JOURNAL OF COMPUTATIONAL PHYSICS **138,** 585–616 (1997)
ARTICLE NO. CP975835

The Embedded Curved Boundary Method for Orthogonal Simulation Meshes

Dennis W. Hewett

Institute for Scientific Computing Research, Lawrence Livermore National Laboratory, Livermore, California 94550

Received December 11, 1996; revised June 11, 1997

A method for incorporating piecewise-linear approximations to curved boundaries is developed for codes using orthogonal meshes. The resulting method retains much of the fidelity of adaptive triangle meshes in 2D while retaining essentially all of the speed of traditional methods relying on banded matrix solutions. We find this technique useful for situations lying between the crude but fast stairstep approximation and the elegant but slower adaptive mesh techniques. This method provides a practical approach to problems such as space charge emission while retaining the simplicity and run time advantage of orthogonal-mesh-based codes. Examples from space-charge-limited flow between curved electrodes are given.

I. INTRODUCTION

The computational representation of curved structure boundaries have become remarkably effective in recent years [1–4]. Adaptive, unstructured finite element mesh systems allow the user the flexibility to place mesh points in precisely those locations that are necessary to achieve fidelity in the computational representation. In many of these systems, these distributed mesh points that represent the structure can move with the structure itself so that, in a Lagrangian sense, the representation follows the motion of the structure. The methods that have evolved from these ideas now provide unprecedented accuracy in spatial resolution.

In contrast to the generality of unstructured finite elements, simple structured meshes offer simple, fast, and agreeable data structures. For example, finding the mesh indices surrounding a point in space is a straightforward process with a structured mesh, compared to the considerably harder problem in an unstructured mesh. Codes based on structured meshes are typically easier to understand and

0021-9991/97 $25.00

lead to well-posed elliptic and parabolic problems that are comparatively easy to solve. Generally, structured mesh representations lead to banded matrices while unstructured representations lead to more complicated sparse matrices. We use here a simple orthogonal mesh, the reward being surplus free energy to be used on more ambitious physics models. Banded matrices also lend themselves to our existing *alternating direction implicit* suite of field solutions [5, 6]. Recent advances in adapting these methods to massively parallel computers reinforce our opinion that iterative field solutions utilizing banded matrix methods will continue to be competitive [7].

Orthogonal mesh simulations, however, cannot claim accurate solutions in the vicinity of a curved boundary represented by a ''stairstep'' boundary; when important physics occurs here, this approximation is clearly inadequate. Our motivation is shown in Fig. 1a, where field emission from such internal stairstep boundaries is unacceptable, even though the physics for other fast moving particles a few cells distant from boundaries is often tolerable. Methods that represent internal boundaries using grid-based intrinsic material properties [8] still need many grid points to avoid blurring the interface. Figure 1b gives a preview of results of obtained with the *embedded curved boundary* (ECB) method we describe here.

Peskin [9] developed a similar concept to study blood flow in hearts. His concern was to develop a representation for heart muscle and valves that could be ''embedded'' within a mesh to deflect and thus receive forces from moving fluids. Although similar in appearance to ECB, Peskin's ''boundaries'' were actually strongly interconnected Lagrangian points that were themselves governed by the Navier–Stokes equations. Sulsky and Brackbill [10] have extended these ideas to the study of suspension flow in which deformable boundaries are used to represent the shapes of the particles in suspension. Similarly, Mayo and Greenbaum [11] cast these concepts into a combination of finite-difference and integral equations. Their methods incorporate a Green's function representation for the boundary contribution that was apparently solved simultaneously with the finite difference equations. Merging these techniques requires the solution of a dense, nonsymmetric matrix that we believe requires more effort than our ECB method (which retains a simple banded matrix). Pointon [12] uses a method more similar to ours. His algorithm represents curved boundaries by requiring that these boundaries be coincident with the orthogonal mesh points for vertical and horizontal sections or by joining diagonal corners of his mesh cells. These limitations are a result of his careful treatment of the particle-in-cell (PIC) current accumulation with a charge-conserving algorithm that will become unimportant with our space-charge-emission algorithm.

We offer here a method with more geometric generality than that of Pointon, with an emphasis on the gradients and curls of electromagnetic potentials rather than the fluid equations of Peskin. We believe that our method produces a matrix that is easier to solve than that of Mayo and Greenbaum. Our ECB method recovers much of the fidelity of unstructured meshes while retaining most of the speed and ''code-friendly'' characteristics of orthogonal meshes. We emphasize the straightforward ability to incorporate our method into the fast solution of elliptic equations, by building ''better,'' but not more, coefficients in the vicinity of these boundaries. Our method alleviates PIC accumulation concerns with space-charge emission by

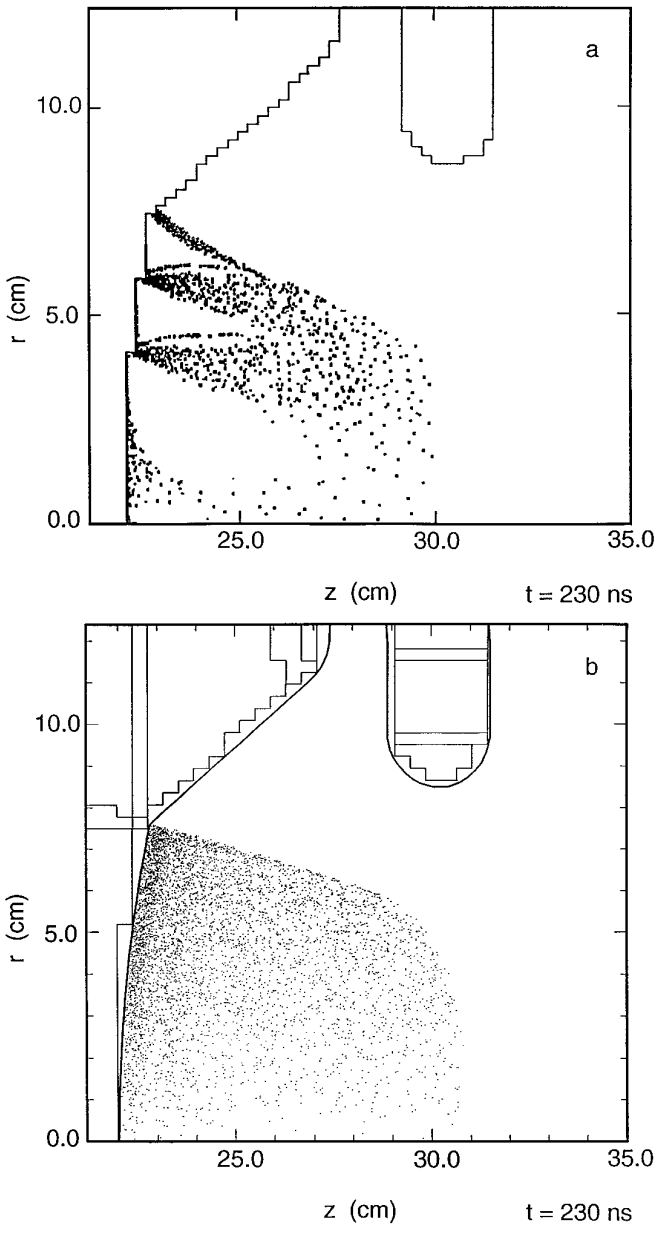

FIG. 1. (a) is a typical, unacceptable simulation of a space-charge-limited ion source using a stairstep representation of the electrode boundaries. This served as motivation for the ECB techniques that provide the results in (b) with an even coarser mesh.

integrating a simple boundary condition that specifies a potential for the emitting surface and emits enough charge to force the normal electric field to zero. In the following sections we present our concepts for embedding curved boundaries in the orthogonal mesh, the data structure surrounding the concepts, and some examples that elucidate its capabilities.

II. THE ECB CONCEPT

The data structure for our orthogonal-grid-based physics codes is already well developed and permeates several very useful codes. Our task was to add a curved boundary capability to this existing data structure. The ECB concept is defined in the context of the information that an elliptic solution technique needs to construct a solution in the presence of these boundaries. The essential features of ECB are:

(a) the data structure

(b) the elliptic solution technique

(c) the finite difference approximations to find the grad/curl.

We will discuss the features in order. First a caveat: the organization about to be described provides the required information; it is sufficient but not unique. While there are many functioning alternatives that may be more convenient in other situations, we have optimized for our applications.

The structures themselves are constructed out of the superposition of analytically prescribed building blocks. We present examples constructed from a POLY4 block, which is simply the axisymmetric volume generated by the rotation of a four-sided polygon defined by four points in the r,z plane. Another building block is called an ANNULUS, which is the axisymmetric toroid with axis at some point r_{cen}, z_{cen} in the r,z plane, an inner poloidal radius (which may be zero), an outer radius (which may be "large"), and poloidal starting and stopping angles. An example of electrodes constructed out of these elements is shown in Fig. 1b. Other structures such as ellipses have been developed. The various descriptors of the numerical boundaries are generated from straightforward coding that determines and assigns indices to those points just inside and just outside structure boundaries. In the ECB scheme, we first determine the relation of each mesh point I, J relative to the boundary structures. In Fig. 2a, for example, point I = 3, J = 3 is inside the structure boundary and point I = 3, J = 2 is just outside the structure. In this case, there is an intersection of the actual structure boundary with the I = 3 line between J = 2 and J = 3. Each intercept of the structure boundary with an I or J grid line is assigned an index. We store the actual r,z value at the intercept, and the slope of the boundary at that intercept, in arrays whose index K is associated with, in this case, the mesh point I = 3, J = 3—that mesh point just inside the structure. Similarly, in Fig. 2b the intercepts and slopes of the boundary between interior point I = 3, J = 2 and the exterior points (I = 4, J = 2), (I = 3, J = 1), and (I = 2, J = 2) are all associated with point I = 3, J = 2.

We now construct piecewise linear segments (PWLS) that will serve as the computational boundary. First we tabulate each intercept of the superposition of analytically-given boundaries with grid lines. We also store the slope of the analytic boundary at each intercept. Each intercept and slope determines a PWLS. The intersection of these PWLSs with other nearby PWLSs defines the endpoints. We insist that there be an endpoint for each sequence of PWLSs coming through a cell. (In the case of collinear segments, the endpoints are specified to be halfway between grid lines. For a few atypical cases where the intersection is not within

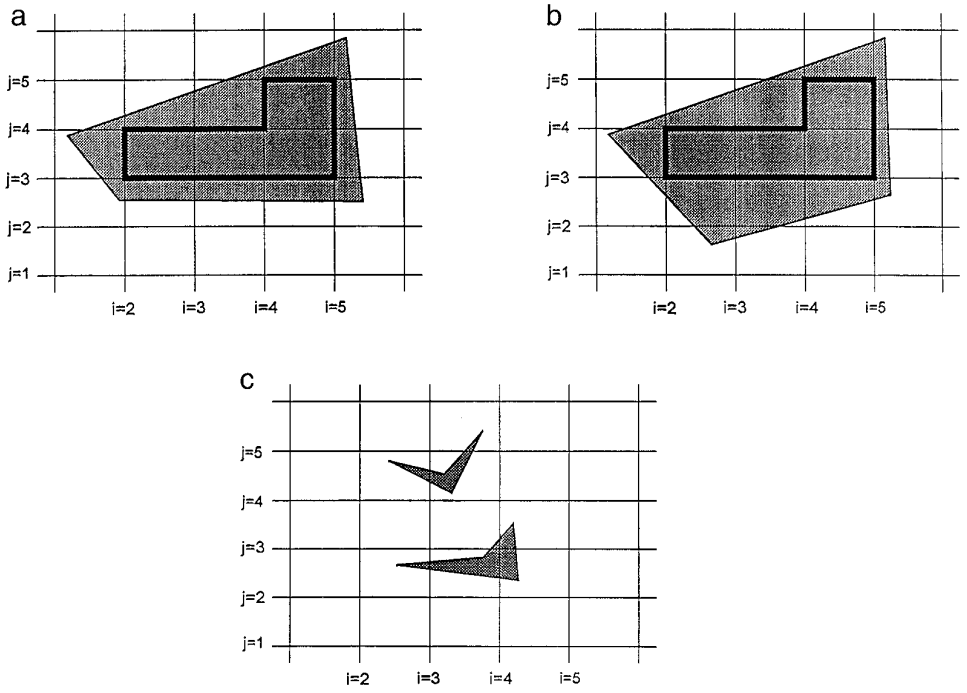

FIG. 2. A simple four-sided polygon structure embedded in an orthogonal mesh: (a) shows the most basic structure used to display those points just inside and just outside of a structure; (b) shows a slight modification to (a) as an example of an inside point that has more than one point just outside the mesh point, and (c) shows points just "inside" that are, in fact, outside the opposite wall of the thin structure. The dark lines on the interior of the structures in (a) and (b) are the stairstep structures that would represent these structures without ECB.

the cell, the node point is pulled back to the nearest cell boundary.) We give directionality to the PWLS; for each segment K we store the endpoints R1(K), Z1(K) and R2(K), Z2(K) for the intercept associated with index K. Directionality is determined by the convention that looking from point 1 to point 2 will place the interior of the structure on the left. Note that point 2 for some index K is a point 1 for some nearby intercept associated with another index K′.

We have already given an example of a point (I = 3, J = 2) in Fig. 2b which has more than one intercept associated with it. Point I = 5, J = 5 is another example with two such intercepts. There may be up to four intercepts associated with any given mesh point I, J. To keep track of where this intercept information is stored, we have defined an array KEYC at each I, J as

KEYC(I, J) = −1 for mesh point inside a structure
KEYC(I, J) = *klmn* for mesh points just outside a structure
KEYC(I, J) = 0 for all other points outside a structure,

where *klmn* denotes the direction(s) from point I, J at which boundary information can be found. We use 1s and 0s for each of the four digits denoted by *klmn*. The

digits k, l, m, and n denote the I − 1, J − 1, I + 1, and J + 1 directions, respectively. If $klmn = 0100$, for example, there exists a structure boundary intercept between I, J and I, J − 1. Note that $klmn$ does not give any information about which of the several possible internal structures may be found in the indicated direction, nor does it identify the location or slope of the intercept; KEYC > 0 merely indicates that this location is just outside of some structure boundary(ies) in the indicated direction(s). Clearly there are other choices that could be made to trade memory for speed. Here, the use of the integer array KEYC(I, J) = $klmn$ minimizes storage and functions as a "pointer" in directions where boundary information may be found. We will see in the next section how this information is used to construct finite difference schemes at those points just outside the structure boundaries that reflect their existence for elliptic solutions, as well as other aspects of the simulated physics.

In practice, our codes have several cross-reference arrays that allow the easy transformation between the K and I, J index representations. Generally we use the index K to identify all points associated with a given structure; we use I, J when we need the spatial dependence of a quantity throughout the entire domain. Use of I, J indexing generally requires more storage and more IF tests to decode the, possibly multiple, intercept details around a mesh point.

This scheme allows the generalization to those boundary structures that do not actually have any mesh points inside a structure (Fig. 2c). Since these structures are defined by their analytically defined bounding walls, points can always be found that are "inside" a given wall segment—even though it may, in fact, be outside the opposite bounding segment. We simply identify points just outside the structure, define an intercept index, and associate that index with a point just "inside" that is, in fact, outside the opposite wall of the thin structure. Questions about the meaning of this subgrid geometry have to be examined on a case by case basis to clarify their meaning. For example, a thin electrode in a gun configuration, perhaps not parallel to any axes of the orthogonal mesh, does make physical sense if there are no nearby beam particles and the electrode simply contributes to an external focusing field. We find this especially useful when applying time-varying potentials that strongly influence the global fields seen by the distant charged particles. This feature provides tremendous computational savings since we need not resolve an electrode if it only provides a boundary condition and has no local physics that needs to be resolved.

III. ECB COEFFICIENTS FOR ELLIPTIC EQUATIONS

For simplicity, we consider a simple Poisson equation in 2D axisymmetric cylindrical coordinates. The equation is

$$\nabla^2\phi = (H + V)\phi = \rho, \tag{1a}$$

where

$$H\phi \equiv \frac{1}{\Delta r^2}[\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}]$$
$$\equiv \text{CRM}(I, J)\phi_{i-1,j} + \text{CRC}(I, J)\phi_{i,j} + \text{CRP}(I, J)\phi_{i+1,j} \quad (1b)$$
$$V\phi \equiv \frac{1}{\Delta z^2}[\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}]$$
$$\equiv \text{CZM}(I, J)\phi_{i,j-1} + \text{CZC}(I, J)\phi_{i,j} + \text{CZP}(I, J)\phi_{i,j+1}.$$

This is written in matrix format, for each mesh location I, J, as

$$\begin{pmatrix} & \text{CZP} & \\ \text{CRM} & \text{CRC} + \text{CZC} & \text{CRP} \\ & \text{CZM} & \end{pmatrix} \Phi = \text{RHS}, \quad (1d)$$

where, in this case, RHS is simply ρ.

No extraordinary effort is required to compute ρ near the PWLS. The equation itself is only expressed at mesh points that lie outside the structures; thus we only need ρ outside as well. In the applications section, we talk about the implications of this feature in regard to space-charge-emission.

In preparation for the solution of this system, we first form and store these coefficients for Eqs. (1b) and (1c) for all points I, J as if there were no boundaries. In the next section, we construct ''smarter'' coefficients that describe the boundary constraints at those points just outside of a structure.

A. *Dirichlet Boundary Conditions*

Dirichlet boundary conditions are used for those cases in which we wish to prescribe the value of Φ at the grid line intercept. The obvious example is the specification of the electrostatic potential on the surface of a conductor. To illustrate how this is accomplished with ECB, consider a point I, J with a grid line intercept between I, J and I − 1, J at which we require $\Phi = \text{BV}$, an array that stores boundary values. First we identify those points just outside the PWLS, where we must modify coefficients to account for the presence of the structure. If KEYC(I, J) > 0, then this point I, J is just outside a structure boundary and we must modify the coefficients at this point. For example, if an intercept lies below the point I, J in z, we will find that the l location in $klmn$ = KEYC(I, J) will have been set to one. If there is not an intercept between I, J and I, J − 1, the l will be zero.

We use simple integer arithmetic to set switches that, when appropriate, turn on one or more Dirichlet constraints. We illustrate here with the case for a Dirichlet condition between I, J and I − 1, J,

$$\text{FLRM} * \Phi(I - 1, J) = \text{BV}(K_{rm}) - (1 - \text{FLRM})\Phi(I, J), \quad (2)$$

where BV(K_{rm}) is the Dirichlet value to be applied in the minus r direction at the K_{rm} intercept. In this case, the digit k will be 1 and we will compute FLRM, the fraction of Δr the intercept is below the just outside point I, J. (We are using uniform mesh spacing here to simplify the description but uniform spacing is *not* required.) Explicitly, this cell fraction is given by

$$\text{FLRM} = +[\text{RG(I)} - \text{RINCPTBV}(K_{rm})]/\Delta r, \tag{3}$$

where RG(I) is the r-coordinate for the I, J point, Δr is the mesh size, and RINCPT(K_{rm}) is the actual intercept location for index K_{rm}.

Our goal is to use (2) and (3) to algebraically eliminate the reference to Φ just inside the structure from the basic equations in (1), evaluated at points just outside the structure. We find it convenient to multiply the constraint (2) by the corresponding field equation coefficient at this particular location and then algebraically eliminate the variable just inside the structure boundary. In our example of a point below the mesh point in the minus r direction, the constraint equation for the point I, J just outside the structure becomes

$$\text{FLRM} * \text{CRM} * \Phi(\text{I} - 1, \text{J}) = \text{CRM} * \text{BV}(K_{rm}) - (1\text{-FLRM}) * \text{CRM} * \Phi(\text{I}, \text{J}), \tag{4a}$$

and multiplying the field equations by FLRM, the coefficients become

$$\begin{aligned}
\text{TRP} &= \text{FLRM} * \text{CRP} \\
\text{TRC} &= \text{FLRM} * \text{CRC} - (1\text{-FLRM}) * \text{CRM} \\
\text{TRM} &= 0 \\
\text{TZP} &= \text{FLRM} * \text{CZP} \\
\text{TZC} &= \text{FLRM} * \text{CZC} \\
\text{TZM} &= \text{FLRM} * \text{CZM} \\
\text{T0} &= \text{FLRM} * \text{RHS-CRM} * \text{BV}(K_{rm})
\end{aligned} \tag{4b}$$

and (1d) now takes the form

$$\begin{pmatrix} & \text{TZP} & \\ \text{TRM} & \text{TRC} + \text{TZC} & \text{TRP} \\ & \text{TZM} & \end{pmatrix} \Phi = \text{T0}. \tag{4c}$$

The subscripts I, J have been suppressed. Note that multiplying all coefficients by the cell fraction FLRM rather than dividing leaves the algorithm vectorizable because no IF tests are needed to avoid division by zero.

As a second example, consider a mesh location I, J just below a z-intercept. In this case,

$$\text{FLZP} * \text{CZP} * \Phi(\text{I}, \text{J} + 1) = \text{CZP} * \text{BV}(K_{zp}) - (1.\text{-FLZP}) * \text{CZP} * \Phi(\text{I}, \text{J}), \tag{5a}$$

and this time, multiplying the field equation by FLZP, the coefficients become

$$\begin{aligned}
\text{TRP} &= \text{FLZP} * \text{CRP} \\
\text{TRC} &= \text{FLZP} * \text{CRC} \\
\text{TRM} &= \text{FLZP} * \text{CRM} \\
\text{TZP} &= 0 \\
\text{TZC} &= \text{FLZP} * \text{CZC} - (1\text{-FLZP}) * \text{CZP} \\
\text{TZM} &= \text{FLZP} * \text{CZM} \\
\text{T0} &= \text{FLZP} * \text{RHS-CZP} * \text{BV}(\text{K}_{zp}).
\end{aligned} \tag{5b}$$

B. *Neumann Boundary Conditions*

We next consider how a Neumann or normal derivative boundary condition is imposed at the ECB intercepts. The flags that determine those points whose coefficients must be modified are nearly the same as for Dirichlet boundary conditions. As before the ''just inside'' location, identified by the KEYC = *klmn* ''just outside,'' has information about at least one, to as many as four, boundary intercepts. If one of those intercepts is to have a specified normal derivative, an additional flag KN(K) will be set to one for each corresponding Kth intercept. (Here again, we make no claim that this is the most efficient vehicle to convey this information. Information such as this is essential; we have done it this way.) The boundary value for the intercept is still carried in the array BV(K). We now describe how this condition is implemented.

As before, if KEYC(I, J) = *klmn* is greater than zero, this location I, J is a location just outside a structure/grid line intercept. Depending upon the value of the digits *klmn*, we identify which directions have nearby intercepts and set the appropriate PWLS indices K. For each intercept labeled by K,

$$\text{KN}(\text{K}) = 0(1) \tag{6}$$

signifies a Dirichlet (Neumann) boundary condition.

We must modify the coefficients in the vicinity of a Neumann intercept so that the elliptic solution will correctly reflect this boundary condition. We consider a simple situation in which only one intercept is associated with location I, J and this intercept requires a Neumann boundary condition. Consider this intercept to be in the negative *r*-direction from mesh location I, J. In this case, the flag KN(K_{rm}) will be 1. This flag setting tells us that there exists an intercept between I, J and I − 1, J whose exact location is found in r_{i} = RINCPT(K_{rm}) for the *r*-coordinate and z_{j} = ZG(J) for the *z*-coordinate. The slope at that location is stored in SLPNCPT(K_{rm}). This point and slope determine the line from which the piece-wise-linear boundary segment (PWLS) has been extracted. As shown in Fig. 3, the point where the normal derivative is to be evaluated, the normal point r_{n}, z_{n}, is at the intersection of this line and the line from the ''just outside'' mesh point I, J with negative reciprocal slope $= -1/\text{SLPNCPT}(\text{K}_{\text{rm}})$. The Appendix discusses a quick way to find this normal point without the difficulties due to infinite slopes.

Once we have the coordinates of this normal point, an approximation for the

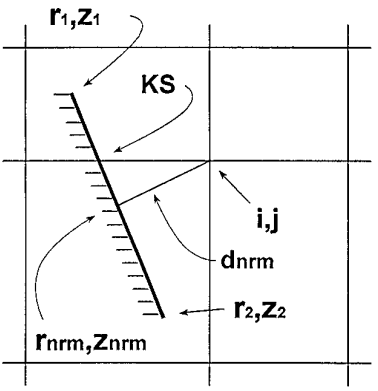

FIG. 3. Geometric definitions for a Neumann boundary condition. For a point I, J just outside a structure, the intercept in the minus r direction has intercept index K_{rm}. The normal point (where the boundary condition is applied) is the intersection of the PWLS with slope SLPNCPT(K_{rm}) that runs through the minus r intercept at RINCPT(K_{rm}) and ZG(J) and the line through mesh point I, J with a slope perpendicular to the PWLS.

normal derivative at that point can be built into the coefficients of our differential equations. When it has been determined that a Neumann condition must be imposed, we determine the unsigned distances, the d_ns, to the normal point from the mesh point just outside where we require modified coefficients. Using the definitions in Fig. 3, these distances are given by

$$d_{nrm} = \sqrt{(RG(I) - r_{nrm})^2 + (ZG(J) - z_{nrm})^2}, \tag{7a}$$

and the approximation to the Neumann derivative we use is given in the constraint equation

$$\Phi(I, J) - \Phi_{nrm} = d_{nrm} * BV(K_{rm}). \tag{7b}$$

We now solve for Φ_{nrm} and treat it as a Dirichlet value at the grid intercept, $BV(K_{rm})$, and ultimately use (2) to algebraically eliminate $\Phi(I - 1, J)$ from the finite difference equation as described in (4a) and (4b).

To summarize, KEYC at location I, J indicates the presence of and direction to nearby intercepts of the structure boundary with grid lines. In 2D, KEYC can indicate the presence of up to four nearby boundary intercepts in each of the four directions from the point. We assign a unique index K to each intercept and use it to find information describing the associated PWLS. The type of boundary condition to be applied is indicated by the array KN(K) and the value to be applied is contained in the array BV(K).

C. *Using these Modified Coefficients with ADI*

As explained in the Introduction, one of the primary reasons that we have persisted in using orthogonal meshes is that they can be solved rapidly compared to other sparse nonbanded methods. Our choice for these solutions is a version of

the alternating direction implicit method (ADI) called Dynamic ADI [12, 5]. The underlying principle is to add a fictitious time derivative to (1a) and then to integrate the resulting parabolic equation forward in time to the time-asymptotic state. At this point the fictitious time derivative goes to zero and we have the desired solution.

The ADI method is just a convenient form of operator splitting to use for this time integration. In two dimensions, the basic ADI algorithm is

$$\begin{aligned}(-\omega + \mathrm{H})\Phi^{+} &= -(\omega + \mathrm{V})\Phi + \mathrm{RHS} \\ (-\omega + \mathrm{V})\Phi^{+1} &= -(\omega + \mathrm{H})\Phi^{+} + \mathrm{RHS},\end{aligned} \tag{8}$$

where H and V are given in (1) and ω is the inverse of the fictitious time step Δt (actually, $\omega = 2/\Delta t$). The superscript identifies an intermediate level, +, and a full time step, +1. The matrix form becomes

$$(\mathrm{CRM} \quad -\omega + \mathrm{CRC} \quad \mathrm{CRP})\Phi^{+} = \mathrm{RHS} - \begin{pmatrix} \mathrm{CZP} \\ \omega + \mathrm{CZC} \\ \mathrm{CZM} \end{pmatrix} \Phi \tag{9a}$$

that, when joined with all the other I values for a given J, gives a tridiagonal system in r for each row in z. Similarly,

$$\begin{pmatrix} \mathrm{CZP} \\ -\omega + \mathrm{CZC} \\ \mathrm{CZM} \end{pmatrix} \Phi^{+1} = \mathrm{RHS} - (\mathrm{CRM} \quad \omega + \mathrm{CRC} \quad \mathrm{CRP})\Phi^{+} \tag{9b}$$

becomes a tridiagonal system in z for each column in r. (See Hewett, Larson, and Doss [5] for the details of this procedure.)

A more convenient scheme can be developed that is algebraically equivalent. This scheme, known as a ''defect correction'' scheme, takes the form

$$\begin{aligned}(-\omega + \mathrm{H})\,\delta\Phi &= \mathrm{RHS}\text{-}(\mathrm{H} + \mathrm{V})\Phi \equiv \mathrm{Res}(\Phi) \\ (-\omega + \mathrm{V})\,\Delta\Phi &= -2\omega\,\delta\Phi,\end{aligned} \tag{10a}$$

where $\mathrm{Res}(\Phi)$ is the residual of the finite-difference equation $(\mathrm{H} + \mathrm{V})\Phi = \mathrm{RHS}$, $\delta\Phi$ is a provisional correction, and $\Delta\Phi$ is the adjustment to be made to the solution as a result of this double sweep

$$\Phi^{+1} = \Phi + \Delta\Phi. \tag{10b}$$

As is apparent, this form leads to slightly less work in setting up each sweep and to a more straightforward evaluation of the residual.

For points that are inside a structure, we preset Φ to a desired interior value (e.g., the Dirichlet value of the potential for a conductor) and then guarantee that $\Delta\Phi$ is zero before the final iteration update. We define a variable δ_D that is 1 at

all grid points inside a structure (for which KEYC(I, J) = −1) and zero if outside. The tridiagonal coefficients for a row J are

$$\begin{aligned}
\text{TP}(\text{I}, \text{J}) &= (1 - \delta_D) * \text{TRP}(\text{I}, \text{J}) \\
\text{TC}(\text{I}, \text{J}) &= -\omega + (1 - \delta_D) * \text{TRC}(\text{I}, \text{J}) \\
\text{TM}(\text{I}, \text{J}) &= (1 - \delta_D) * \text{TRM}(\text{I}, \text{J}) \\
\text{T0}(\text{I}, \text{J}) &= (1 - \delta_D) * (\text{RHS}(\text{I}, \text{J}) \\
&\quad -\text{TRM}(\text{I}, \text{J}) * \Phi(\text{I} - 1, \text{J})\text{-TRC}(\text{I}, \text{J}) * \Phi(\text{I}, \text{J})\text{-TRP}(\text{I}, \text{J}) * \Phi(\text{I} + 1, \text{J}) \\
&\quad -\text{TZM}(\text{I}, \text{J}) * \Phi(\text{I}, \text{J} - 1)\text{-TZC}(\text{I}, \text{J}) * \Phi(\text{I}, \text{J})\text{-TZP}(\text{I}, \text{J}) * \Phi(\text{I}, \text{J} + 1))
\end{aligned} \tag{11a}$$

for the tridiagonal H pass (9a) and

$$\begin{aligned}
\text{TP}(\text{I}, \text{J}) &= (1 - \delta_D) * \text{TZP}(\text{I}, \text{J}) \\
\text{TC}(\text{I}, \text{J}) &= (1 - \delta_D) * \text{TZC}(\text{I}, \text{J}) - \omega) + \delta_D \\
\text{TM}(\text{I}, \text{J}) &= (1 - \delta_D) * \text{TZM}(\text{I}, \text{J}) \\
\text{T0}(\text{I}, \text{J}) &= (1 - \delta_D) * (-2\omega) * \text{T0}(\text{I}, \text{J})
\end{aligned} \tag{11b}$$

for the tridiagonal V pass (9b) where TRP(I, J), etc. come from operations similar to those that generated (4b) or (5b).

One final note is that the four edges of the 2D mesh are each considered a "structure" in our scheme. Consequently the edge points will be among those that are algebraically eliminated. Thus we only build coefficients for I, J's that are inside the 2D mesh so that we never have to worry about I ± 1 or J ± 1 reaching outside the mesh. As can be easily seen, inside a structure where $\delta_D = 1$, (11a) and (11b) are equivalent to

$$\begin{aligned}
(-\omega)\, \delta\Phi &= 0 \\
(1)\, \Delta\Phi &= 0 \\
\Phi^{+1} &= \Phi.
\end{aligned} \tag{11c}$$

IV. COMPUTING GRADIENTS AND CURLS NEAR STRUCTURE BOUNDARIES

Constructing the $\nabla\Phi$ or $\nabla \times \Phi$ (primarily for 2D solutions in which either the theta component of the magnetic field or the theta component of the magnetic vector potential has taken the place of Φ) is a matter of center differencing for the majority of the mesh locations away from both external and internal boundaries. The challenge in this section is to find suitable expressions for these operations near boundaries. At such points, we will assume that an adequately converged solution to the finite difference equation described in Section III.C exists. Using this solution, known on all points outside structures, and information about the location and slope of the intercepts, we can construct Φ on the underlying stairstep boundary corners. We will also construct approximate values for the vector fields on these same corners. These fields must be available for interpolation in cells through which the PWLS boundary passes.

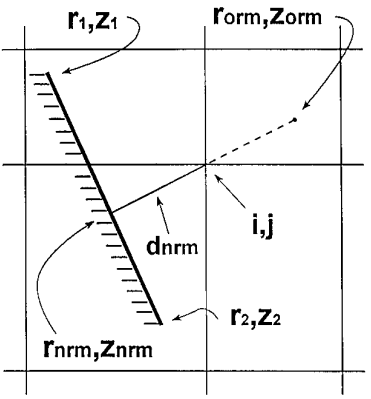

FIG. 4. The symmetry point r_o, z_o is easily determined once the location of r_n, z_n has been found.

A. *The Construction of* Φ *and* $-\nabla\Phi$ *on the Stairstep Boundary*

We begin with the construction of Φ on the stairstep boundaries. These stairstep corners are easily identified; all points for which KEYC(I, J) > 0 have a neighbor which is on a stairstep boundary. (Obviously, "thin" structures, those with no mesh points within the structure, need no Φ extrapolation and will be skipped until vector fields are computed.)

Consider a point I, J for which KEYC $= 1\delta\delta\delta$, where each of the δs could be zero or 1. We know that, regardless of the three δs, there is a structure boundary between I, J and I $-$ 1, J. Thus point I, J is just outside a stairstep boundary grid location of a structure; the intercept location is at RINCPT(K), ZINCPT(K) $=$ ZG(J), where K $=$ K_{rm} and the slope is found in SLPNCPT(K). The intercept is a fraction of a cell to the left of I, J given by (3). If this point is a Dirichlet point (KN(K) $= 0$), then we can now extrapolate to find Φ(I $-$ 1, J) from

$$\Phi(\text{I} - 1, \text{J}) = [\text{BV}(\text{K}_{\text{rm}}) - (1 - \text{FLRM})\Phi(\text{I}, \text{J})]/\text{FLRM} \tag{12}$$

and we are finished.

If this point is a Neumann point (KN(K) $= 1$), we need to find the perpendicular distance from the point I, J to the surface normal point. A straightforward way to do this that avoids the problem of infinite slopes for the *r*-intercepts is given in the Appendix. We compute the Φ value at this normal point by computing an approximate normal derivative at I, J then projecting Φ back to the normal point itself. We compute the approximate normal derivative by interpolating for Φ at the symmetry point on the opposite side of point I, J from the normal point. The symmetry point is given by

$$r_{\text{orm}} = 2. * \text{RG}(\text{I}) - r_{\text{nrm}}$$
$$z_{\text{orm}} = 2. * \text{ZG}(\text{J}) - z_{\text{nrm}}.$$

The value of Φ at that location (see Fig. 4), Φ_o is established by area weighting,

an approximation consistent with those used to construct Φ in the previous section. With Φ_o, we use a one-sided finite-difference

$$E_{nij} = -(\Phi_o - \Phi(I, J))/d_{nrm}, \tag{13}$$

where d_{nrm} is given by (7a), to get a value for the normal vector field at point I, J. This field is used to project $\Phi(I, J)$ back to the normal point with

$$\Phi_{normal} = \Phi(I, J) + d_{nrm}E_{nij}. \tag{14}$$

We do limit the magnitude of d_{nrm} to less than half of the smaller of Δr or Δz since we are, by this reflection, in a region surrounded with computed values of Φ that are consistent with small residuals. Φ_{normal} is now used as if it were a given Dirichlet value (i.e., $BV(K_{rm})$ in (12)) on the structure to linearly extrapolate back to $\Phi(I - 1, J)$.

Intercepts may exist in any of the four directions around a stairstep point. When more than one intercept exists, all will generate a Φ for this point. Ideally all values should be equal but, in practice, we average all contributions to ensure symmetry in cases where it should exist. Using this technique gives the potential at all stairstep points just inside a structure with a grid intercept between them and the $KEYC(I, J) > 0$ point. (Note that this does not include *all* the stairstep mesh points. Mesh points at a *concave corner* of the stairstep structure such as point I = 4, J = 4 in Fig. 2a are a special case and will be addressed later in this section.)

Next, we determine the vector fields ($=-\nabla\Phi$) on the points just outside the structure, $KEYC(I, J) > 0$, and those just inside on a stairstep boundary location. For the field at a point just outside the boundary, we again use Φ_o at the symmetry point opposite of the normal point behind point I, J. We now construct $E_{normal}(I, J)$ as

$$E_{normal}(i, j) = -(\Phi_o - \Phi_{normal})/2d_{nrm}, \tag{15a}$$

where Φ_{normal} comes either from the Dirichlet value or the just constructed value (14) in the Neumann case. We next interpolate values the same distance d_{nrm} on each side of point I, J but this time on a line parallel to the PWLS. We use these values to compute

$$E_{tangent}(i, j) = -(\Phi_2 - \Phi_1)/2d_{nrm}. \tag{15b}$$

We now use the slope of the associated PWLS to construct $E_r(I, J)$ and $E_z(I, J)$ from E_{normal} and $E_{tangent}$ at the point I, J just outside. Once again, these vector fields may be computed by any or all of the four possible PWLS associated with this point just outside, using the average of contributions should there be more than one. Finally, for space charge emission, we will need a $E_{normal}(K)$ at the normal point for the line segment associated with K. For the conducting wall or Dirichlet case, we have found no interpolation that is adequate for this field; we project back to this normal point using Gauss's law assuming negligible tangential derivative. The equations are

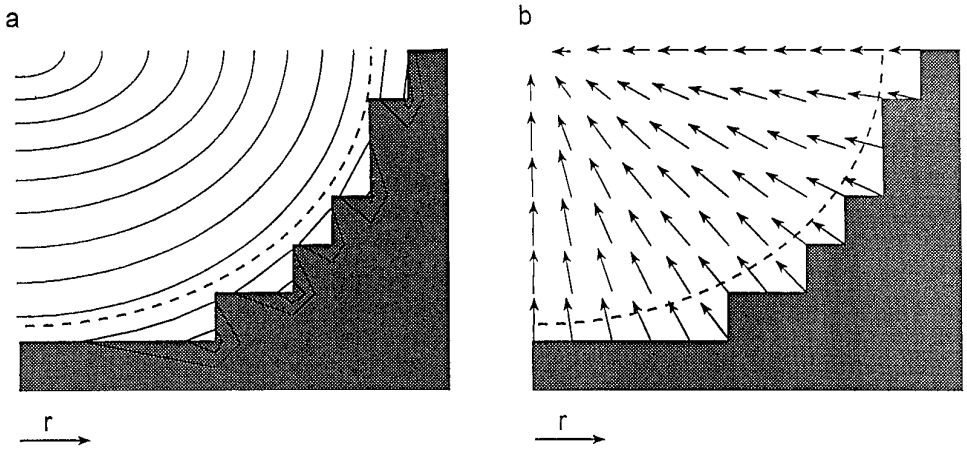

FIG. 5. The construction of Φ and $-\nabla\Phi$ values needed for interpolation to all points outside the structure requires these quantities to be constructed on all stairstep boundary locations. Shown in (a) are contours of Φ after it has been modified at the stairstep locations that lie inside the dashed structure boundary. These nonphysical values are used in the calculation of $-\nabla\Phi$ at these same locations, shown in (b).

$$E_{\text{normal}}(K_{\text{rm}}) = E_{\text{normal}}(I, J) + d_{\text{nrm}} * \rho(I, J) \tag{16a}$$

if Dirichlet, or simply

$$E_{\text{normal}}(K_{rm}) = -BV(K_{\text{rm}}) \tag{16b}$$

if Neumann.

Finally we must compute the fields on the stairstep mesh locations just inside the structure boundary. We have approximated these fields by assuming $E_{\text{normal}}(K_{\text{rm}})$ is constant along the entire PWLS and, further, that $E_{\text{tangent}}(K_{\text{rm}}) = 0$ on this PWLS. These assumptions provide the field on the grid intercept point with index K_{rm}, where a simple linear projection produces values on the interior stairstep locations—analogous to the procedure for Φ that was discussed following (14). It might be easy to improve these assumptions but this method is adequate for our present problems. We now have both Φ and the fields from its gradient approximated on the points just outside the structure and those stairstep points just inside the grid intercept locations.

There are only a few locations at which Φ and its gradient have not been computed. Values for both potential and fields still need to be set at concave stairstep locations. It is difficult to be very creative with these points because we have little physics information that is close enough for useful extrapolation. Statistically, the PWLSs relevant to these corners are more than one half of the cell diagonal away so we simply project from both sides and average. To make our procedure concrete, consider the mesh point I = 3, J = 5, in Fig. 2a, whose KEYC(3,5) describes the concave stairstep point I = 4, J = 4. Here we must find a value for $\Phi(4,4)$, having already have obtained values for $\Phi(3,5)$ (a KEYC(I, J) > 0 point with Φ from the Poisson solution) and $\Phi(4,5)$, $\Phi(4,6)$, $\Phi(3,4)$, $\Phi(2,4)$ (from the procedure outlined

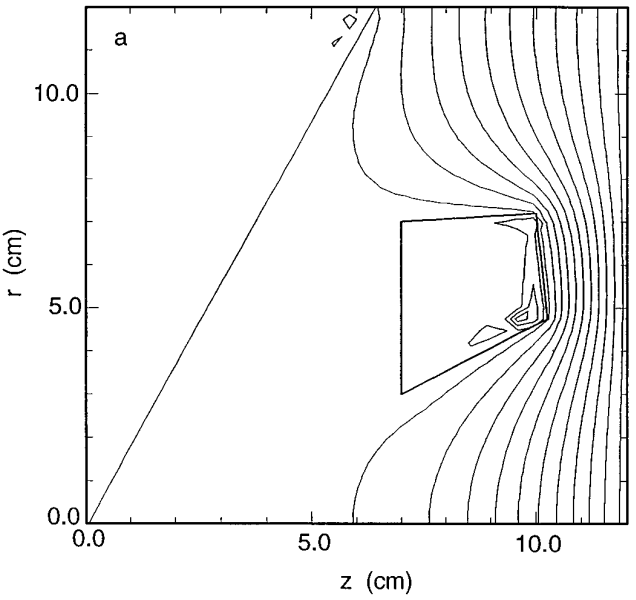

FIG. 6. A Poisson solution about a POLY4 structure (a) with a Dirichlet ($\Phi = 0$) condition on its surface and the associated negative gradient field in (b) and (c). In (c) only those fields outside the structure are plotted.

immediately above). We find that an average of the projections from both directions works well for time-dependent, space-charge-emission simulations. Explicitly,

$$\begin{aligned} \Phi^* &= 2 * \Phi(4,5) - \Phi(4,6) \\ \Phi(4,4) &= [\Phi^* + 2 * \Phi(3,4) - \Phi(2,4)]/2, \end{aligned} \tag{17a}$$

where Φ^* is a work array holding the first projection. More generally for an $r+$, z intercept associated with a KEYC(I, J) $= \delta 11 \delta$ point,

$$\begin{aligned} \Phi^* &= 2 * \Phi(\text{I} + 1, \text{J}) - \Phi(\text{I} + 1, \text{J} + 1) \\ \Phi(\text{I} + 1, \text{J} - 1) &= [\Phi^* + 2 * \Phi(\text{I}, \text{J} - 1) - \Phi(\text{I} - 1, \text{J} - 1)]/2. \end{aligned} \tag{17b}$$

The attentive reader may have noticed that in some cases in which a corner or nose of a structure extends into a cell, some values may not as yet be defined, depending on the order of operations. In these pathological cases, the geometry is considered to be sufficiently beyond grid resolution to justify setting the concave points to $\Phi(\text{I}, \text{J})$ from the associated KEYC(I, J) > 0 point; we are well past achieving further benefit from additional effort.

Finally, we find values for the fields at these concave points. Here again we use a simple average, given in this case by

$$\begin{aligned} \text{E}_r(\text{I} + 1, \text{J} - 1) &= .5 * (\text{E}_r(\text{I} + 1, \text{J}) + \text{E}_r(\text{I}, \text{J} - 1)) \\ \text{E}_z(\text{I} + 1, \text{J} - 1) &= .5 * (\text{E}_z(\text{I} + 1, \text{J}) + \text{E}_z(\text{I}, \text{J} - 1)). \end{aligned} \tag{18}$$

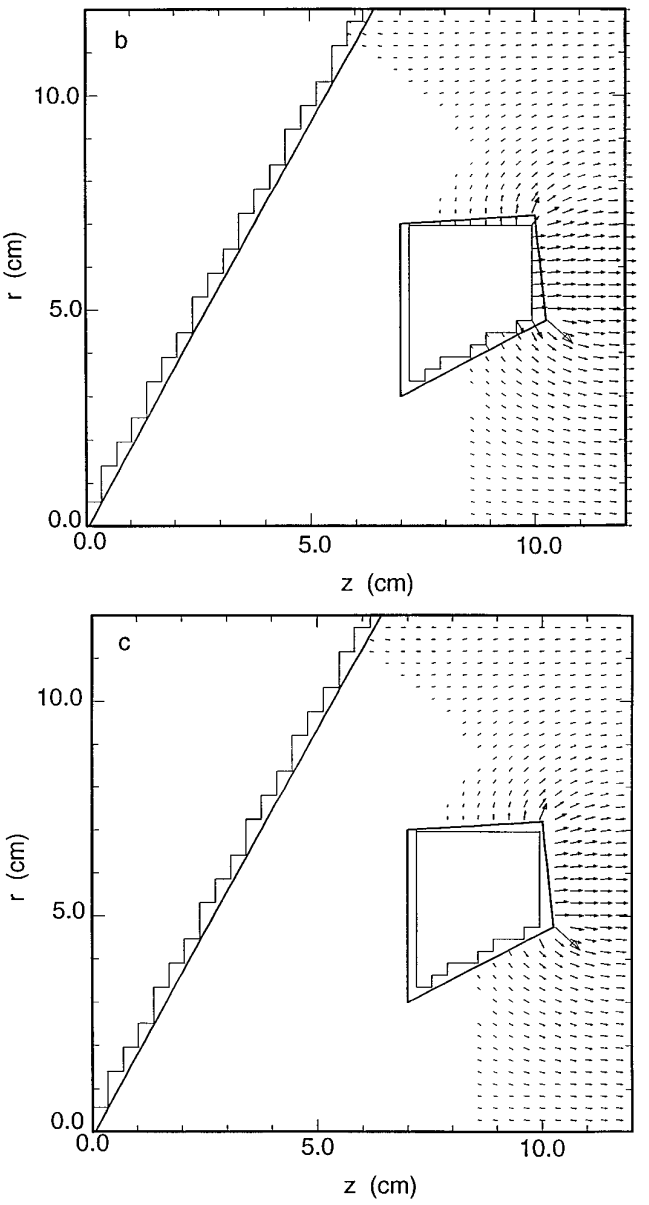

FIG. 6—*Continued*

This procedure has proven stable and, so far, better than other, more expensive options.

If there is a boundary intercept just above I, J in r, we have already computed $\Phi(I + 1, J)$ and $E_r(I + 1, J)$. However, we still need to set $E_z(I + 1, J)$ for some purposes. (Notably, PIC particles between I, J and the boundary intercept need all neighboring fields.) Fortunately, the preceding loops have set all of the *needed* eight neighboring Φ values for each KEYC > 0 and we can generate the final components of E by finite differencing. In this case for the r-direction,

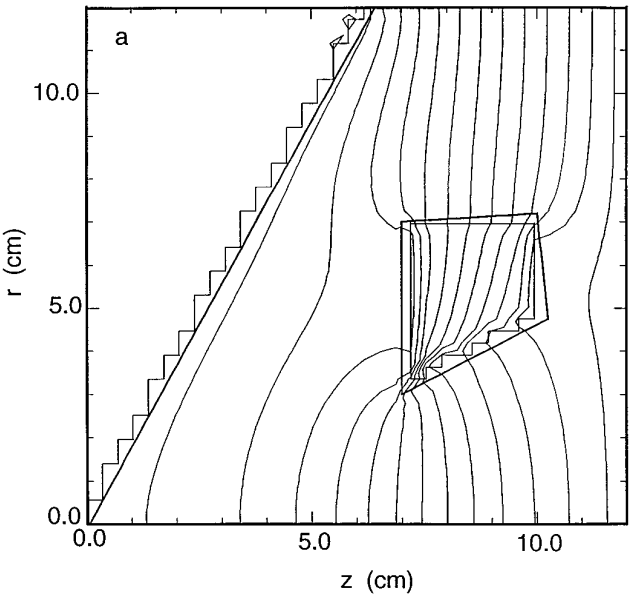

FIG. 7. Similar to Fig. 6, a Poisson solution about a POLY4 structure (a) with a Neumann ($\partial_{\text{normal}}\Phi = 0$) condition on its surface and the associated negative gradient field in (b) and (c).

$$E_z(\text{I} + 1, \text{J}) = (\Phi(\text{I} + 1, \text{J} + 1) + \Phi(\text{I} + 1, \text{J} - 1))/2\Delta z. \tag{19}$$

The obvious objective of the preceding discussion is to produce the negative gradient of the electrostatic potential. In 2D, however, it is easily possible to use these methods to obtain the curl should the solution have been a single component of a vector potential or a flux function. Had we started with a solution for the normal component of the vector potential from $\nabla^2 A_\theta = J_\theta$ rather than the Poisson equation for Φ, we could return the components of $\nabla \times A_\theta = B$ rather than $-\nabla\Phi$ by simply changing the sign of the r component then exchanging r and z components.

B. *The Results of Grad/Curl*

The preceding section, in fact, introduces nonphysical values Φ on mesh points inside the structures. As shown in Fig. 5a, a contour plot of Φ after the operations described above produces the somewhat surprising four sided contours on the stairstep corners within the dashed-line structure. The key point is that these are the Φ contours that are consistent with (and necessary for the construction of) the electrostatic field $-\nabla\Phi$ shown in Fig. 5b. If Φ is needed for other uses in the algorithm, these *ad hoc* values may be discarded or another prescription may then be used for these locations. We are now investigating the possibility of simultaneously solving Poisson's equation in multiple regions, e.g. finding Φ both inside and outside a dielectric structure, but coupled with a dielectric constraint condition on the common boundary.

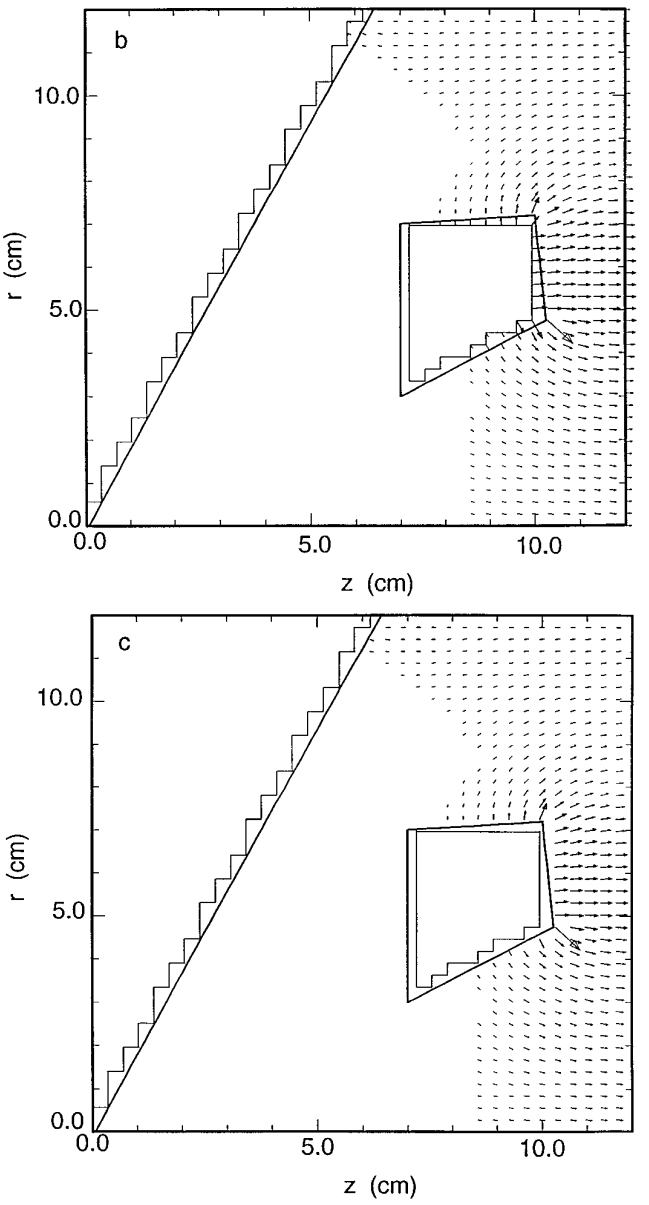

FIG. 7—*Continued*

V. APPLICATIONS

A. *Simple Structures*

We show first a simple test configuration using a four-sided POLY4 structure. Shown in Fig. 6 is a Poisson solution with Neumann ($\partial_{\text{normal}}\Phi = 0$) boundary conditions at $r_{\min}$ and $r_{\max}$. A Dirichlet zero boundary condition ($\Phi = 0$) is used for the potential on the slanted region that covers $z_{\min}$ and a nonzero Dirichlet is

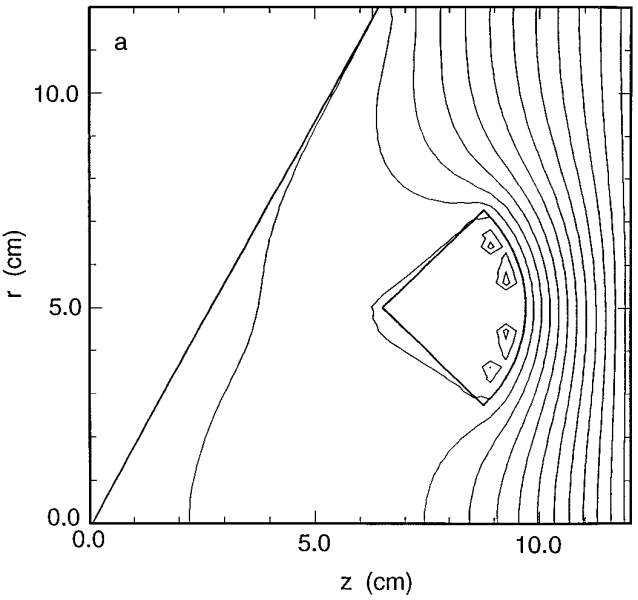

FIG. 8. Similar to Fig. 6, a Poisson solution about an ANNULUS structure (a) with a Dirichlet (Φ = const) condition on its surface and the associated negative gradient field in (b) and (c).

applied at $z_{\max}$. On the POLY4 structure in the center of Fig. 6, we have applied an intermediate Dirichlet value in (a), with the associated negative gradient field shown in (b) and (c). Figure 6b is a vector plot of $-\nabla\Phi$ which shows the field vectors at *all* mesh locations; the largest vectors are often those on the stairstep locations. (Vectors whose magnitude is more than an order of magnitude smaller than the largest vector are not plotted.) Figure 6c is the same plot as in (b) but with all vectors inside the POLY4 boundaries set to zero—such plots often give a better indication of the field inside the computational domain. As discussed above, those fields inside the structure are determined so that linear interpolation will give the correct field on the actual structure boundary. In those cases in which the intercept is ''near'' to the grid points just outside, this interpolation can require large fields at the relatively remote stairstep locations for proper interpolation. We find that acceptable field magnitudes are relatively easy to obtain; the shape of the contours and the direction of the gradients are the challenge.

In Figs. 7a–c are similar figures, but with the Neumann condition ($\partial_{\text{normal}}\Phi = 0$) on the POLY4 structure. This case shows the capability for dealing with regions with highly disparate dielectric coefficients or other similar coefficients that exhibit strong spatial dependence. Another use of such boundary conditions is the situation in which a superconducting region absorbs magnetic flux lines.

Figure 7 also highlights a limitation of ECB. With Neumann boundary conditions, we are allowing each intercept to, in effect, determine the induced charge at that intercept that is required to produce zero normal field. At structure corners at which more than one intercept attempts to set the Φ value at a stairstep location (recall that we use the average in these cases), there are not enough mesh points

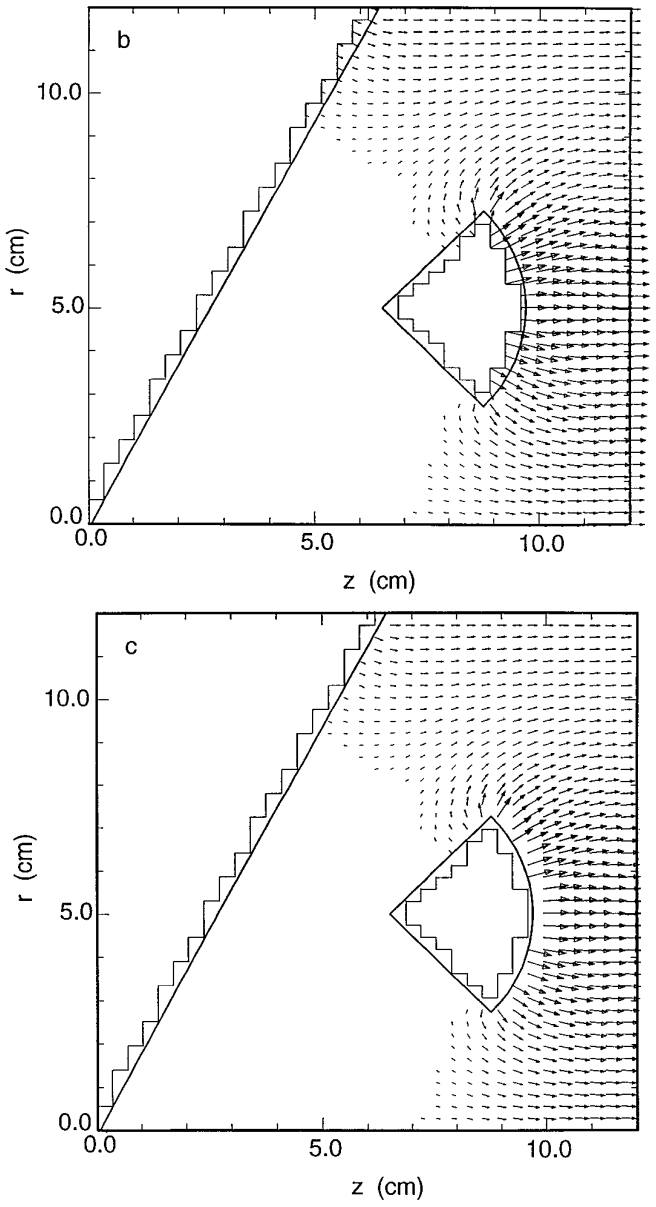

FIG. 8—*Continued*

to properly represent Φ. The difficulty shows up in the contour plots which clearly show contours of Φ that are not purely normal to the surface in that region between the mesh point just outside and the stairstep mesh locations. It is clearly possible to store the stairstep Φ value associated with each intercept and thereby get the correct gradient field to first order in the mesh spacing, but there is no way to fix Φ values on a simple mesh to recover the desired contours. Adaptive triangular meshes have the required locations; the price is the loss of a simple orthogonal mesh and increased complexity and CPU time for the field solution—the very issues

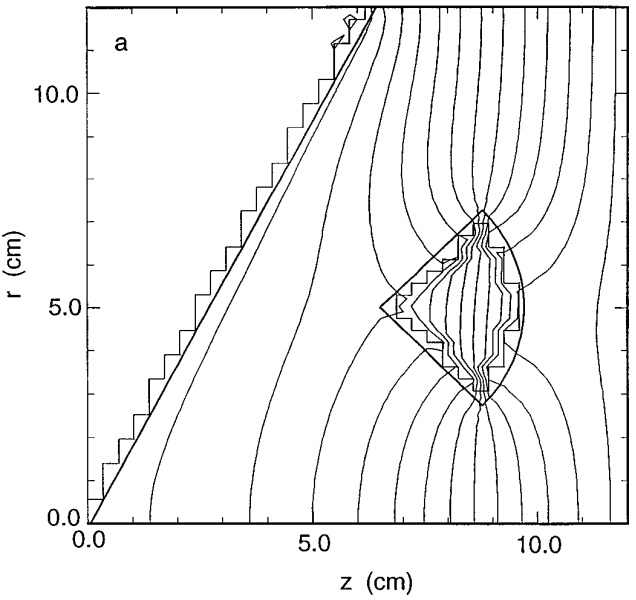

FIG. 9. Similar to Fig. 8, a Poisson solution about an ANNULUS structure (a) with a Neumann ($\partial_{\text{normal}}\Phi = 0$) condition on its surface and the associated negative gradient field in (b) and (c).

which motivated this work. This situation reminds us that to do the job better takes more mesh points, to do the job better still takes multiple values of Φ defined at these stairstep locations—and a more clever contour plotting algorithm. We emphasis that these problems occur only when more that one intercept attempts to set the associated stairstep Φ value. Straight or concave structure surfaces present no difficulty.

In Figs. 8 and 9 we show configurations completely analogous to Figs. 6 and 7 but with an ANNULUS structure substituted for the POLY4.

B. *Space-Charge-Emission*

Our space-charge-emission algorithm is strongly intertwined with the ECB concept. First we consider the results of the Poisson solution described in Section III. Recall that our scheme expresses the Poisson operator only at those points outside the structure; linear interpolation expressions for the potential at the actual location of the boundary are used to eliminate the value of the potential just inside the structure. For a conducting boundary (or Dirichlet boundary at which the potential is specified), it is implied that the amount of charge needed to produce that potential is induced on the surface. For space-charge-limited emission, it is assumed that the system is not source limited; if the normal E field is not zero, more charge is emitted from the surface until E_{normal} is zero. The surface charge density is in fact just that amount of charge necessary to make E_{normal} zero. What we do in practice is to ignore this induced charge completely in the Φ calculation. We compute the ρ at points just outside this boundary, just as at all other external mesh corners, with the standard simple linear weighting scheme. For space-charge-limited emission,

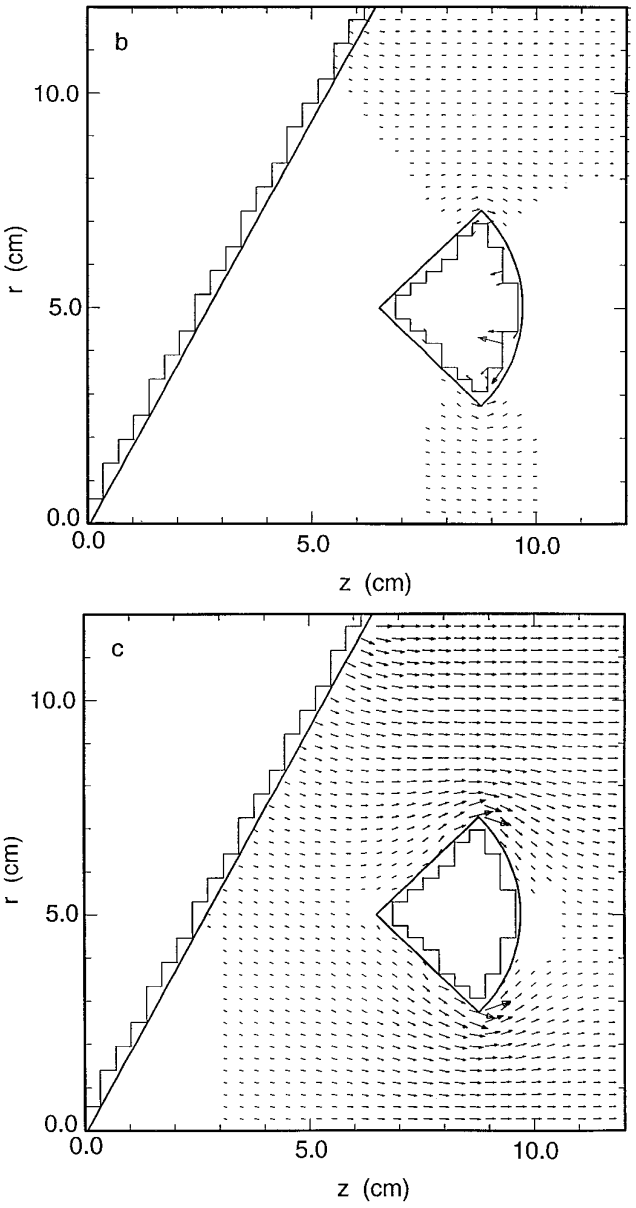

FIG. 9—*Continued*

this PIC-generated ρ will be much smaller than the total charge density at this cell. The necessary induced charge is effectively reintroduced by our algorithm in response to the nonzero E_{normal} that is produced.

In this spherical example, we keep a reservoir QRES(K) in which we place that amount of charge that the instantaneous normal E field, $E_{normal}(K)$ from (16a), would have induced at each PWLS. We emit that amount of charge (subtracting each particle's charge from QRES(K) as it is emitted) so that, if no other changes occur, $E_{normal}(K)$ will be nearly zero after the next time step. Particles are placed

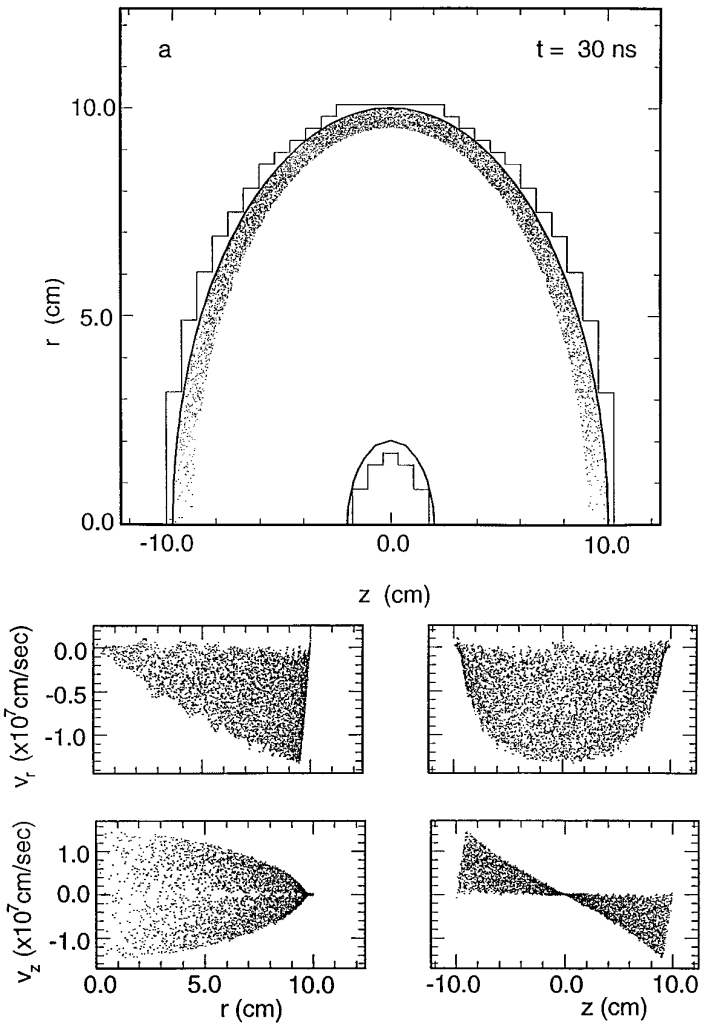

FIG. 10. Particle $r - z$ position space from an ECB simulation of space-charge-limited flow between concentric spheres. A voltage applied to the internal sphere causes space-charge-emission from the wall of the cavity. The piece-wise-linear boundary is shown along with the underlying stairstep data structure in the top position space plots for both early (a) and late times (b) in the space-charge dominated flow. In the lower parts of both (a) and (b) are the corresponding velocity vs position plots. The effect of the coarse mesh is evident primarily in early time phase space plots.

randomly along each PWLS, given a normal velocity equal to that energy the particle would have achieved if it experienced this $E_{normal}(\kappa)$ throughout the first cell,

$$qE_{normal}v_{normal} = \frac{mv_{normal}^2}{2\Delta t}$$

$$v_{normal} = \frac{2q\Delta t E_{normal}}{m},$$

and given a tangential velocity related to a realistic surface temperature,

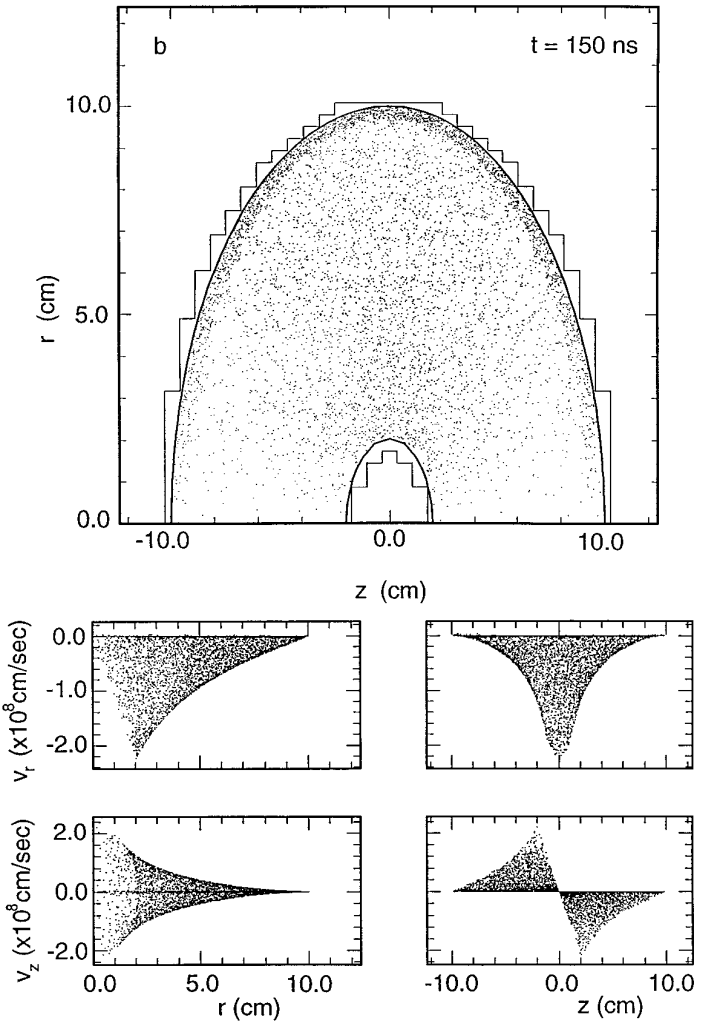

FIG. 10—*Continued*

$$v_{\text{tangential}} = \sqrt{k_b T_{\text{surf}}/m}.$$

A normal trajectory can now be defined for each particle that is the path this particle would traverse if allowed to move ballistically for a full time step. Each particle is then randomly positioned along its normal trajectory, consistent with their being emitted at random times during the previous time step.

The effect of this type of boundary condition is to provide enough charge in the first cell to cancel the E_{normal} that existed the preceding time step. Movement of the particle previously emitted then generates a new E_{normal} that provides for realistic, continuous emission. One-dimensional tests show that Child's law is easily satisfied to a small fraction of a percent over about 40 uniformly spaced mesh points [15].

We demonstrate this space-charge-emission algorithm and its compatibility with ECB with a test case consisting of space-charge-limited emission and flow between

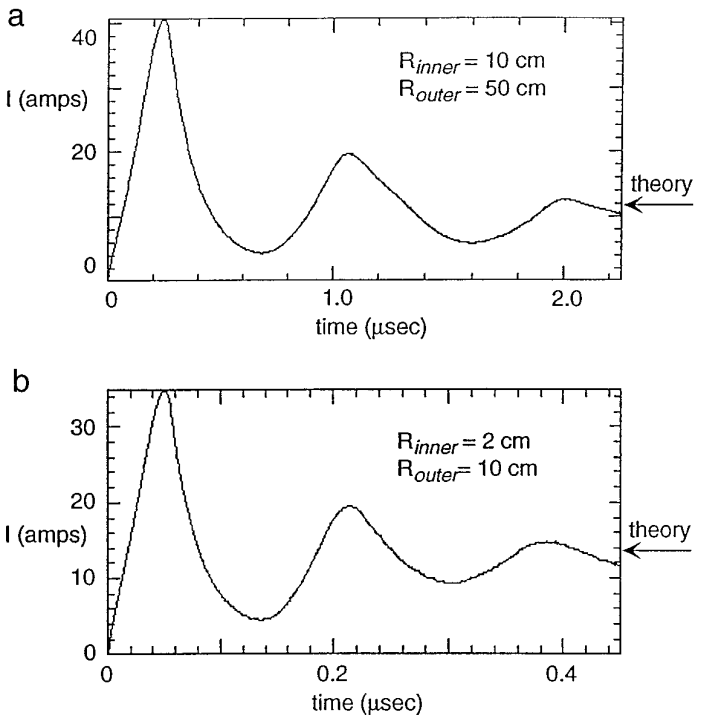

FIG. 11. Ion currents for the concentric ion source shown in Fig. 10. In both (a) and (b) is the current averaged over 0.9 to 1.0 times R_{outer}. A space-charge-limited flow of potassium ions is extracted by a 900 kV potential on the center electrode. In (a) is a system with R_{inner} = 10 cm and R_{outer} = 50 cm. In (b) is the analogous system with R_{inner} = 2 cm and R_{outer} = 10 cm. The theory of Langmuir and Blodgett suggests that the total current I should be independent of the ratio of collector to emitter radii, in agreement with our simulations (courtesy of L. S. Tung).

two concentric spherical electrodes. In Fig. 10 we use this emission boundary condition on the inside surface of the outer electrode and pull charged particles from this surface with a Heaviside function voltage applied to the smaller internal sphere. The piece-wise linear boundary is shown, along with the underlying stairstep data structure in the top position space plots for both early (Fig. 10a) and late times (Fig. 10b), in the space-charge dominated flow. In the lower parts of both (a) and (b) are the corresponding velocity versus position plots. In each of these phase space plots we have plotted all particles in the entire simulation.

Little effect of the coarse mesh is evident; the particle flow is quite smooth even though the data structures are very coarse. The underlying coarse mesh is visible primarily in the v_r versus r and v_r versus z plots of the coarse mesh early in time, although this "ripple" is well within the emitted particle envelope and certainly is not evident in the late time plots. The particles in this simulation spend over 90% of their simulation lifetime within one cell of the ECB boundary; evidently what little ripple these particles experience has little deleterious effect on their later trajectories.

As shown in Fig. 11 we find steady-state currents agree with theory [16] within a few percentages, even for this coarse mesh resolution of the concentric spheres.

Figure 11 also shows the expected size invariance, so long as the ratio of inner to outer radius is kept constant.

Since computational speed is a rather strong function of the number of mesh unknowns, the speed advantage of such a coarse representation is obvious. Even with a very large number of mesh points for a stairstep boundary, the direction of emitted particle velocities will be incorrect, producing nonphysical transverse temperatures for the generated beam. Stairstep meshes based on orthogonal adaptive mesh refinement will also have this difficulty; the correct orientation of particle velocities is essential. As we have pointed out, ECB is not perfect either, but we find the ratio of performance to computational effort to be very attractive.

We have also applied these techniques to more general geometries. As can be seen in Fig. 12, realistic geometries can be built using a combination of our POLY4 and ANNULUS structures. The 2D results shown here have been found to be in excellent agreement with experiment and steady state EGUN [17] calculations originally used for the design of this system. It was our time dependent simulations, however, that revealed beam transient issues in early designs which led to the present electrode configuration [18].

VI. SUMMARY

We have developed a new technique to represent a curved boundary by piecewise linear segments within an orthogonal computational mesh. For many problems, the simplicity and speed advantages achieved with our embedded curved boundaries techniques outweigh the more powerful and precise capabilities of adaptive mesh methods. While careful attention to detail is essential to achieving the accuracy that ECB can provide, the methods described here are now robust and easy to use. ECB provides most of the generality of finite elements while retaining most of the speed of orthogonal meshes. The ECB technique provides accuracy near curved boundaries attainable by adaptive finite element representations of similar size while retaining the speed of banded matrix solutions. As suggested early in the text, it is easy to extend these concepts to a nonuniform mesh (see Fig. 13), as well as to include the ability to handle $[\varepsilon \nabla_{\text{normal}} \Phi]_1^2 = \sigma$ boundary conditions.

Most importantly, since ECB does not change the matrix structure but only the element values near a structure, our scalable elliptic solution technique for massively parallel processors [7] can be used without modification. We are now working on this new combination.

APPENDIX: AVOIDING ZEROS AND INFINITIES IN DETERMINING THE NORMAL POINT

Certainly we expect that the slopes of piece-wise linear segments PWLS can have all values between $-\infty$ and $+\infty$. However, we may exploit our knowledge of these boundaries to avoid problems with singularities when we solve for these intercepts. If we have a PWLS intercept between $\text{I} - 1$ and $\text{I} + 1$, we will assume that the slope of this line is not zero. Similarly, if the intercept is between $\text{J} - 1$ and $\text{J} + 1$,

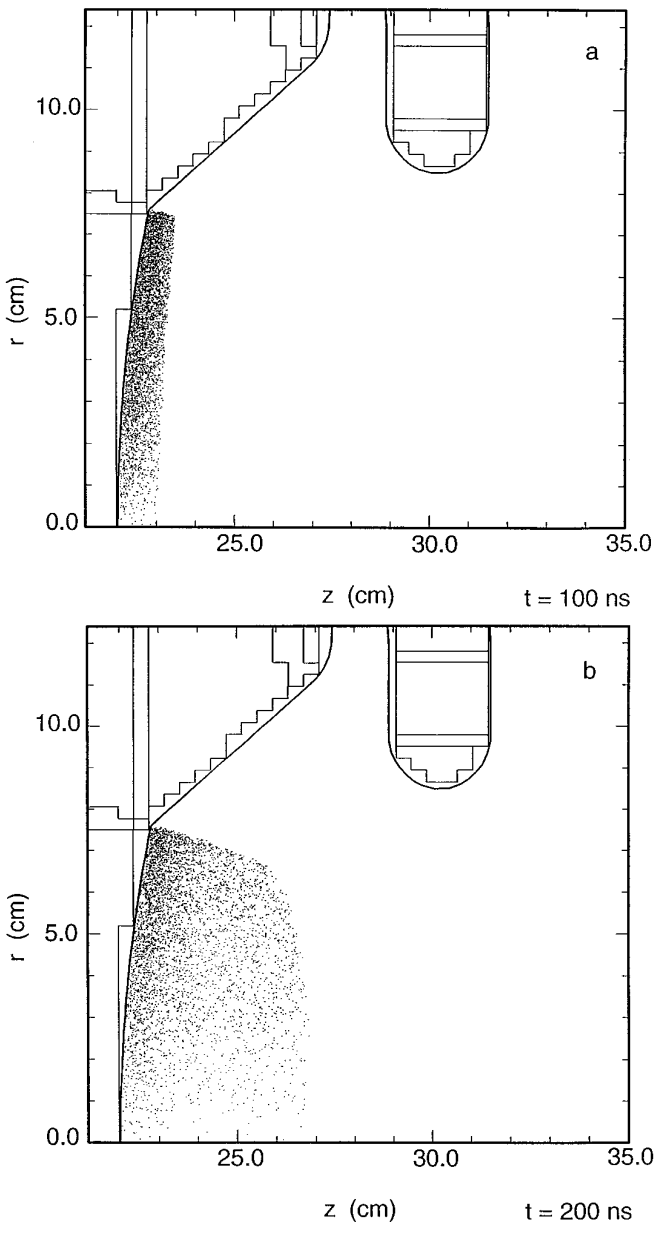

FIG. 12. A time sequence of particle positions in $r - z$ reveals the expected laminar particle trajectories as well as benign transients on the head and the tail of the heavy ion pulse. In (a) and (b) are the ion beam configurations at 100 and 200 ns, respectively. In (c) and (d) are contours of Φ and the E field at 200 ns, and (e) and (f) show more ion position space configurations at 350 and 600 ns, respectively. The voltage on the rounded electrode extending from the top of each plot is held at 900 kV while the voltage on the spherical anode on the left (as well as the "Pierce" angle electrode that surrounds it) is initially at 800 kV. The spherical electrode group was linearly raised to 1 MV during the first 150 ns, held until 350 ns, and then reduced to 800 kV. Note the tendency of the space-charge-dominated beam to expand sideways as the voltage is reduced.

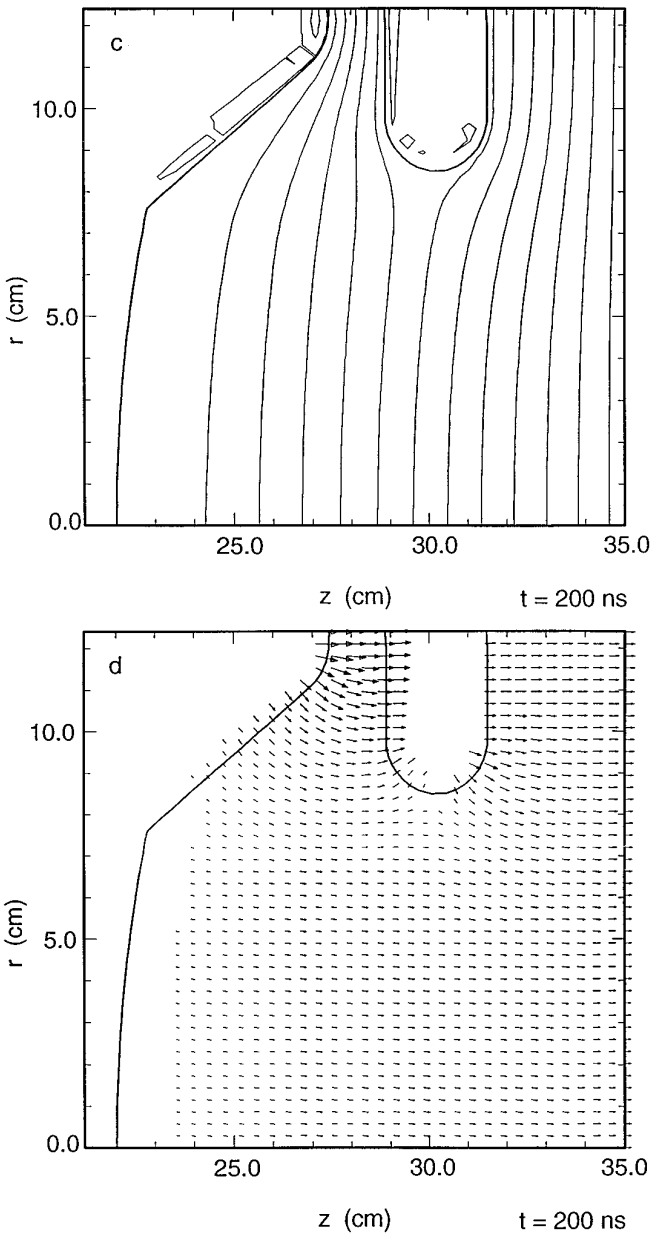

FIG. 12—*Continued*

we will assume that the segment does not have infinite slope. Exploiting these concepts and, together with the information in the flags, we can find the intercepts by

r minus intercept,

$$\text{SLPIRM} = +1/\text{SLPNCPT}(\text{K}_{\text{rm}})$$

$$r_{\text{rm}} = \frac{\text{RINCPT}(\text{K}_{\text{rm}}) + \text{RG}(\text{I}) * \text{SLPIRM}**2}{1. + \text{SLPIRM}**2}$$

$$z_{\text{rm}} = \text{ZG}(\text{J}) - \text{SLPIRM} * (r_{\text{rm}} - \text{RG}(\text{I}))$$

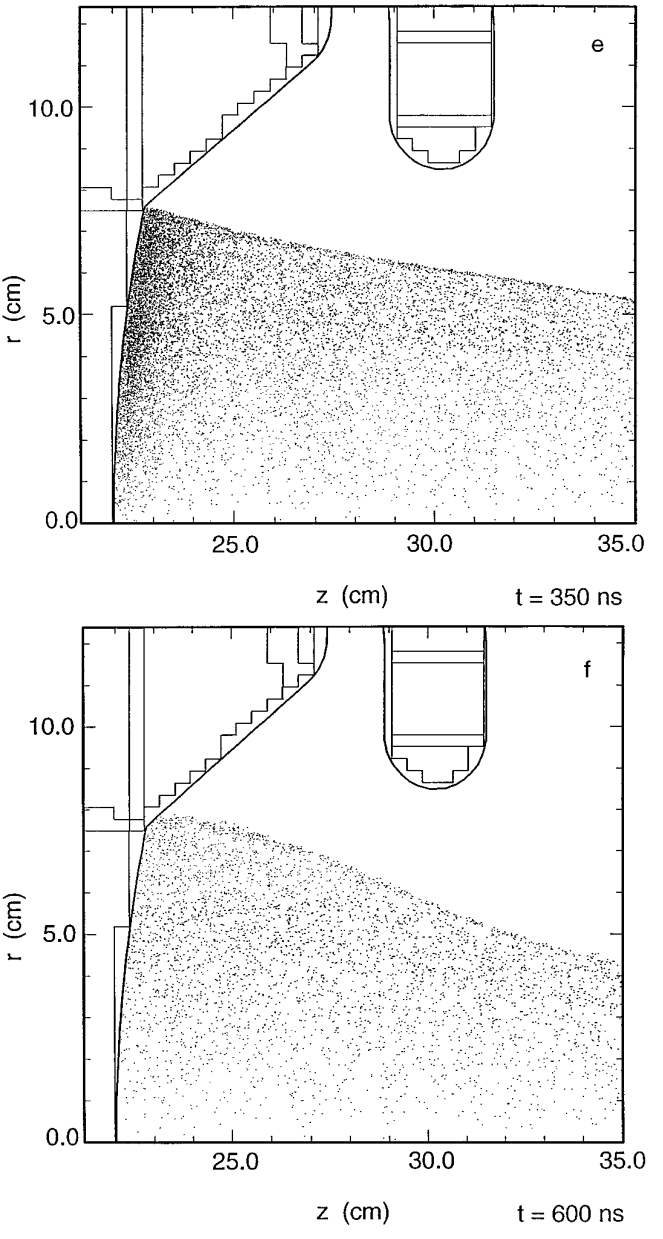

FIG. 12—*Continued*

z minus intercept,

$$r_{\mathrm{zm}} = \frac{\mathrm{ZINCPT}(\mathrm{K_{rm}}) + \mathrm{ZG}(\mathrm{J}) * \mathrm{SLPNCPT}(\mathrm{K_{rm}})^{**}2}{1 + \mathrm{SLPNCPT}(\mathrm{K_{rm}})^{**}2}$$

$$r_{\mathrm{zm}} = \mathrm{RG}(\mathrm{I}) - \mathrm{SLPNCPT}(\mathrm{K_{zm}}) * (z_{\mathrm{zm}} - \mathrm{ZG}(\mathrm{J}))$$

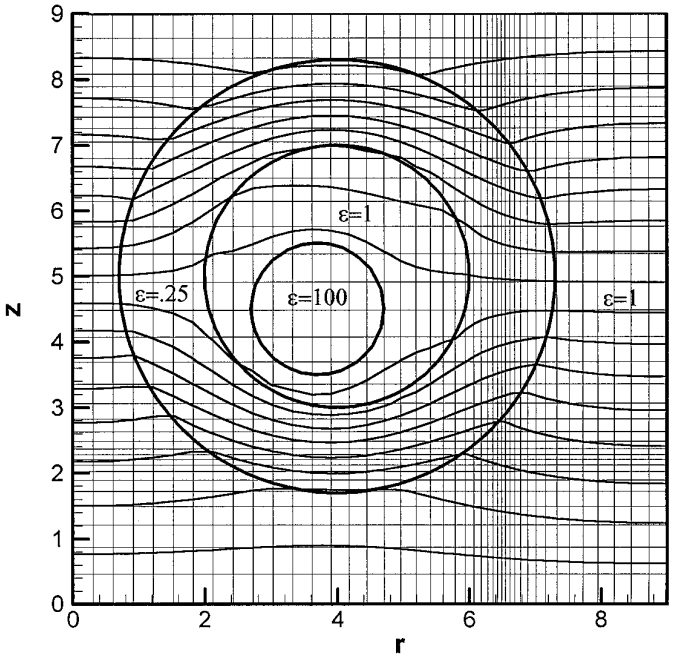

FIG. 13. Straightforward extensions to ECB concepts allow nonuniform orthogonal meshes as well as the ability to handle dielectric $|\varepsilon\nabla_{\text{normal}}\Phi|_1^2 = \sigma$ boundary conditions.

r plus intercept,

$$\text{SLPIRP} = -1/\text{SLPNCP}t(\text{K}_{\text{rp}})$$

$$r_{\text{rp}} = \frac{\text{RINCPT}(\text{K}_{\text{rp}}) + \text{RG}(\text{I}) * \text{SLPIRP}**2}{1 + \text{SLPIRP}**2}$$

$$z_{\text{rp}} = \text{ZG}(\text{J}) - \text{SLPIRP}*(r_{\text{rp}} - \text{RG}(\text{I}))$$

z plus intercept,

$$z_{\text{zp}} = \frac{\text{ZINCPT}(\text{K}_{\text{zp}}) + \text{ZG}(\text{J}) * \text{SLPNCPT}(\text{K}_{\text{zp}})**2}{1 + \text{SLPNCPT}(\text{K}_{\text{zp}})**2}$$

$$r_{\text{zp}} = \text{RG}(\text{I}) - \text{SLPNCPT}(\text{K}_{\text{zp}}) * (z_{\text{zp}} - \text{ZG}(\text{J})),$$

where the r-intercepts are formulae found from interchanging the r- and z-coordinates.

ACKNOWLEDGMENTS

I am pleased to acknowledge useful discussions with D. J. Larson concerning ADI and M. R. Gibbons for reinforcing some interpolation procedures. Y.-J. Chen and L. S. Tung have assisted with space-

charge emission applications. H. M. Lades has provided assistance with the graphics, and C. S. Kueny's suggestions have improved clarity.

REFERENCES

1. R. Löhner, An adaptive finite element scheme for transient problems in CFD, *Comput. Methods Appl. Mech. Eng.* **61,** 323 (1987).
2. G. L. Lapenta and J. U. Brackbill, Dynamic and selective control of the number of particles in kinetic plasma simulation, *J. Comput. Phys.* **115,** 213 (1994).
3. A. Taflove, *Computational Electrodynamics* (ARTECH House, Norwood, MA, 1995).
4. M. E. Jones, Electromagnetic PIC codes with body-fitted coordinates, in *12th Conference on the Numerical Simulation of Plasmas, San Francisco, CA, Sept. 20–23, 1987.*
5. D. W. Hewett, D. J. Larson, and S. K. Doss, Solution of simultaneous partial differential equations: Solution of the streamlined Darwin equations, *J. Comput. Phys.* **101,** 11 (1992).
6. D. W. Hewett, Low-frequency electromagnetic (Darwin) applications in plasma simulation, *Comput. Phys. Commun.* **84,** 243 (1994).
7. M. A. Lambert, G. L. Rodrigue, and D. W. Hewett, A parallel DSDADI method for solution of the steady state diffusion equation, *Parallel Comput.*, in press.
8. G. L. Lapenta and J. U. Brackbill, Immersed boundary method for plasma simulation in complex geometries, *IEEE Trans. Plasma Sci.* **24**(1), 105 (1996).
9. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25,** 220 (1977).
10. D. Sulsky and J. U. Brackbill, A numerical method for suspension flow, *J. Comput. Phys.* **96,** 339 (1991).
11. A. Mayo and A. Greenbaum, Fast parallel iterative solution of poisson's and the biharmonic equations on irregular regions, *SIAM J. Stat. Comput.* **13**(1), 101 (1992).
12. T. D. Pointon, Slanted conducting boundaries and field emission of particles in an electromagnetic particle simulation code, *J. Comput. Phys.* **96,** 143 (1991).
13. N. Mattor, T. J. Williams, and D. W. Hewett, An algorithm for massively parallel solution of banded matrices, *Parallel Comput.* **21,** 1769 (1995).
14. S. K. Doss and K. Miller, Dynamic ADI methods for elliptic equations, *SIAM J. Numer. Anal.* **16,** 837 (1979).
15. D. W. Hewett and Y. J. Chen, Proc. 1993 Comp. Accl. Phys. Conf., Pleasanton, CA; H.R. Rutkowski, D. W. Hewett, and S. Humphries Jr., *IEEE Trans. Plasma Sci.* **19**(5), 782 (1991).
16. I. Langmuir and K. B. Blodgett, Currents limited by space charge between concentric spheres, *Phys. Rev.* **24,** 50 (1924).
17. W. B. Hermannsfeldt, *Electron Trajectory Program*, Standford Linear Accelerator, Standford, CA, SLAC-226 (November 1979).
18. D. W. Hewett and Y. J. Chen, *PIC Space-Charge Emission with Finite* Δt *and* Δz, LLNL ICF Quarterly Report **3,** No. 4, 166.