbitrary Number of Refinement Levels

The algorithm just described for two grid levels can be generalized to an arbitrary number of levels. Let $\{\Omega_l, l = 0, \ldots, l_{\max}\}$ denote a refinement hierarchy with successive levels related as in the two-level case by

$$P(\Omega_l) \subset \Omega_{l-1}, \qquad l = 1, \ldots, l_{\max}, \tag{4.92}$$

where P is a projection operator. Each Ω_l is again the union of disjoint, rectangular grids obtained by refining rectangular subgrids of Ω_{l-1} by a factor $n_{\text{ref},l}$. For this general case, we

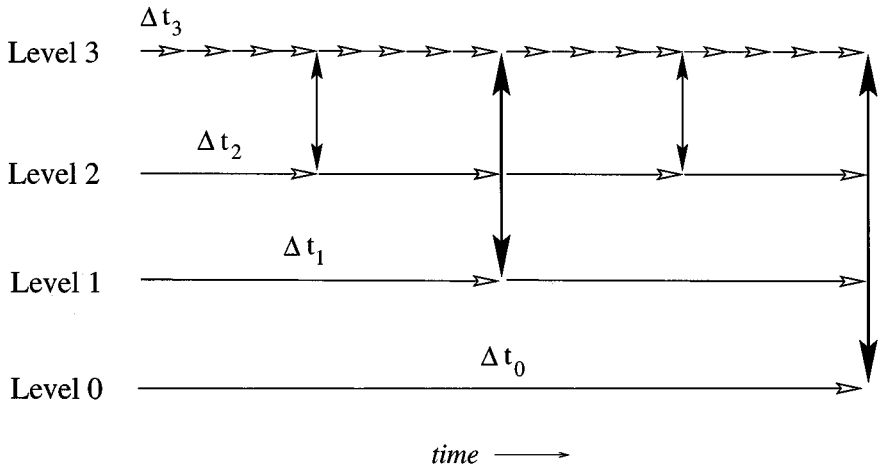

FIG. 4.3. Multilevel advance and composite synchronization schedule.

also require the *proper nesting* of levels. Specifically, we stipulate that $P(\Omega_l)$ be properly contained in Ω_{l-1} for $l = 1, \ldots, l_{\max}$, except perhaps at the physical domain boundary.

Our two-level algorithm consisted of the integration of the coarse grid equations over a time step, followed by n_{cycle} fine grid integrations over the same coarse time step and a composite synchronization of the two levels. For the general case, this can be implemented recursively, with the composite potential and temperature corrections occurring at the ends of time steps at each level (except the finest level) and involving all finer levels. Figure 4.3 schematically depicts the time step advance and composite synchronization schedule for a four-level problem with $n_{\text{ref},0} = n_{\text{ref},1} = 2$ and $n_{\text{ref},2} = 4$. The horizontal arrows denote single-level time advance steps, while the vertical arrows denote synchronization of the composite grid.

A perhaps subtle detail in our description of the two-level algorithm was the use of approximations such as

$$\frac{1}{\Delta t_c} \sum_m \Delta t_f n_e^{m+1/2} \mu^m \approx n_e^{m(n+1)-1/2} \mu^{m(n+1)-1}, \tag{4.93}$$

which replace the average of fine edge data over the coarse time step with the data at the last fine time step. If such approximations are not made, then in the case of more than two grid levels the composite synchronizations would require the similar accumulation and averaging of fine edge data on a given level l over *all* coarser time steps $\Delta t_{l'}$, $l' = 0, \ldots, l-1$, since level l will eventually participate in a composite synchronization with each of these coarser levels. Although it is possible to do this, we have determined empirically that approximations such as (4.93), which enable a totally recursive implementation, do not harm the accuracy of the method.

4.4. Solution of the Composite Systems

The synchronization of potential and electron temperature lead to the composite systems (4.68)–(4.72) and (4.89)–(4.91), respectively. In this subsection, we discuss the solution of these systems.

The system (4.68)–(4.72) is nonlinear due to the boundary condition (2.12) on the electron flux. We therefore employ a composite Newton iteration. In each Newton iteration, we must solve a composite Jacobian system with the composite nonlinear residual as the right-hand side, where on each refinement level Ω_l the Jacobian has the form of a second-order, symmetric, elliptic operator

$$L_l^{nf}(\phi_l) \equiv -\nabla \cdot (b_l \nabla \phi_l) + a_l \phi_l. \tag{4.94}$$

Since the level operators of the composite temperature system (4.89)–(4.91) are also of the form (4.94), it suffices to consider the solution of composite systems with operators L of this general form on each level. We generalize the approach described in [20] for Poisson's equation to these variable coefficient cases. This is a multilevel algorithm that iterates over a grid hierarchy $\{\Omega_l, l = l_{\text{base}}, \ldots, l_{\max}\}$ in a multigrid-like fashion, taking care to enforce the required matching conditions at the boundaries between successive levels. Letting f denote the composite system right-hand side and u denote the desired composite solution, the algorithm can be summarized as

$R := f - L(u)$.
While($|R| > \epsilon |f|$)
  $R := f - L(u)$.
  MGRelax($l_{\max}$)
EndWhile

Procedure MGRelax(l):

If ($l = l_{\max}$) then $R_l := f_l - L_l^{nf}(u_l, u_{l-1})$
If ($l > l_{\text{base}}$) then
  $u_{l,\text{save}} := u_l$
  $e_{l-1} := 0$
  $e_l := \mathit{Smooth}(e_l, R_l, h_l)$
  $u_l := u_l + e_l$
  $$R_{l-1} := \begin{cases} \left\langle \left(R_l - L_l^{nf}(e_l, e_{l-1}) \right) \right\rangle_{P(\Omega_l)} & \text{on } P(\Omega_l) \\ f_{l-1} - L_{l-1}(u_l) & \text{on } \Omega_{l-1} \backslash P(\Omega_l) \end{cases}$$
  MGRelax($l-1$)
  $e_l := e_l + \mathit{Interpolate}(e_l, e_{l-1})$
  $e_l := \mathit{Smooth}(e_l, R_l, h_l)$
  $u_l := u_{l,\text{save}} + e_l$
Else
  Smooth $L_{l_{\text{base}}}(e_{l_{\text{base}}}) = R_{l_{\text{base}}}$
  $u_{l_{\text{base}}} := u_{l_{\text{base}}} + e_{l_{\text{base}}}$
EndIf

Here, L_l denotes the level operator L_l^{nf} restricted to $\Omega_l \backslash P(\Omega_{l+1})$ and augmented with the appropriate refluxing terms to impose the level $l+1$ fluxes on I (the *nf* in the superscript of the operator defined by (4.94) means "no fine," implying that there is no dependence on fine grid information). For the *Smooth()* function, we use a single red–black Gauss–Seidel relaxation sweep, and for the *Interpolate()* step we use piecewise-constant interpolation, enforcing the boundary condition $e_l = Q(e_l, e_{l-1})$ at the interface between levels l and $l-1$.

Using this multilevel composite solution algorithm, we anticipate effective convergence rates typical of multigrid methods for elliptic problems. Our definition of "effective" in this

context is nominally an order of magnitude reduction in the residual for each cycle over the level hierarchy. The convergence of the algorithm depends significantly upon the behavior of the level operator coefficients a_l and b_l (4.94), and especially the relationship of these coefficients on $P(\Omega_{l+1})$ to a_{l+1} and b_{l+1}. If these coefficients are smoothly varying over the problem domain, a simple arithmetic averaging of a_{l+1} and b_{l+1} to obtain coarsened a_l and b_l on $P(\Omega_{l+1})$ is sufficient to obtain an effective convergence rate. This is indeed the case for the solution of the composite temperature synchronization system (4.89)–(4.91). However, in the composite Newton solution of the potential synchronization system (4.68)–(4.72), the linearization of the boundary condition (2.12) results in Jacobian coefficients a_l and b_l that vary by a few orders of magnitude at the physical boundary. For such coefficients, arithmetic averaging of the b_l coefficients fails to yield an adequate coarsening. If, instead, the coefficients b_l are harmonically averaged in the coordinate directions normal to their respective cell edges, we again obtain good convergence rates. We have also found it generally advantageous to "W-cycle" the multilevel algorithm, i.e., visit each refinement level twice before returning to the next finer level. This biases the computational work toward the coarser levels, which tends to improve both the robustness and the convergence rate of the iteration, although the work per cycle is increased.

The cost of the composite synchronization steps relative to the sum of the integration and sychronization times depends upon a number of factors. For problems like those for which results are reported in Section 5, this ratio is typically about 20–30% in our current software implementation.

5. RESULTS

The algorithm described above has been implemented in the Adaptive Plasma Model (APM) computer code. APM is a hybrid C++/FORTRAN code built upon an object oriented adaptive mesh refinement framework [10].

In this section we present numerical results that will help assess the utility of the previously described algorithm. Criteria such as accuracy, efficiency, and applicability to problems of practical interest are addressed. The accuracy of APM on locally refined grids is analyzed by comparisons to uniform grid calculations. The efficiency of the code is illustrated with a comparison of timings and memory requirements for refined grid and uniform grid calculations with the same peak resolution. The ability to attain very high resolutions is demonstrated with a simulation of a hydrogen plasma in which a 4 cm long volume is modeled with peak resolution of approximately 150 μm. This represents a ratio of length scales of over 250.

Though the previously described algorithm is not limited to a specific application, we have developed the algorithm with an eye toward modeling ICP reactors. Consequently, this section concludes with two simulations of practical interest for semiconductor manufacturing. First, the relationship between RF coil placement and the ion flux striking a silicon wafer is examined. Second, RF biasing of a wafer is examined for its effect on ion flux and energy distributions. Each of these studies shows the need for high resolution calculations.

5.1. Accuracy Using Locally Refined Grids

Consider a locally refined grid calculation with cell sizes h_l, $l = 0, \ldots, l_{\max}$, where $h_{l_{\max}}$ is the cell size corresponding to the finest cells and h_0 the cell size corresponding to the

coarsest cells. Such a calculation cannot be expected to be more accurate than a uniform grid calculation with cell size $h_{l_{\max}}$. In fact, the degree to which the locally refined calculation matches the fine, uniform grid calculation is a measure of its accuracy.

Figure 5.4 shows a comparison of potential, ion flux, electron density, and electron temperature profiles using a locally refined grid, a fine uniform grid, and a coarse uniform grid. Symmetry boundary conditions on top and bottom of the rectangular domain restrict the solution variation to the x-direction. The uniform fine grid spacing corresponds to the finest cells in the refined grid and the uniform coarse grid spacing corresponds to the coarsest cells. The refined grid uses three levels. Each level was generated by refining the leftmost three cells of the next coarser level. This grid structure is shown in Fig. 5.5. The agreement in the fine portion of the locally refined grid is excellent while the agreement in the coarse region is much better than that in the equivalent coarse grid.

The accuracy of the locally refined calculation depends critically on where the refined grids are used. The high accuracy of this calculation is due in large part to the placement of the fine grids near the boundary. As expected this region has the most variation in potential, density, and charge. Consequently, smaller cells are required to resolve this variation. Conversely, the use of coarse grids has little effect on the quasineutral region in the center of the plasma. In this case the correct placement for the grids was clear, but in a more complex calculation this may not be the case. We are continuing to investigate the relationship between grid placement and accuracy. Most of our work to date has used heuristically placed grids, but our code is capable of refining grids based on values of the state variables or their gradients.

5.2. Computational Efficiency

In order for a local refinement strategy to make sense, refined calculations need to show some computational savings. Figure 5.6a shows the relative CPU times for uniform grid calculations compared to the locally refined calculations. The model problem for these timings was an 8 cm square grid with symmetry boundaries on the top and right side and grounded conductors on the other sides. At each level, the outermost two cells along the physical boundary were chosen for refinement. One of these refined grids (corresponding to four levels refined by a factor of 2) is shown in Fig. 5.7.

Results are shown for refinement ratios of 2 and 4. In each case calculations were performed with a refined grid and a uniform grid with the same peak resolution. The uniform grid cell size corresponded to the finest cells in the refined grid. Each pair of calculations was run to the same simulation time and the ratio of CPU times was calculated. Because of prohibitive runtimes for the higher resolution cases (uniform or locally refined), the final time for each pair of calculations was the same but different pairs used different final times. Our experience has been that the CPU time per iteration does not vary much during a calculation, so these timings should be representative.

Figure 5.6a shows that speedup is a strong function of refinement ratio. For the model problem, four refined levels were required to obtain an improvement over the single fine grid case using a refinement ratio of 2. When a refinement ratio of 4 was used, only two refined levels were required to obtain a significant speedup. These trends depend on a number of factors. Chief among these are the time spent performing composite synchronizations and the fraction of the domain covered by fine cells. For this model problem, the number of cells at the fine levels was rather large compared to the number at the coarse levels. This is required if one wants to refine along an entire edge. If, for example, a corner

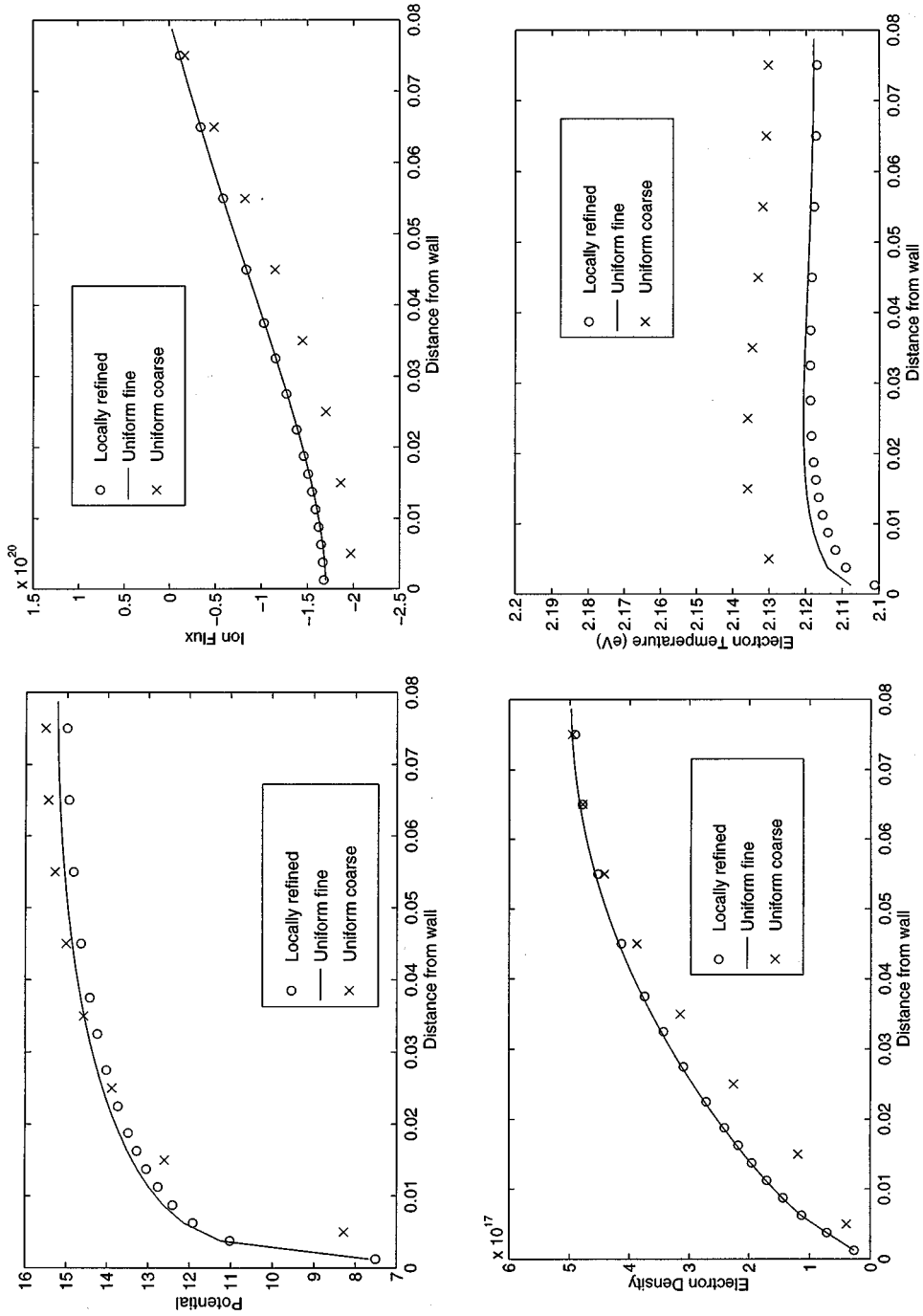

FIG. 5.4. Potential, ion flux, electron density, and temperature profiles for locally refined, fine, and coarse grid calculations.

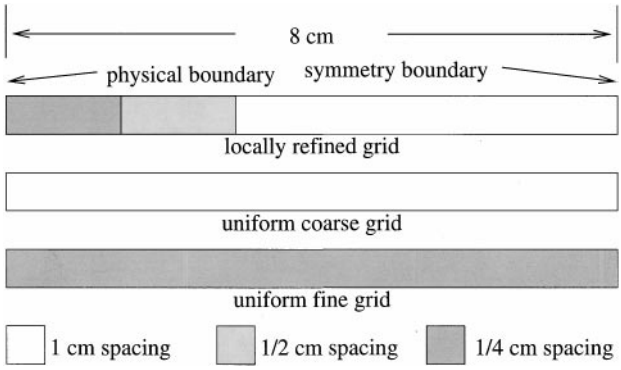

FIG. 5.5. Locally refined, uniform fine and uniform coarse grids used to generate results shown in Fig. 5.4.

were recursively refined we would expect an even larger speedup using the locally refined grid.

The dependence of time spent performing composite synchronizations on the grid hierarchy is hard to quantify, but is typically about 20–30% of the total simulation time. We do know that the number of composite solves grows quickly with the number of levels. If m refined levels are used and each is refined by a factor of n in time and space, this will require n^{m-1} composite solves per coarse time step. This fact and the speedup results shown in Fig. 5.6a both show the need for higher refinement ratios.

Decreased runtime is not the only benefit of locally refined grids. Figure 5.6b shows the relative reduction in memory requirements for the locally refined calculation. The memory requirements are much less severe for refined grids, provided that the finest levels do not make up a large fraction of the domain. At some level of refinement, it is no longer feasible to compare highly refined grids to uniform equivalent grids because the latter cannot fit within physical memory. This fact limited the number of refined levels we could use in these comparisons.

The finest of the calculations shown in Fig. 5.6 cannot be run to steady state in a reasonable amount of time. Even with the speedup shown with local refinement, very high resolution calculations are still very expensive [9]. Significant savings can be made by gradually refining the grid as the solution approaches steady state. This approach was used for the remaining calculations described in this paper.

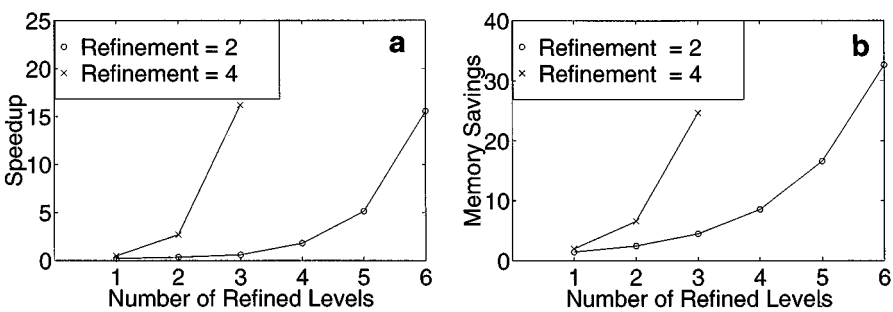

FIG. 5.6. Ratio of CPU times and memory requirements for uniform and locally refined grid calculations.

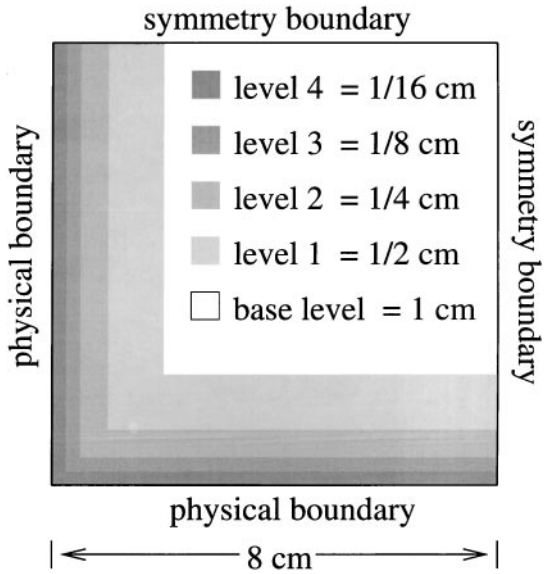

FIG. 5.7. Sample grid used to calculate time and storage savings shown in Fig. 5.6. This grid corresponds to the case with refinement ratios of 2 and 4 refined levels.

5.3. Sheath-Scale Hydrogen Plasma Simulation

One of the primary motivations for using locally refined grids is to allow for the simulation of features with disparate length scales. In the next example, we illustrate this by modeling a hydrogen plasma with sheath-scale resolution in a domain that is a few hundred times larger. Specifically, we take advantage of symmetry and model an 8 cm region with a 4 cm long volume and a symmetry boundary condition. In addition to the symmetry boundary in the x-direction, we enforce symmetry in the transverse direction. This forces the variation in the solution to one direction, making it a one-dimensional problem. We employ two refined levels, each with a refinement ratio of 4. Each refined level covers the exterior $1/4$ of the next coarser level. This grid structure is shown in Fig. 5.8. With these refinements, the finest grids are 156 μm, which turns out to be approximately the Debye length. At this resolution, we have a grid spacing that is smaller than the plasma sheath width.

In order to verify that our algorithm properly calculates the plasma behavior in the sheath, within the limits of our fluid model, we compared the APM results to the results of the sheath model of Ingold [15]. For the comparison, we took APM calculated values for ion and electron density, flux, and electric field at a point outside of the sheath region and used these data as an initial condition for the Ingold model. The model was then integrated to the bounding wall using the CVODE [6, 7] ordinary differential equation solver. These results

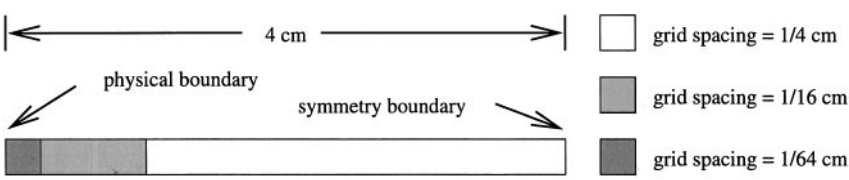

FIG. 5.8. Grid structure for hydrogen sheath calculation.

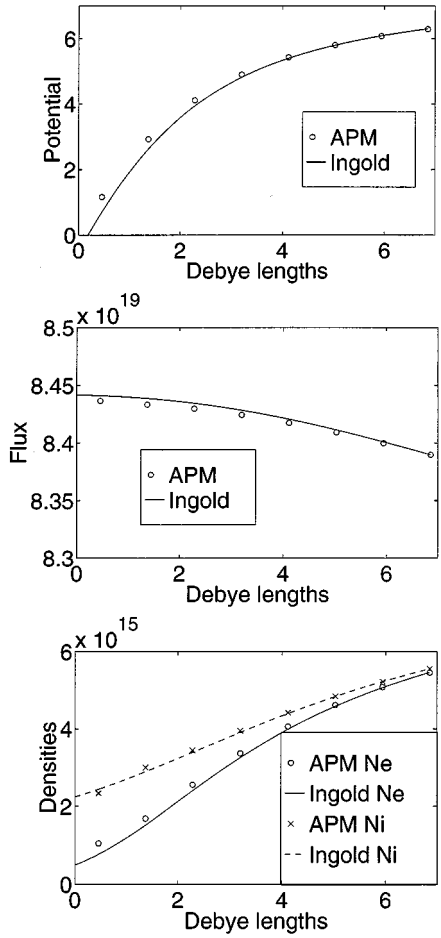

FIG. 5.9. Comparison of APM and Ingold sheath model for potential, flux, and densities, respectively.

were then compared with the APM results in the sheath region and are shown in Fig. 5.9. The agreement is excellent. The small variations may be attributable to the omission of electron inertia from APM, different treatment of the boundary conditions, or discretization effects. The assumption of constant electron temperature and constant collisional parameters in the Ingold model was not found to be an important difference from APM in this experiment. This was determined by varying the temperatures and collisional parameters in the Ingold model within the range used in the APM calculation.

5.4. Application to Inductively Coupled Plasma Reactor Simulation

We next consider the application of our method to two problems of interest for ICP reactor-based plasma processing. Both of these examples demonstrate the importance of using sufficient resolution, which can be facilitated with local grid refinement. A generic ICP reactor is depicted in Fig. 5.10. The first simulations examine the effects of different power deposition profiles on the uniformity of ion flux at a wafer surface. These profiles correspond to specific RF coil positions and hence address the important engineering issue of optimal coil placement. The second set of simulations considers RF biasing of the wafer surface, in which we examine the effect on ion particle and energy flux.

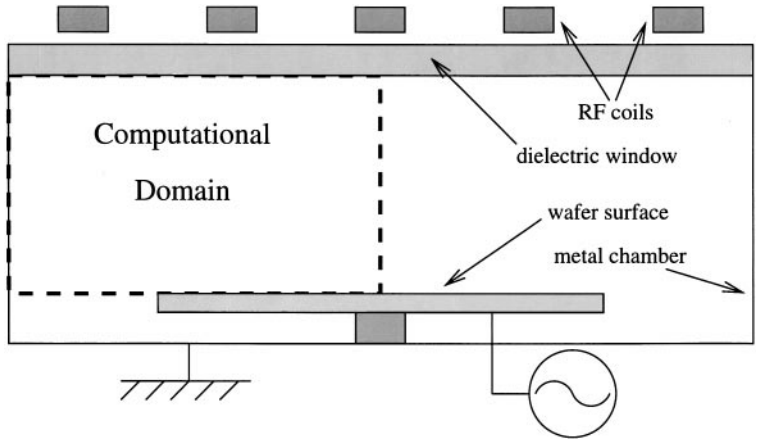

FIG. 5.10. Sample inductively coupled reactor and the domain on which the simulations occur.

5.4.1. *Power Deposition Profile Effects*

Along with selectivity and etch rate, spatial etch uniformity is a critical process parameter. A uniform ion flux will help produce a uniform etch. One of the controls over the ion flux is the power deposition within the plasma. The input power drives up the electron temperature and ionization rate. This results in the production of ions which ultimately bombard the wafer or the chamber walls.

The dimensions and grid structure used in these simulations are shown in Fig. 5.11. In these simulations, the time-averaged RF power profile is based on the location of driving coils and an assumed exponential falloff for the time-averaged power within the plasma. The model does not resolve the inductive fields, nor are the effects of density variation on the power profile in the plasma considered.

Figure 5.12 shows the relation between the assumed reactor coil positions and the resulting power deposition profiles. In each case, the coil power was assumed to fall off with a 1 cm skin depth normalized to 1000 W total power. The resulting ion fluxes at the wafer surface were compared for these five cases using both a relatively coarse uniform grid and a grid with local refinement near the wafer surface. The uniformity of the normal ion flux at the

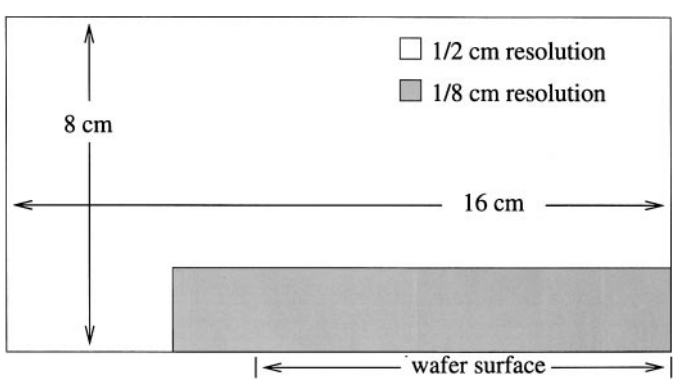

FIG. 5.11. Computational domain for the locally refined calculations used to study power deposition effects.

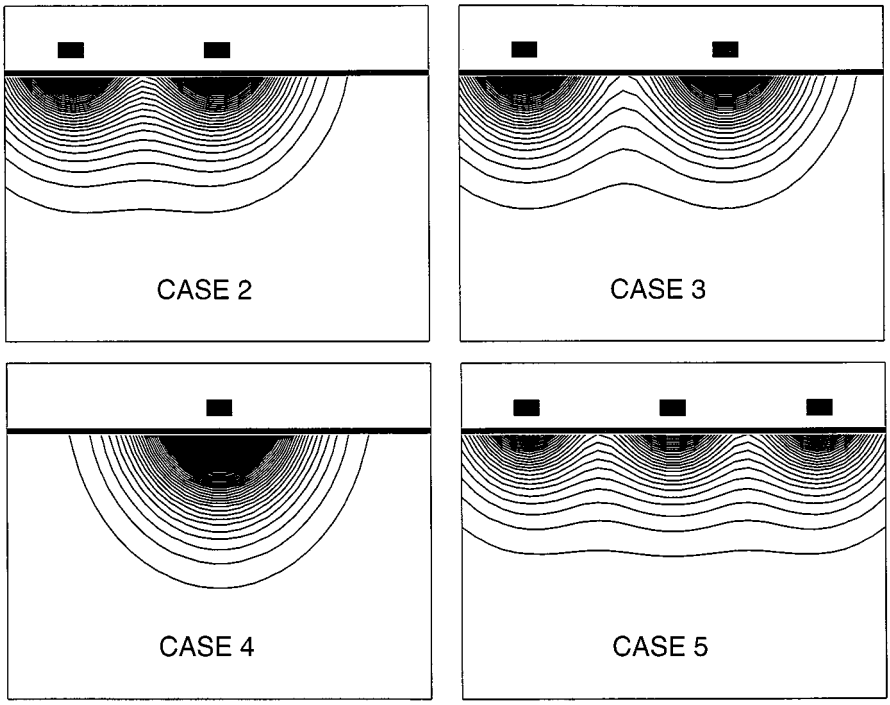

FIG. 5.12. Coil positions and power density contours for power deposition study (uniform power density case is not shown).

wafer surface was measured and is shown for all five cases in Fig. 5.14. The different cases were compared using the figure of merit

$$\text{uniformity} = \frac{\text{max flux} - \text{min flux}}{\text{average flux}}. \tag{5.95}$$

It is interesting to compare the results for two of these cases, labeled 2 and 3. Using only the coarse grid solutions, case 3 appears to be a far superior coil configuration. Subsequent refined calculations, however, show that cases 2 and 3 were very similar and that 2 was in fact superior. This is shown in Fig. 5.13. Although total ion flux is not sensitive to grid spacing, this simulation suggests that the details of the spatial distribution are.

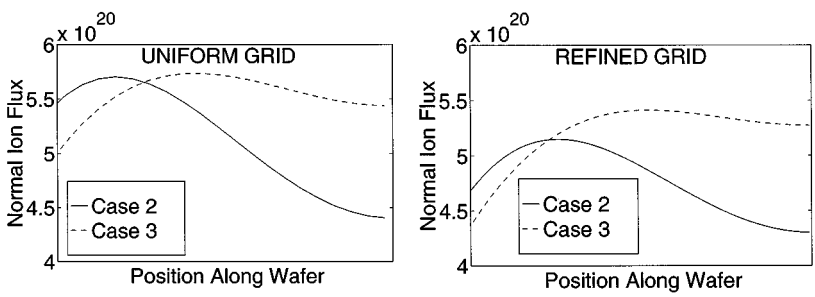

FIG. 5.13. Coarse, uniform solution for coil placements 2 and 3.

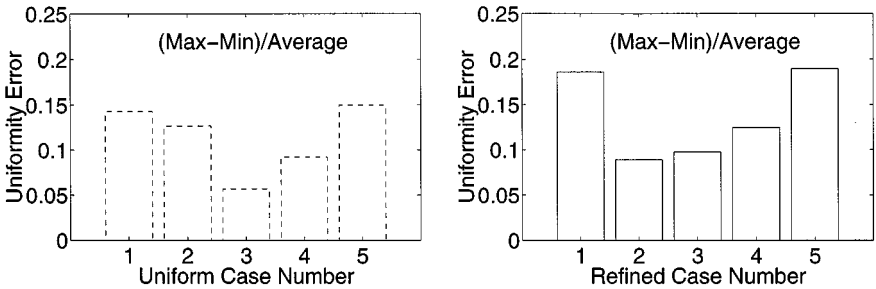

FIG. 5.14. Wafer flux uniformity error for refined and uniform grids.

5.4.2. *Wafer Biasing Effects*

One of the key advantages to inductively coupled plasma reactors is the ability to independently control ion flux and energy. Capacitively coupled plasmas have an electron production rate dependent on the driving potential. High potential gradients required for high densities often result in bombarding of the wafer surface by very high energy ions, which is usually undesirable. In many ICP reactors, the chuck (or platten) holding the semiconductor wafer has an RF potential bias applied to it. This has the effect of increasing the average ion energy at the wafer for the same total ion flux. The driving RF coils dump energy into the plasma, but the resulting fields are not seen by the wafer, which is many skin depths away. Similarly, the bias field at the wafer does not affect the bulk plasma. Surface plots of the state variables for one of these calculations are shown in Fig. 5.15.

In order to quantify these effects and determine their dependence on grid resolution, APM was used to study the effect of RF wafer biasing on the ion flux and energy impinging on the wafer surface. The geometry and grid structure for these calculations are shown in

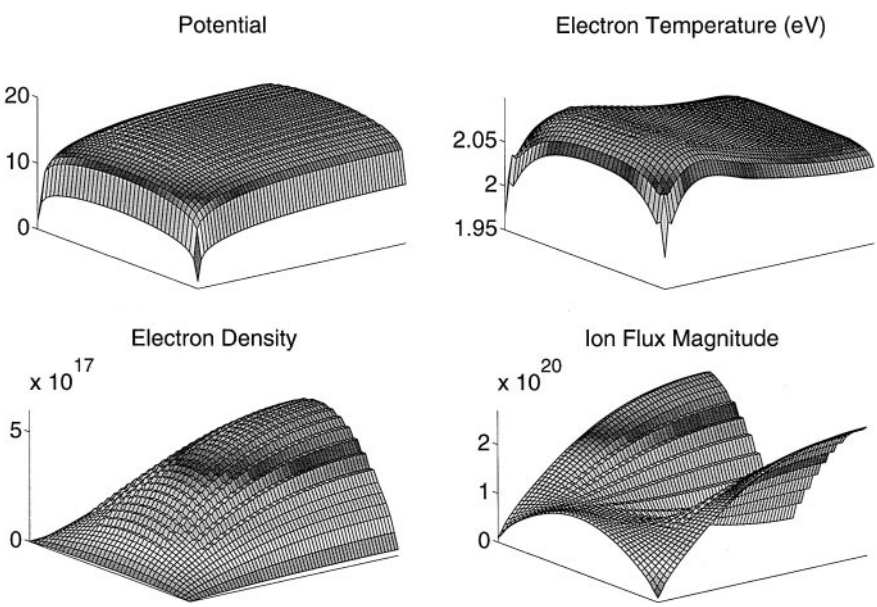

FIG. 5.15. Potential, electron density, electron temperature, and ion flux magnitude plots for one of the wafer bias calculations.

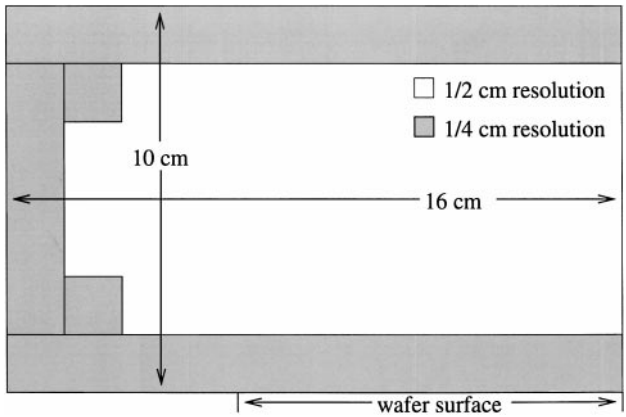

FIG. 5.16. Computational domain used in the locally refined calculations as part of the wafer bias study.

Fig. 5.16. Three calculations were performed corresponding to 13.56 MHz bias potentials of 0, 10, and 20 V (peak). Each of these was calculated on a relatively coarse, uniform grid and a locally refined grid. The refined grid included regions near the domain boundaries, both those representing the wafer surface and the chamber walls. Figure 5.17 shows the calculated ion flux and energy, respectively, at the wafer surface.

As expected, the ion flux was insensitive to the bias potential as well as grid resolution effects. The effect of the bias was less than 2.1% and that of the grid resolution was less than 3.4%. Note how this contrasts with the previously discussed power deposition study. The ion energy changed significantly at high bias potentials, however. The 20 V bias resulted in

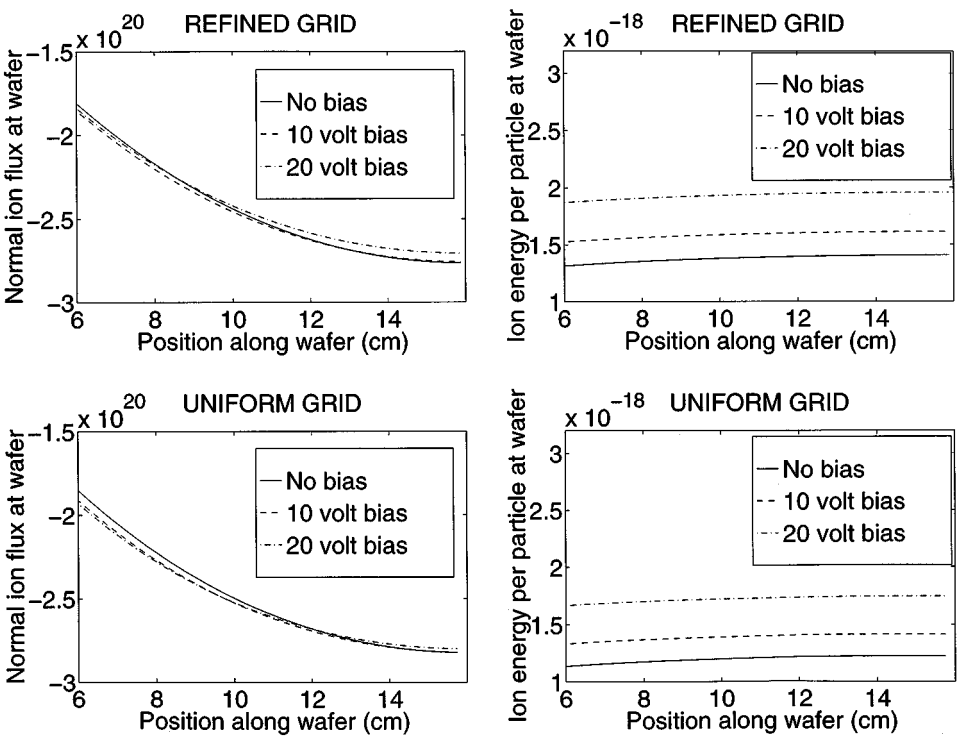

FIG. 5.17. Ion normal flux and energy per particle for uniform and locally refined grids.

higher ion energies of up to 61.4%. The higher resolution calculations differed significantly from the corresponding uniform calculations as well. These grid resolution effects accounted for up to 17.0% variations in the ion energy. This suggests that an accurate computational study of RF biasing effects requires high spatial accuracy in the wafer region to properly resolve ion energy profiles.

6. CONCLUSION

We have described a numerical method for the solution of plasma fluid equations on block-structured, locally refined grids. The accuracy and efficiency of the algorithm for some representative problems have been demonstrated. We have also presented calculations that address questions of interest for semiconductor processing.

The algorithm presented in this paper assumes fixed, locally refined grids. However, nothing in the algorithm precludes adaptive changes to the grid structure. In fact, the code used to obtain the results presented in Section 5 can adaptively regrid based on user-specified criteria. The selection of such criteria has been investigated in [32]; however, more study is warranted. In particular, the use of error estimation for grid selection should be investigated.

In spite of the large computational savings achieved with locally refined grids, high-resolution steady-state calculations can still be extremely expensive using our time-dependent algorithm. Consequently, we have investigated some modifications that allow the use of larger time steps. These include the reduction to first order of the electron density edge predictions and the use of more aggressive subcycling ($n_{\text{cycle}} > n_{\text{ref}}$) [32]). A fully implicit integration of the coupled electron continuity and Poisson equations is also being investigated. The accuracy and efficiency of such alternatives, relative to the algorithm presented here, remain to be determined, however.

REFERENCES

1. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, *J. Comput. Phys.* **142**, 1 (1998).
2. M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* **82**, No. 1, 64 (1989).
3. M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* **53**, 484 (1984).
4. 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** (September 1996).
5. J. D. Bukowski, R. A. Stewart, D. B. Graves, and P. Vitello, Modeling inductively coupled plasma tools with chlorine chemistry, in *Proceedings, Electrochemical Society Meeting, May 1994.*
6. S. D. Cohen and A. C. Hindmarsh, *CVODE User Guide*, Technical Report UCRL-MA-118618, Lawrence Livermore National Laboratory, September 1994.
7. S. D. Cohen and A. C. Hindmarsh, CVODE, a stiff/nonstiff ODE solver in C, *Comput. Phys.* **10**, No. 2, 138 (1996).
8. P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *J. Comput. Phys.* **87**, 171 (1990).
9. 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).
10. W. Y. Crutchfield and M. L. Welcome, *Object Oriented Implementation of Adaptive Mesh Refinement Algorithms*, Technical Report UCRL-JC-113502, Lawrence Livermore National Laboratory, Livermore, CA, April 1993.

11. G. W. Hedstrom, G. H. Rodrigue, M. Berger, and J. Oliger, Adaptive mesh refinement for 1-dimensional gas dynamics, in *IMACS* (North-Holland, Amsterdam, 1983), p. 43.
12. R. J. Hoekstra and M. J. Kushner, The effect of subwafer dielectrics on plasma properties in plasma etching reactors, *J. Appl. Phys*. **77**, No. 8, 3668 (1995).
13. R. J. Hoekstra and M. J. Kushner, Predictions of ion energy distributions and radical fluxes in radio frequency biased inductively coupled plasma etching reactors, *J. Appl. Phys*. **79**, No. 5, 2275 (1996).
14. L. H. Howell, R. B. Pember, P. Colella, J. P. Jessee, and W. A. Fiveland, A conservative adaptive-mesh algorithm for unsteady, combined-mode heat transfer using the discrete ordinates method, *Numer. Heat Transfer, Part B*, in press.
15. J. H. Ingold, Two-fluid theory of the positive column of a gas discharge, *Phys. Fluids* **15**, No. 1, 75 (January 1972).
16. A. E. Koniges, G. G. Craddock, D. D. Schnack, and H. R. Strauss (Eds.), *Proceedings, the Workshop on Adaptive Grid Methods for Fusion Plasmas*, Lawrence Livermore National Laboratory, Technical Report CONF-941279, July 1995.
17. 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**, No. 3, 1337 (1996).
18. M. A. Lieberman and A. J. Lichtenberg, *Principles of Plasma Discharges and Materials Processing* (Wiley, New York, 1994).
19. D. M. Manos and D. L. Flamm, *Plasma Etching: An Introduction* (Academic Press, San Diego, 1989).
20. D. F. Martin, *An Adaptive Cell-Centered Projection Method for the Incompressible Euler Equations*, Ph.D. thesis, University of California at Berkeley, December 1998.
21. S. F. McCormick, *MultiLevel Adaptive Methods for Partial Differential Equations*, Frontiers in Applied Mathematics, Vol. 6 (SIAM, Philadelphia, 1989).
22. R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jesse, An adaptive projection method for the modeling of unsteady, low-mach number combustion, in *Proceedings, Fall Meeting of the Western States Section of the Combustion Institute, 1997*.
23. I. Peres and M. J. Kushner, Spatial distributions of power and ion densities in rf excited remote plasma reactors, *Plasma Sources Sci. Technol*. **5**, 499 (1996).
24. T. J. Sommerer and M. J. Kushner, Monte Carlo-fluid model of chlorine atom production in Cl_2, HCl, and CCl_4 radio-frequency discharges for plasma etching, *J. Vacuum Sci. Technol. B* **10**, No. 5, 2179 (1992).
25. O. Steiner, M. Knolker, and M. Schussler, *Solar Surface Magnetism* (Kluwer Academic, Dordrecht/Norwell, MA, 1994), p. 441.
26. R. A. Stewart, P. Vitello, and D. B. Graves, Two-dimensional fluid model of high density inductively coupled plasma sources, *J. Vacuum Sci. Technol. B* **12** (January 1994).
27. R. A. Stewart, P. Vitello, D. B. Graves, E. F. Jaeger, and L. A. Berry, Plasma uniformity in high-density inductively coupled plasma tools, *Plasma Sources Sci. Technol*. **4** (1995).
28. W. Tan, R. J. Hoekstra, and M. J. Kushner, A time dependent propagator method for long mean free path transport of neutral particles in plasma processing reactors, *J. Appl. Phys*. **79**, No. 7, 3423 (1996).
29. P. L. G. Ventzek, M. Grapperhaus, and M. J. Kushner, Investigation of electron source and ion flux uniformity in high plasma density inductively coupled etching tools using two-dimensional modeling, *J. Vacuum Sci. Technol. B* **12**, No. 6, 3118 (1994).
30. P. L. G. Ventzek, R. J, Hoekstra, and M. J. Kushner, Two-dimensional modeling of high density inductively coupled sources for materials processing, *J. Vacuum Sci. Technol. B* **12**, No. 1, 461 (1994).
31. P. Wainman, R. A. Stewart, M. A. Lieberman, D. B. Graves, and P. Vitello, Comparison of langmuir probe characterization and model predictions in a high density ICP source, *Bull. Am. Phys. Soc*. **39**, No. 6 (1994).
32. D. D. Wake, *Simulation of Plasma Based Semiconductor Manufacturing Using Block Structured Locally Refined Grids*, Ph.D. thesis, University of California at Davis, 1998.
33. H. Wu, B. W. Yu, M. L. Li, and Y. Yang, Two-dimensional fluid model simulation of bell jar top inductively coupled plasma, *IEEE Trans. Plasma Sci*. **25**, No. 1, 1 (February 1997).