JOURNAL OF COMPUTATIONAL PHYSICS 114, 161–175 (1994)

Particle-in-Cell Simulations with Moving Boundaries—Adaptive Mesh Generation

T. WESTERMANN*

Kernforschungszentrum Karlsruhe GmbH, Postfach 3460, D-76021 Karlsruhe, Germany

Received July 27, 1993; revised February 2, 1994

The particle-in-cell method is coupled with boundary-fitted grids in order to model the stationary Maxwell–Lorentz problem in technical devices. The aim of this paper is to introduce techniques with which one is able to change the plasma boundaries and subsequently the computational grid within a particle-in-cell simulation according to the physical quantities which are calculated from the dynamical behavior of the electromagnetic device. Hence, the discussion is focused on the problem arising with moving boundaries and adaptive meshes. New numerical methods handling the reorganization of the grid, the transformation and rescaling of the particles, and the recalculation of the field data are presented. The modelling process is described and the numerical algorithms are discussed. © 1994 Academic Press, Inc.

1. INTRODUCTION

Calculating electromagnetic effects by solving Maxwell's equation often requires the aid of computers. This is especially true when the system consists of complicated boundaries. Only in very special cases in which the boundaries play no important role, is one able to solve Maxwell's equations analytically. Hence, computer simulations are an indispensable tool in plasma physics and other fields dealing with electromagnetic phenomena. Maxwell's partial differential equations describe the fields, and the Lorentz equation expresses the motion of electrically charged particles in these fields. The electromagnetic fields determine the behavior of charged particles. The space charge of the particles themselves create electromagnetic fields. These fields again influence the particles, and so on. A traditional numerical method for solving this nonlinear Maxwell–Lorentz problem is the particle-in-cell (PIC) approach [1, 2].

The PIC method is an attractive computational tool to study kinetic phenomena, in particular in plasma physics: A grid is introduced in order to compute the partial differential equations for the electromagnetic fields, and particles carrying electric charge and mass are advanced in these fields by solving the Lorentz equation of motion. Originally, the PIC method was developed using uniform [1, 2] or nonuniform [3–5] grid zoning with grid lines parallel to the coordinate axes. However, with these methods it is not possible to treat complicated devices without simplifying the geometry.

In particular, when charged particles are created at surfaces and accelerated in these technical devices, it is essential to model the boundaries carefully in the numerical approximation. In the past, most concepts were not able to treat curved shapes of the emitting parts, and thus they could not adequately model the influence of these shapes on the beam properties. It also turned out that simplifications of parts of the geometry, e.g., at edges and curved parts of the electrodes, led to artificial field-enhancement and, thus, to a distorted flow of particles.

To overcome these difficulties, PIC methods using grids fitted to the boundary of the physical domain [6–9] were developed. The concept of structured boundary-fitted coordinates [10] is based on the basic idea that a logical, rectangular grid is introduced and transformed onto the physical domain. The field equations are solved on the logical grid but the particles are advanced in the physical space.

By applying boundary-fitted coordinates as the grid concept underlying the simulations, it is possible to realistically treat technical devices. The BFCPIC code (particle-in-cell code based on boundary-fitted coordinates) [7, 11] is designed in such a way that curved boundaries and corners are permitted. In particular, the shapes of the anode and cathode surfaces are modelled quantitatively, and thus the simulation of the emission of particles on highly stressed electrodes is treated accurately. However, when modelling pulsed power ion diodes it turned out that the boundary of the electromagnetic device is not only curved and arbitrarily shaped but also time-dependent. If the normal electric field exceeds a threshold field of about 10^7 V/m, a dense plasma quickly forms on the conducting surfaces. Instantaneously, particles are emitted from the outer surface of the plasma.

Since the anode, as well as the cathode, plasma expand,

* Present address: Fachhochschule Karlsruhe, Fachbereich Naturwissenschaften, Postfach 2440, D-76012 Karlsruhe, Germany.

0021-9991/94 $6.00

the experimental gap distance is time-dependent. In particular, this effective gap width is smaller than assumed by the geometrical gap. Up to now, this time-dependent movement of the plasmas was taken into account simply by shifting the anode and cathode plasmas into the gap and performing simulations with a reduced but fixed gap. But this movement of the plasma surfaces not only has an impact on the gap distance but also on the focusing properties: depending on the dynamic shape of the emitting plasma surface, different focusing properties are obtained.

In order to be able to include the dynamical expansion of the emitting plasma surfaces, we extended the grid model to react on the different shapes of the anode and cathode plasma surfaces during a simulation.

The aim of this paper is to introduce techniques with which one is able to change the emission surfaces in the BFCPIC code according to the physical quantities computed from the dynamical behavior of the electromagnetic device. In contrast to previous work (e.g., Brackbill *et al.* [12, 13], Thompson [14], who focused on the problem of rezoning the grid due to the solution in a sophisticated manner), we chose a simple grid modification model introduced by Thompson [10] and outline the procedure of combining adaptive mesh zoning with the PIC method. In more detail, we do not adapt our mesh to the solution of the problem in order to obtain a better resolution of this numerical solution or to improve the grid in terms of smoothness and orthogonality, but we treat the problem how to handle physical effects where the boundary of the computational domain changes during the simulation.

Thus, it is the purpose of this paper to describe the coupling of adaptive meshes with the PIC method and to develop techniques in order to obtain a smooth transition of the particles in these changing grids.

The organization of the paper is as follows: In Section 2 we introduce the experimental apparatus and the mechanics of the externally applied-B ion diode. In Section 3 the grid generation procedure is reviewed and the governing equations as well as the numerical representation are discussed. In Section 4 characteristic features of the numerical program BFCPIC used for the simulations are outlined. Section 5 deals with the methods of handling the movement of emitting surfaces as well as the special treatment of the particles. The implementation of the methods is outlined and improvements are described. Examples of computations for the externally applied-B ion diode are presented in Section 6. Finally, we restate our conclusions in Section 7.

2. THE EXTERNALLY APPLIED-B ION DIODE

Intense light-ion-beam diodes [15] for inertial confinement fusion [16] are an example for the necessity to model the boundaries accurately. Figure 1 shows a schematic cross

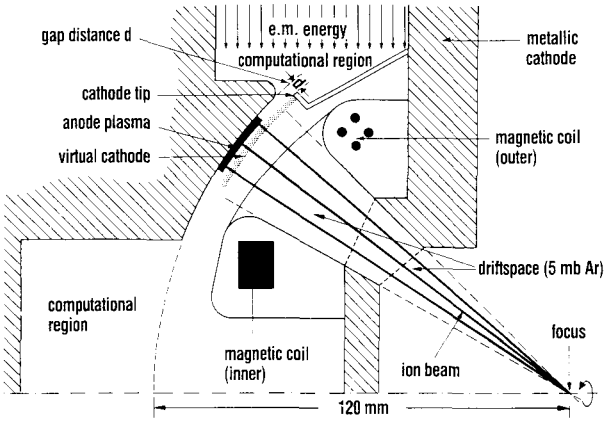


FIG. 1. Schematic cross section of the externally applied-B ion diode in the (z, r)-plane.

section of a rotationally symmetrical externally applied-B ion diode in the (z, r)-plane developed at the Nuclear Research Center in Karlsruhe by Bluhm *et al.* [17]. The vacuum diode consists of a solid anode plate and a cathode ring. A mylar foil attached to the cathode ring seals the vacuum diode from the gas-filled drift space. Depending on the ion species to be produced, parts of the anode plate are coated with an appropriate material. These parts are outlined by the dashed lines in the schematic diagram of the diode. The pulse length is typically about 50 to 100 ns, the applied voltage is several mega volts, and the gap distance d between the anode and the cathode tip is a few millimeters.

The electron current follows the outer conductor of the generator to the cathode area and to the cathode tip. Electric field enhancement causes electrons to be emitted mainly at this edge. Due to a special preparation an anode plasma is generated in front of the anode. This anode plasma serves as an ion source. The ions are accelerated in the anode–cathode gap and leave the diode through the mylar foil into the gas-filled drift space. The electrons are influenced by magnetic fields induced by the motion of the particles themselves and also by external magnetic fields in the (z, r)-plane. The external magnetic field is produced by two magnetic coils and indicated in Fig. 1. The electrons drift in the gap perpendicular to the electric and magnetic fields.

Due to the strong magnetic field, the electrons stay close to the cathode and are not able to cross the gap; this is known as magnetic insulation. In order to illustrate the principle of magnetic insulation, the orbit of a single electron is studied. It can be seen in Fig. 2a that the electron cannot reach the anode surface. The gap is magnetically insulated. Due to the external (B_z, B_r)-field, the electron velocity also has an azimuthal component, indicated in Fig. 2b.

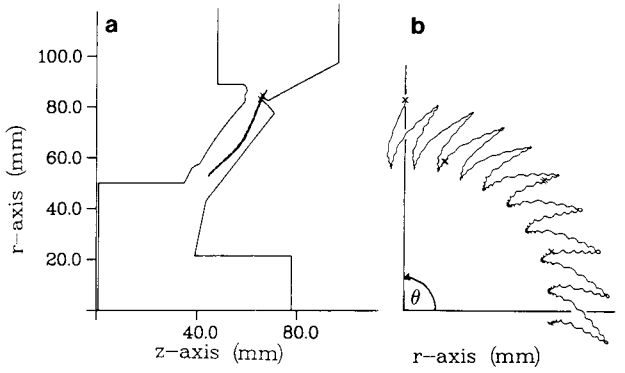


FIG. 2. Trajectory of a single electron emitted at the cathode tip: (a) (z, r)-plane; (b) (r, θ)-plane.

The aim of ion diode research is to convert the supplied electric energy into ion beam energy. To reach sufficiently high power densities the electric power supplied to the system has to be converted efficiently into focused ion beam energy. Due to the complexity of the geometrical influences and space charge effects, analytic models do not describe the focusing properties of these diodes satisfactorily. Therefore, numerical simulations are performed in order to be able to compute the focusing properties numerically.

Since the anode, as well as the cathode, plasmas expand during the 50–100 ns pulse, the experimental gap distance is time-dependent. In particular, this effective gap width is smaller than the geometrical gap. Up to now, this time-dependent movement of the plasmas was taken into account using a rough model: To obtain an estimate for the plasma thickness at the voltage plateau we assumed an anode and cathode plasma movement of 3 cm/μs and 1.5 cm/μs, respectively. Since we know from experimental measurements that this plateau is reached after about 50 ns, the plasma thickness should be about 2 mm for the anode plasma and 1 mm for the cathode plasma. Hence, in order to take the plasma movement into account, the emitting anode surface was shifted by 2 mm and the cathode tip by 1 mm into the gap, reducing the gap width to 5.75 mm. Thereby, magnetic diffusion in these surface plasmas is assumed.

This movement of the plasma surfaces not only has an impact on the gap distance but also on the focusing properties: It has been found experimentally [17] that even for an aspherically shaped anode surface, the ion beamlets overfocus at the drift section. This should not be the case if there is only a shift of the emitting surfaces. One possible explaination is that the different expansion of the plasma surface is responsible for this effect.

In order to simulate such an effect, we extended the grid model to be able to react on the different shapes of the anode and cathode plasma surfaces during the simulation.

3. GRID GENERATION PROCEDURE

3.1. *Basic Concept*

In order to compute discrete solutions for the electromagnetic fields in a technical device, partial differential equations have to be solved in a domain with complicated boundaries and mixed boundary conditions. A standard discrete description of a geometry is given by a monoblock equidistant representation. It is the simplest one with respect to the data structure: Each interior grid point has an upper, a lower, a right, and a left neighboring point. A more flexible grid is a block-structured grid, where different blocks can be used for different parts of the geometry. The most flexible concept for the discrete description of a geometry is given by an unstructured mesh. It has the highest degree of freedom. However, the drawback consists in the fact that the data structure is very complicated.

A serious alternative to unstructured meshes are boundary-fitted grids. They are nearly as flexible as unstructured meshes but have a regular data structure. They are logically equivalent to a monoblock grid. Hence, they are well suited for finite difference schemes. As will be discussed, they are easy to construct and standard methods can be used for solving the grid equations. Moreover, for our special purposes to change the boundary as well as the computational mesh within a simulation it is convenient to work with boundary-fitted coordinates, as long as the logical structure is not distorted.

The computational grid is created numerically using the powerful approach of Thompson [10] and which was adapted for our special applications of pulsed power ion diodes by Halter [18]. The basic idea is the transformation of the physical area to a logical area. The boundary of the physical area is mapped onto the boundary of a rectangle. In this rectangle an equidistant grid is defined and transformed back to the physical area (cf. Fig. 3).

3.2. *Governing Equations*

The grid system is computed by solving elliptic partial differential equations: Let $\xi(x, y)$, $\eta(x, y)$ be the mappings of the physical area in the (x, y)-plane onto the logical area in the (ξ, η)-plane given as the solution of the elliptic system

$$\Delta\xi(x, y) = P, \qquad \Delta\eta(x, y) = Q. \tag{1}$$

Then the inverse maps $x(\xi, \eta)$ and $y(\xi, \eta)$ satisfy

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2 x_\xi P + J^2 x_\eta Q = 0, \tag{2}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2 y_\xi P + J^2 y_\eta Q = 0, \tag{3}$$

where

$$J := x_\xi y_\eta - y_\xi x_\eta \tag{4}$$

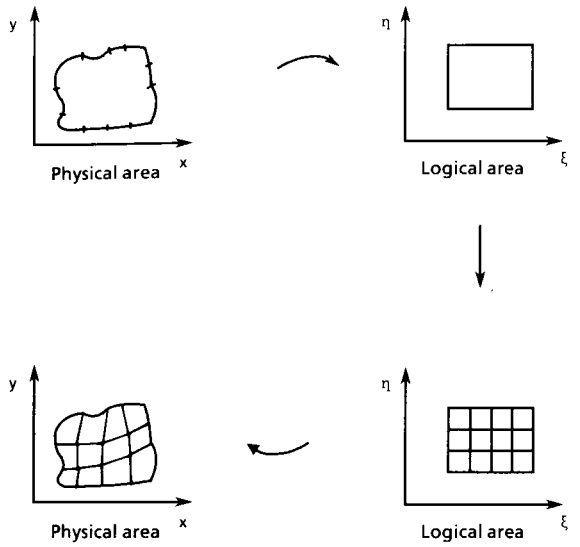


FIG. 3. Mesh concept of boundary-fitted coordinates.

and

$$\alpha := x_\eta^2 + y_\eta^2, \tag{5}$$

$$\beta := x_\xi x_\eta + y_\xi y_\eta, \tag{6}$$

$$\gamma := x_\xi^2 + y_\xi^2. \tag{7}$$

If the boundary points are specified as Dirichlet points, which means that they are known and fixed, the interior points are uniquely determined by Eqs. (2) and (3).

For the solution of the finite difference equations, the mesh data are stored in ordered arrays of numbers $(x_{i,j}, y_{i,j})$ in which the indices $i = 1, ..., imax, j = 1, ..., jmax$ give also the physical relationship between the mesh point $(x_{i,j}, y_{i,j})$ and another point $(x_{i',j'}, y_{i',j'})$. E.g., the four neighbors of the mesh point $(x_{i,j}, y_{i,j})$ are $(x_{i+1,j}, y_{i+1,j})$, $(x_{i,j+1}, y_{i,j+1})$, $(x_{i-1,j}, y_{i-1,j})$, and $(x_{i,j-1}, y_{i,j-1})$.

Hence, solving the discretized Dirichlet problem leads to the grid points $(x_{i,j}, y_{i,j})$ in the physical space. With the functions P and Q the density of interior lines can be controlled. Attraction and repulsion of grid lines towards specified points and lines can be achieved. Usually $P = 0$ and $Q = 0$ are chosen, especially when the grid is to be modified within a simulation.

The advantages of elliptic grid generator systems are that the resulting grids are inherently smooth, the danger of overlapping of grid lines is small, and they can easily be adapted to general boundary configurations.

3.3. *Numerical Procedure*

In this section we will summarize the numerical procedure to solve the governing equations of the grid generator. The standard approach of discretizing Eqs. (2) and (3) is to use central differences, accurate in second order with respect to the logical space. We will present the procedure only for Eq. (2); the discretization of Eq. (3) is quite the same. Using the following difference schemes for representing the first-order derivatives,

$$x_\xi = (x_{i+1,j} - x_{i-1,j})/(2\Delta\xi), \tag{8}$$

$$x_\eta = (x_{i,j+1} - x_{i,j-1})/(2\Delta\eta), \tag{9}$$

and for the second-order derivatives,

$$x_{\xi\xi} = (x_{i+1,j} - 2x_{i,j} + x_{i-1,j})/(\Delta\xi)^2, \tag{10}$$

$$x_{\eta\eta} = (x_{i,j+1} - 2x_{i,j} + x_{i,j-1})/(\Delta\eta)^2, \tag{11}$$

Eq. (2) can easily be evaluated.

In the case of $P = Q = 0$, $\Delta\xi$ and $\Delta\eta$ are eliminated in Eqs. (2) and (3). For this case, the iteration process for point $(x_{i,j}, y_{i,j})$ reduces to

$$x_\xi = \tfrac{1}{2}(x_{i+1,j} - x_{i-1,j}), \tag{12}$$

$$x_\eta = \tfrac{1}{2}(x_{i,j+1} - x_{i,j-1}), \tag{13}$$

$$\alpha = x_\eta^2 + y_\eta^2, \tag{14}$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta, \tag{15}$$

$$\gamma = x_\xi^2 + y_\xi^2, \tag{16}$$

and

$$x_{i,j} = \frac{1}{2(\alpha+\gamma)}\left(\alpha(x_{i+1,j} + x_{i-1,j}) + \gamma(x_{i,j+1} + x_{i,j-1}) - \frac{1}{2}\beta(x_{i+1,j+1} - x_{i+1,j-1} - x_{i-1,j+1} - x_{i-1,j-1})\right), \tag{17}$$

$$y_{i,j} = \frac{1}{2(\alpha+\gamma)}\left(\alpha(y_{i+1,j} + y_{i-1,j}) + \gamma(y_{i,j+1} + y_{i,j-1}) - \frac{1}{2}\beta(y_{i+1,j+1} - y_{i+1,j-1} - y_{i-1,j+1} - y_{i-1,j-1})\right). \tag{18}$$

A local relaxation scheme (ad-hoc SOR) [19] is applied for the iterative procedure in solving Eq. (17) and (18).

3.4. *Example*

The numerical work presented in this paper treats the externally applied-B diode in a realistic manner by taking into account the curved parts of the diode without simplifying the geometry. Figure 4a shows a boundary-fitted grid designed for the B_{appl}-diode. As can be seen, with the assistance of a logical grid (cf. Fig. 4b), a boundary-fitted

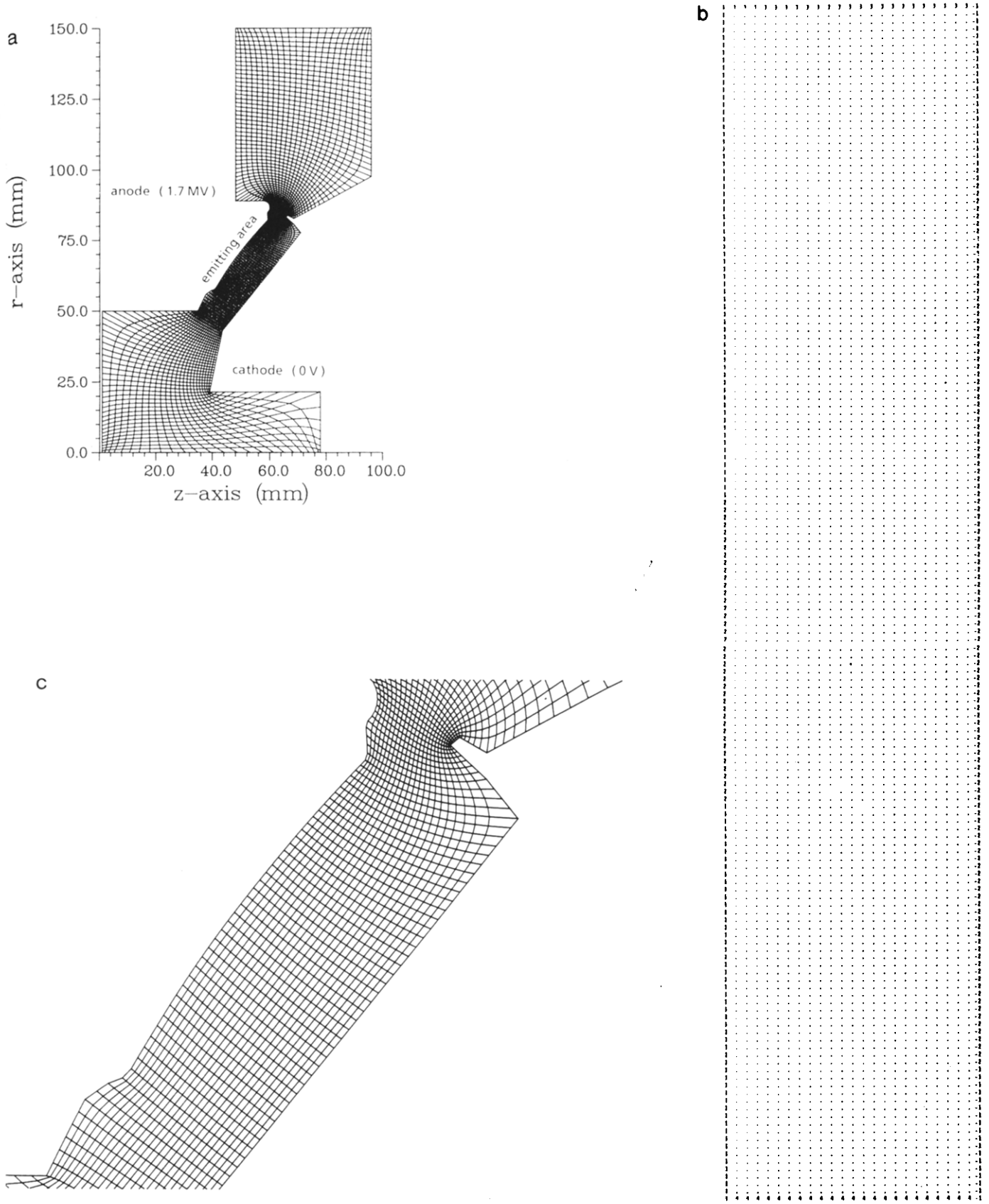


FIG. 4. Numerical model of the externally applied-B ion diode with a (25×159)-boundary-fitted grid: (a) overall physical grid; (b) overall logical grid; (c) detail of the region near the cathode tip.

grid is introduced in the anode–cathode area with grid lines accurately fitted to the curved boundaries. Furthermore, with the introduction of point attributes, boundary conditions can easily be assigned to the grid points.

Dirichlet conditions are carried over directly to the corresponding boundary in the logical domain, and in the case of Neumann conditions the derivatives normal to the boundary are transformed by

$$f_n = (\alpha f_\xi - \beta f_\eta)/(J\sqrt{\alpha}) \quad \text{on lines with constant} \quad \xi = \xi_0, \tag{19}$$

$$f_n = (\gamma f_\eta - \beta f_\xi)/(J\sqrt{\gamma}) \quad \text{on lines with constant} \quad \eta = \eta_0. \tag{20}$$

The recessed edge of the cathode ring is difficult to treat numerically. Amplified errors occur at such edges. Therefore, the grid was made particularly fine-meshed in the region of the cathode ring, as indicated in more detail in Fig. 4c.

Due to the use of boundary-fitted coordinates the curved parts of the emitting anode surface are treated accurately. No interpolation from the physical domain onto the computational area is necessary and a stair-step approximation of the curved parts is avoided. Modifications of the geometry such as the consideration of the movement of the anode and cathode plasmas are possible.

4. THE BFCPIC CODE

The simulations presented in this paper were performed using the BFCPIC code (particle-in-cell code based on boundary-fitted-coordinates) [7, 11]. BFCPIC is a $2\frac{1}{2}$-dimensional, electro- and magnetostatic particle-in-cell code in (z, r)-geometry. In order to treat complicated devices without simplification of the geometry BFCPIC was developed using grids fitted to the boundary of the physical domain.

In order to self-consistently model complex electromagnetic systems, two numerical methods have been coupled: boundary-fitted coordinates are introduced to be able to treat technical geometries and the particle-in-cell method is used to describe the orbits of the electrically charged particles in electromagnetic fields in a self-consistent manner. On the discrete boundary-fitted grid the field quantities are computed by solving the stationary Maxwell equations, whereas the particles are advanced in the physical space. New algorithms describing the transition between the grid model and the mesh-free model, interpolation, and localization, were developed [20, 21]. Furthermore, the existing techniques for equidistant grids were extended to the case of non-equidistant, arbitrarily shaped, convex, four-point meshes. A detailed description of the models and algorithms used and implemented in BFCPIC can be found in [11]. One should be aware of the fact that boundary-fitted coordinates increase the complexity of the algorithms.

By applying boundary-fitted coordinates as the grid concept underlying the simulations, the shapes of the anode and cathode surfaces are modelled quantitatively, and thus the simulation of the emission of particles on highly stressed electrodes is treated accurately. For the field emission of the electrically charged particles a macroscopic model is used. If the normal electric field exceeds a threshold field of about 10^7 V/m, a dense plasma quickly forms on the conducting surfaces. Instantaneously, particles are emitted from the outer surface of the plasma. The amount of charge generated at the plasma surface is space-charge limited: Particles are created in such a way that, in addition to the tangential component, also the normal component of the electric field vanishes at the plasma surface. The latter is regarded as a conductor with infinite conductivity.

Depending on the type of particles, the relativistic or non-relativistic equations of motion are solved. To compensate the much smaller time scale required for electrons compared with ions, two different time scales are introduced in order to advance the electrons several smaller time steps. Field calculations are carried out on the larger ion time scale.

In order to eliminate the influence of fluctuations arising from the coarseness of the grid, the discretization in time, and the limited number of particles, special techniques are used: For the simulation of ion diodes the charge and current densities are relaxed at time step n by

$$\rho_{ij}^n \leftarrow (1 - \omega(n))\, \rho_{i,j}^{n-1} + \omega(n)\, \rho_{ij}^n, \tag{21}$$

where

$$\omega(n) = \max\left(\frac{1}{100}, \frac{1}{n}\right). \tag{22}$$

The simulations are split into two parts: The first run consists of 2000 time steps in which the voltage linearly rises to its maximum value within the first 1500 time steps. In the succeeding run the densities are relaxed.

5. MOVING BOUNDARIES

5.1. *Physical Model*

To accelerate the ion beam into a point-like focus on the axis the ions must have a vanishing canonical angular momentum. Since the field coils induce a magnetic field which is zero on the axis, the ions have to be generated on an equipotential surface with vanishing angular momentum. Hence, the external magnetic field is designed in such a way that the anode surface coincides with such an equipotential surface.

Although this is realized in the experimental device, measurements [17] show that the momentum of the ions on the axis is much larger than expected. This can be explained by an expansion of the anode plasma by 1.5–2 mm and by the fact that the magnetic flux penetrates into the plasma surface. Later in the pulse these plasmas become surfaces with infinite conductivity and the magnetic field cannot penetrate any further. The position of the emitting plasma surface is then defined by pressure balance:

Due to the externally applied-B field the movement of the particles is not only in the (z, r)-plane but the velocity of the particles has also a non-vanishing azimuthal component. This θ-component of the velocity induces a self-generated vector potential via the equation

$$\delta_z^2(rA_\theta) + \delta_r^2(rA_\theta) - \frac{1}{r}\delta_r(rA_\theta) = -\mu_0 r j_\theta. \tag{23}$$

When solving this equation it is assumed that the boundary condition for the self-generated fraction of the vector potential A_θ is zero on the conducting surfaces. This is reasonable, since during the 50–100 ns pulse, the self-magnetic field cannot penetrate into conducting surfaces as well as into plasmas. Since the magnetic field is the curl of the vector potential,

$$B_r = -\frac{1}{r}\delta_r(rA_\theta), \qquad B_z = \frac{1}{r}\delta_z(rA_\theta), \tag{24}$$

it follows that the total magnetic field, externally applied and self-generated, increases at the anode and decreases at the cathode. As a consequence, the anode surface is shifted towards the geometric anode and the cathode surface is shifted away from the geometric cathode side. This mechanism proceeds until a stationary balance is achieved.

In summary, due to the rising current inside the anode–cathode gap, a magnetic field (B_r, B_z) is induced by this current and the anode surface is determined by the position of the constant magnetic flux line $\psi_a = rA_\theta$ (anode) and the cathode surface by $\psi_c = rA_\theta$ (cathode). This means that the plasma surfaces remain equipotential surfaces and the positions depend dynamically on the currents inside the diode. Hence, the anode and cathode plasma surfaces expand during the pulse. The shape of the anode surface determines the focusing properties of the ion beam. In order to obtain a realistic description of this characteristic feature of the diode, the position of the emitting anode and cathode surfaces must be taken into account self-consistently.

Hence, in the numerical simulation the position of the equipotential lines ψ_a and ψ_c for the anode and cathode, respectively, define the positions of the emitting surfaces and have to be determined. Due to time-dependent effects, the currents in the diode change and, therefore, also the magnetic field. Due to the change in the magnetic field, the emission surfaces have to be adapted in a proper way.

5.2. *Starting Conditions*

The basic idea of our numerical model is that we do not simulate the expansion of the plasma across the field lines during the rise time of the voltage. But we assume that after an appropriate time the anode plasma has expanded into the gap by 2 mm and that the magnetic flux has penetrated into this plasma surface. Furthermore, we assume that a plasma has been established at the cathode tip and has moved by fast expansion along the ψ_c-line [22].

In order to realize this scenario we start with a numerical grid for the B_{appl}-diode without inner plasma surfaces and with the geometrical dimensions as shown in Fig. 4a. Assuming an expansion of the anode and cathode plasmas by 2 mm and 1 mm into the gap, respectively, we obtain a shifted anode surface and cathode tip.

The corresponding computational grid was obtained by first looking for the grid points which are closest to this shifted virtual anode line inside the diode region. For the sake of simplicity we chose a line $\xi_a = \text{constant}$ for the representation of the new emitting anode. The inner anode points are assigned new point attributes, indicating that these grid points now belong to the anode plasma. The same procedure holds for the cathode tip.

Furthermore, in order to realize the fast expansion of the

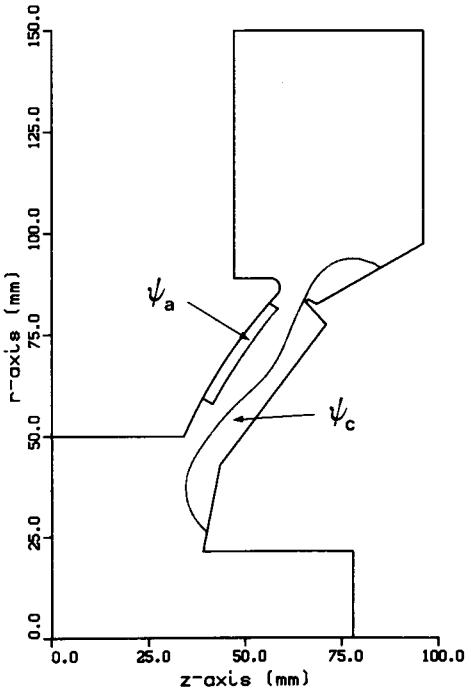


FIG. 5. Plasma boundaries at the beginning of the run. ψ_a and ψ_c represent the anode and cathode plasmas, respectively.

cathode tip plasma along the magnetic field lines, we determine the line of constant magnetic flux ψ_c shown in Fig. 5. Once again the nearest grid points are shifted towards this line representing this boundary as a ξ_c = constant line in the emitting regime.

ψ_c cannot be represented in the whole computational domain by one ξ_c-line. A stepwise representation of ψ_c connects the interior ξ_c-line with the geometric boundary that can be recognized in the logical grid in Fig. 6b.

After having defined the interior boundary points, the grid points between the geometric boundaries and plasma surfaces were assigned plasma point attributes. The outer lines represent the geometric surfaces and the inner lines represent the plasma surfaces after the expansion. Figure 6 shows the corresponding boundary-fitted grid (a), together with the logical grid (b). One identifies three different types of grid points in the logical structure: The grid points indicated by a dot are interior grid points where the field equations are solved. The grid points marked with "5" and "−2" are boundary points on the moving plasma surfaces and the points marked with "*" and "×" are points inside the plasmas.

From the particle data the charge and current densities can be computed and in principle the fields can be actualized.

However, this is an enormous computational effort because not only the densities but the whole field computations as well must be repeated. This means that the total CPU-time for a simulation with moving boundaries is nearly doubled. Hence, it is not convenient to update the grid and field data in each time step, especially at the beginning of the simulation where the currents and induced magnetic fields are small and there is only a little change in the position of the plasma surface.

Therefore, we developed a strategy to avoid unnecessary work by moving the grid only after some hundred time steps, depending on the shift of the boundaries. We calculate the actual position of the plasma boundaries in each time step and decide whether or not to replace the grid and repeat the calculation of the fields on the new grid.

It turned out that, due to fluctuations in the particle quantities, the position of the plasma boundary in general is not a smooth curve but varies from time step to time step, noisily depending on the actual particle distribution. In order to avoid this statistical fluctuation, the positions of

5.3. *Dynamic Movement of the Boundaries*

As the self-generated current induced by the movement of particles increases, the self-magnetic field (B_r, B_z) increases as well. Hence, the position of the equipotential lines ψ_c and ψ_a at the cathode and anode move, respectively.

In the first step the new positions of these two lines in the current grid have to be identified. When these positions are known the surface grid points are shifted onto the corresponding equipotential lines. These surface grid points now represent the inner boundary of the new grid. With the estimate of the old grid, a grid relaxation is performed and as the result of the iteration process the new grid is calculated.

the surface points are relaxed at time step n by

$$\mathbf{r}^n_{\text{boundary}} \leftarrow (1-\omega(n))\,\mathbf{r}^{n-1}_{\text{boundary}} + \omega(n)\,\mathbf{r}^n_{\text{boundary}}, \quad (25)$$

where

$$\omega(n) = \max\left(\frac{1}{100}, \frac{1}{n}\right). \quad (26)$$

With this specific choice of the relaxation parameter $\omega(n)$ a smooth shape of the surfaces with little fluctuation is achieved.

5.5. *Particle Transformation*

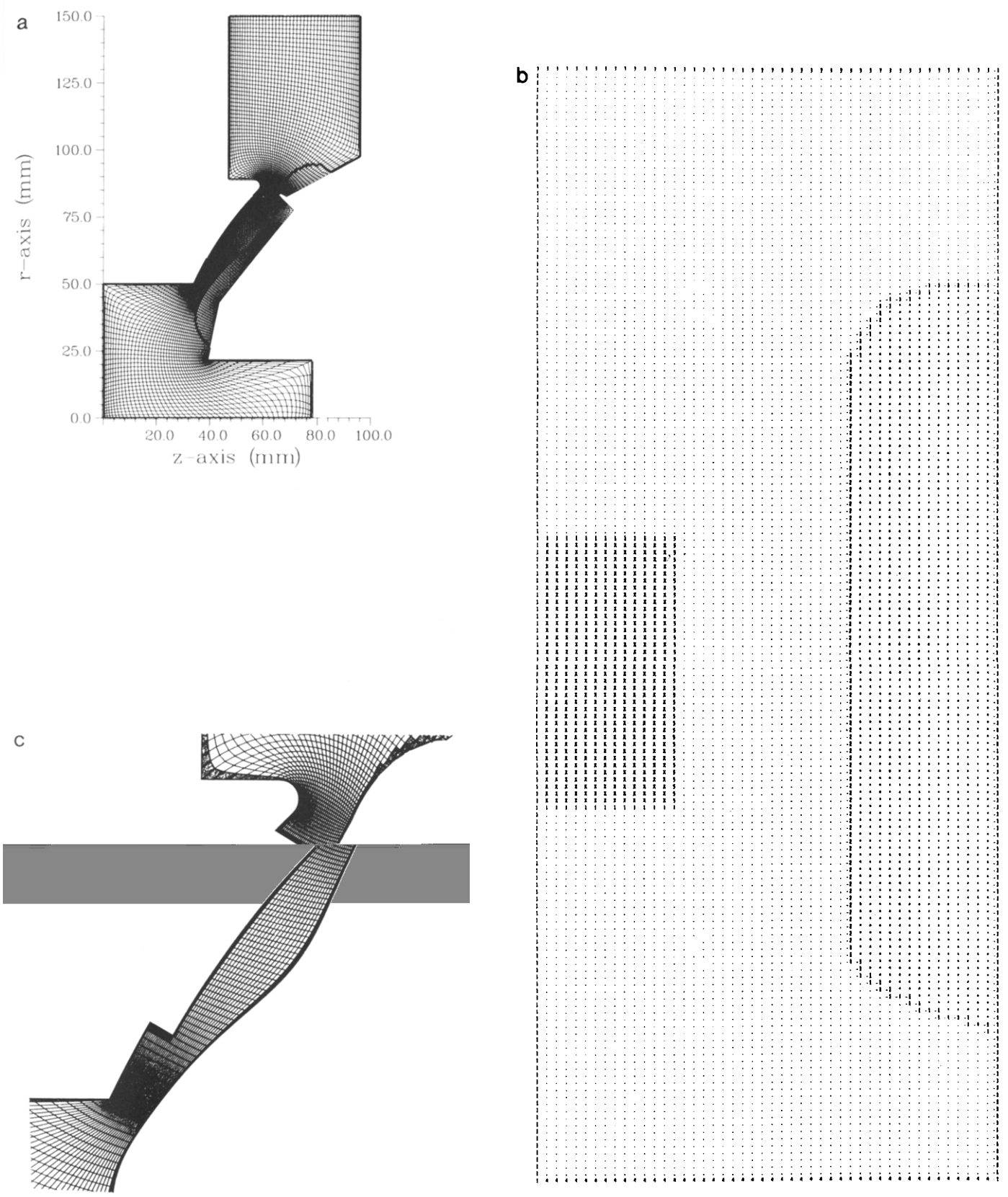


FIG. 6. (48 × 159)-boundary-fitted grid for the externally applied-B ion diode for a run with moving plasma boundaries: (a) overall physical grid with plasma grid points; (b) overall logical grid; (c) detail of the region near the cathode tip with computational points only.

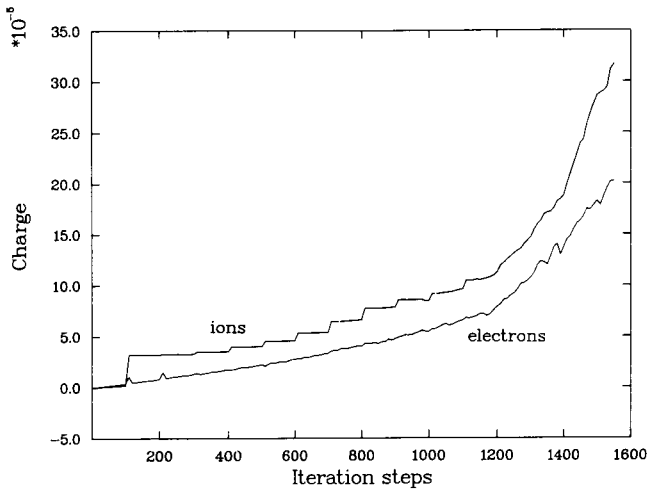


FIG. 7. Emitted charge per time step during a simulation without the transformation of the particles.

charge, than during previous time steps, is emitted. This leads to a stepwise evolution of the charge in Fig. 7, producing artificial oscillations in the solution of the fields.

To overcome this problem, the particle data also have to be reorganized with respect to the new grid. It is not adequate to localize the particles with their (z, r)-coordinates inside the new grid, but the particles must be rescaled in the physical grid. As a consequence, after the reorganization of the particles' coordinates all particles are inside the modified grid.

The basic idea for this procedure lies in assigning weights (w_1, w_2) to each particle as the associated coordinates in the logical grid [20] before the grid is adapted. These weights are responsible for the interpolation of the fields onto the particle position: The integer fractions of (w_1, w_2),

$$I = \mathrm{Int}(w_1) \quad \text{and} \quad J = \mathrm{Int}(w_2), \tag{27}$$

indicate the address of the cell where the particle is located, and the decimal fractions,

$$\alpha_1 = w_1 - I \quad \text{and} \quad \alpha_2 = w_2 - J, \tag{28}$$

indicate the location inside this cell. According to the area weighting method a field f defined on the grid points is interpolated onto the particle position by

$$\begin{aligned} f = {} & (1-\alpha_1)(1-\alpha_2)\, f(I, J) \\ & + (1-\alpha_1)\, \alpha_2 f(I, J+1) \\ & + \alpha_1 (1-\alpha_2)\, f(I+1, J) \\ & + \alpha_1 \alpha_2 f(I+1, J+1). \end{aligned} \tag{29}$$

These associated coordinates are conserved and used to transform the particle coordinates to the new grid. Hence, we assume that the associated coordinates do not change—only the physical ones do, according to

$$\begin{aligned} z = {} & (1-\alpha_1)(1-\alpha_2)\, z^{\mathrm{new}}(I, J) \\ & + (1-\alpha_1)\, \alpha_2 z^{\mathrm{new}}(I, J+1) \\ & + \alpha_1 (1-\alpha_2)\, z^{\mathrm{new}}(I+1, J) \\ & + \alpha_1 \alpha_2 z^{\mathrm{new}}(I+1, J+1). \end{aligned} \tag{30}$$

The same equation holds for the r-coordinate. By applying this transformation on the particles' coordinates all particles located inside the gap before the movement of the grid remain inside the computational domain afterwards. Moreover, it is obvious that the profile of the particle distribution is quite similar and the shape has not changed.

With this procedure we have smoothed the charge emission and therefore also the artificial oscillations in the fields. However, by considering a plane condenser as a test example there still remain small oscillations and peaks of the emitted current after the movement of the surfaces. The reason for this behavior can be explained by the fact that, although the particles are transformed correctly from the old grid to the new one and although the charge profile is correct, the absolute value of the density is incorrect. In addition to the particles' coordinates we also have to rescale the charge per particle to preserve, e.g., the Child–Langmuir law for monopolar particle flow in the x-direction.

By applying a scaling law for the charge in the gap, according to

$$q_{\mathrm{new}}(x) = \frac{d_{\mathrm{old}}}{d_{\mathrm{new}}}\, q_{\mathrm{old}}\left(x \frac{d_{\mathrm{old}}}{d_{\mathrm{new}}}\right), \tag{31}$$

a Child–Langmuir law in the new computational domain with gap distance d_{new} is obtained if such a law in the old domain with gap width d_{old} is valid.

As a consequence, after the transformation of the particles' (z, r)-coordinates also the charge per particle has to be rescaled and multiplied by the factor $d_{\mathrm{old}}/d_{\mathrm{new}}$.

In summary, in order to obtain a smooth transition of the particle and field data within a simulation with moving plasma boundaries, not only a relaxation scheme for the movement of the boundaries has to be applied but also the particles have to be transformed into the new grid and the charge per particle has to be rescaled.

5.6. *Iteration Scheme*

The following algorithm indicates the flow diagram of the extended BFCPIC program. Steps 1–5 are well-known, standard modules of a PIC code. Steps 6–12 were added to

the standard cycle in order to permit the movement of the emitting plasma surfaces.

The iteration cycle starts after a preparatory phase which consists of the input of the computational grid, of identifying the emitting electrode cells, and of determining the volumes of the cells, the input of already existing particles, and calculating the externally applied fields as well as the self-fields. Steps 1–6 are performed in each time step. In Step 6 the position of the plasma boundaries with respect to the current vector potential is determined. After deciding whether or not to change the boundary surfaces, the iteration proceeds either by moving the plasma surfaces and continuing with Steps 7–12 or by emitting new particles (Step 1).

The extended cycle involves the following steps:

1. Generating new particles at the plasma boundaries according to the fields on the emitting cells.
2. Interpolating the fields onto the particle position.
3. Advancing particle positions and velocities consistent with the electromagnetic fields.
4. Depositing particle charge and current densities associated with each particle to the grid.
5. Solving the field equations on the grid.
6. Determining the position of the plasma boundaries with respect to the vector potential and relaxing the grid points representing the plasma surfaces.

If the change in the position of the surface points extends a certain threshold the cycle continues with Step 7; otherwise with Step 1.

7. Shifting the boundary points to the new position of the vector potential ψ_c and ψ_a, respectively.
8. Relaxing the grid points with respect to the modified surfaces.
9. Computing the new volumes of the cells.
10. Transforming the particle coordinates onto the new grid and rescaling the charge of the particles.
11. Depositing particle charge and current densities associated with each particle to the new grid.
12. Solving the field equations on this grid.

This iteration cycle proceeds until a stationary profile of the boundaries, as well as of the fields, is achieved.

6. NUMERICAL SIMULATIONS

For the simulation of the externally applied-B ion diode an applied voltage of $\phi = 1.7$ MV is assumed. The strength of the applied-B field in the gap varies between 1.7 and 2 T. A geometric anode–cathode gap of $d = 8.75$ mm is chosen. In order to take the plasma movement into account the emitting anode surface is shifted by 2 mm and the cathode tip by 1 mm into the gap, reducing the gap width to 5.75 mm. Thereby, magnetic diffusion in these surface plasmas is assumed.

For the numerical simulation of the B_{appl}-diode the time step is set to $dt = 10^{-12}$ s. As ions we use protons. Over 50,000 electrons and ions are emitted from the cathode and anode surfaces, respectively, requiring about 2–4 h of CPU-time on a Fujitsu VP400 EX.

6.1. *Simulations without Boundary Movement*

In Fig. 8 the distribution of electrons inside the anode–cathode gap is plotted. Although electrons are allowed to be emitted all over the cathode, most of them are generated at the cathode tip. These electrons move radially inside the gap and form an electron sheath between the anode and the geometric cathode. Near the symmetry axis they are reflected by the increasing magnetic field. A small amount of kinetic energy is directed in θ-direction (see Fig. 2b). Hence, the current density j_θ induced by this movement produces only a small self-generated component of the vector potential A_θ. The particles stay on lines $\psi_c = r * A_\theta$ at the cathode which is dominated by the externally applied-B field.

In order to determine the focusing properties, ions are generated at the anode surface and accelerated in the anode–cathode gap. When leaving the diode through the cathode the positions and the velocities of the particles are known.

When they leave the diode region they enter the so-called drift section (cf. Fig. 1). The drift section is filled with a background gas (5-mb Ar) and separated from the diode region by a mylar foil. The electrons of the ionized back-

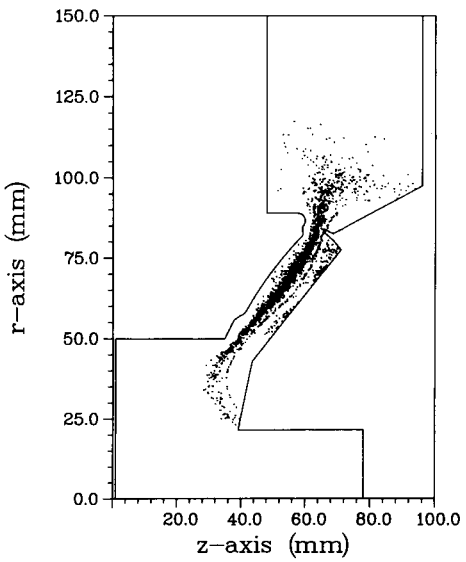


FIG. 8. Particle map of electrons at steady state from simulation without boundary movement; 2055 out of 55,600 electrons are shown.

ground gas fully compensate the positive space charge of the ion beam. Due to the external magnetic field the electrons are prevented from directly following the ions. Hence, the ion beam is charge, but not current, neutralized. The current neutralization is only about 90 to 95%. Although the magnetic field is primarily determined by the externally applied-B field, the rest current, 5 to 10% of the original 500-kA ion current, is not negligible. This uncompensated ion current leads to a B_θ-eigenfield which must be taken into account.

Our main interest, however, lies in the qualitative effect of the focusing properties, without and with plasma movement, and in the differences therein and not in the details of the physics in the drift section. Therefore, we neglect these magnetic fields and assume a force-free movement of the ions. (The effects of the magnetic fields principally can be taken into account by the simulation program DRIFTPIC [23]. But the general statements presented with this simplified model remain valid.)

Assuming a force-free movement of the ions in the drift section the points of intersection of the trajectories and the symmetry axis are computed. In Fig. 9 some sample orbits

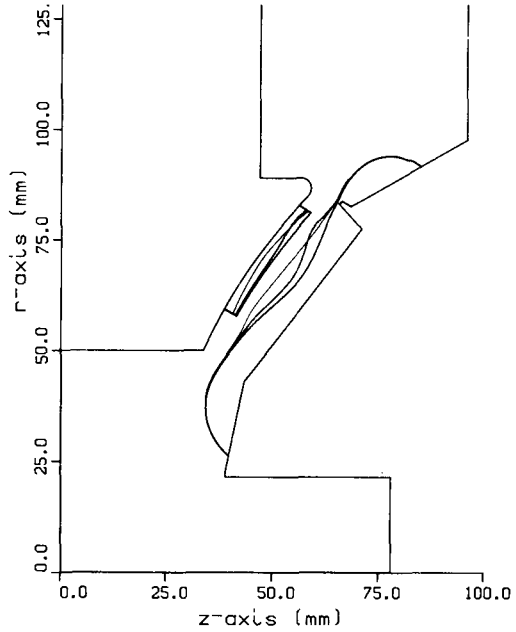
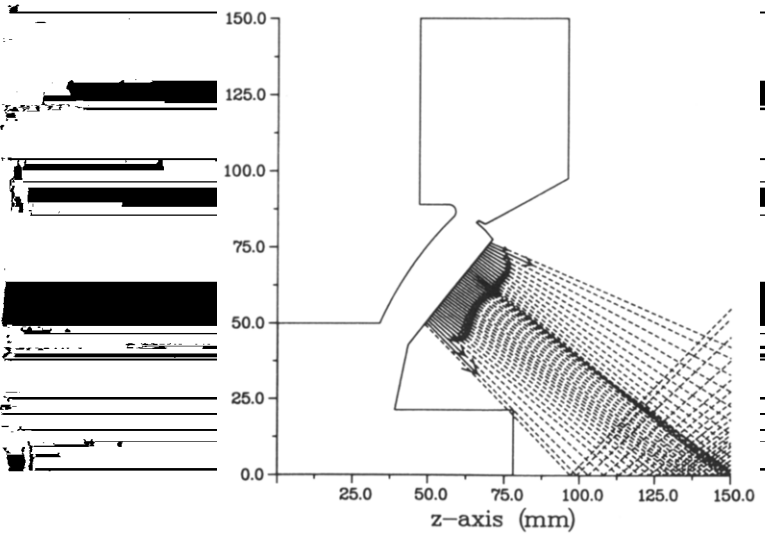
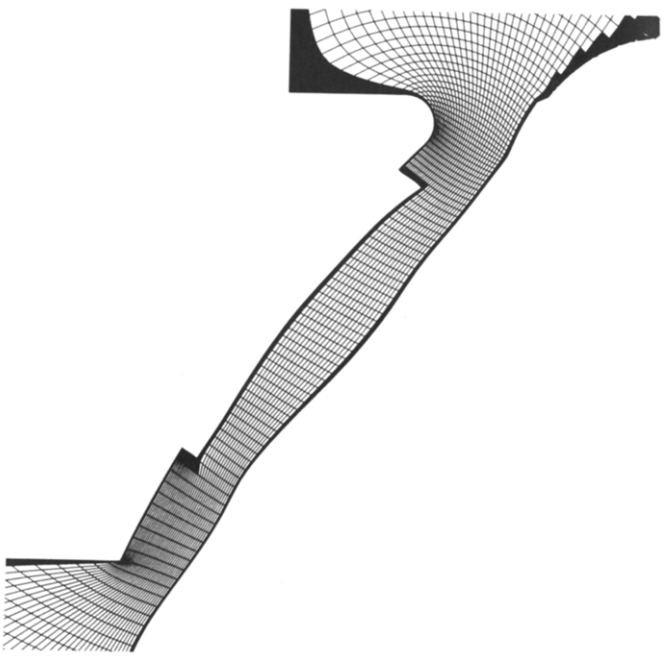


FIG. 10. Positions of the emitting anode and cathode surfaces represented by lines of constant magnetic flux ψ_1 and ψ_2 (zoom of the





the interior part of the emitting anode is shifted towards the geometric anode. As a consequence, the curvature of the emitting anode surface is larger, compared with the geometric surface, and the focusing effect on the ions is increased. Looking at the focusing properties of the ion beam (see Fig. 12) the ions intersect above the axis and a ring focus develops above the axis. This is exactly the phenomenon observed in the experiments where the measured ion trajectories have been evaluated, as discussed in Ref. [17].

Comparing the current density of the ions as a function of the radius it can be seen that, for the simulation without plasma movement, the highest amplitude is reached at the edges. In contrast to these results, the simulations with plasma boundary movement show amplified densities at the inner part of the cathode, where the ions leave the diode.

The reason for this effect is that the vector potential induced by the current of particles has the highest curl in the middle part of the diode region. Hence, the gap becomes

tion performed with the $2\frac{1}{2}$-dimensional PIC code BFCPIC. Since the grid model underlying BFCPIC is that of boundary-fitted coordinates, it is possible to treat complicated technical devices without simplifying the geometry. In particular, the shapes of the anode and cathode surfaces are modelled quantitatively and thus the simulation of the emission of particles on highly stressed electrodes is treated accurately.

However, it turned out that in pulsed power ion diodes the plasma surfaces are determined by the dynamical behavior of the charged particles. The anode surface is defined by the position of constant magnetic flux $\psi_a = rA_\theta$ (anode) and the cathode surface by $\psi_c = rA_\theta$ (cathode). If the current rises, a magnetic field is induced and the anode, as well as the cathode surface, move on the surfaces with constant magnetic flux.

New techniques were introduced in order to dynamically describe the expansion the emitting surfaces within a simulation. Subsequently, the computational grid model

smallest at this part and the current density of ions increases there (cf. Fig. 12).

In summary, it has been demonstrated that characteristic features of the externally applied-B diode can be reproduced by simulation if the movement of the emission surfaces is taken into account self-consistently. When correcting the plasma boundary according to the dynamical behavior of the currents the focusing properties, as well as the shape

was extended to react on the different shapes of the anode and cathode plasmas during a simulation. Hence, the discussion was focused on the problem arising with moving boundaries and adaptive meshes.

New numerical methods handling the reorganization of the grid, the transformation, and the rescaling of the particles, and the recalculation of the field data are presented. As outlined, in order to obtain a smooth transition of the

```
C     I2MX  (I)   :  ACTUAL GRID SIZE IN 2-DIRECTION
C     ICONV (O)   :  INDICATES THE CONVERGENCE: 0 = FAILED
C                                               1 = TRUE
C
C INTERNAL PARAMETERS
C
C     ITERMX :  MAXIMAL NUMBER OF ITERATIONS
C     DIFFMX :  MAXIMAL ALLOWED DIFFERENCE OF SUCCEEDING ITERATIONS
C     DEFTMX :  MAXIMAL ALLOWED RESIDUUM VALUE
C
C


      REAL     X1(NDIM1,NDIM2),X2(NDIM1,NDIM2)
      LOGICAL  IGEW(NDIM1,NDIM2)

      DATA PI                    /3.141592653/
      DATA ITERMX,DIFFMX,DEFTMX  /1000,1.5E-6,1.5E-6/

      ICONV = 0

C---- ITERATION LOOP
C
      DO 100 ITER=1,ITERMX
         DXMX = 0.0

C***  COMPUTATION OF THE NEW COORDINATES OF THE GRID POINTS
C
         DO 50 I1=2,I1MX-1
            I1P1 = I1 + 1
            I1M1 = I1 - 1
            DO 50 I2=2,I2MX-1
            IF(IGEW(I1,I2)) THEN
            I2P1 = I2 + 1
            I2M1 = I2 - 1
            X1XI1 = 0.5*(X1(I1P1,I2)-X1(I1M1,I2))
            X1XI2 = 0.5*(X1(I1,I2P1)-X1(I1,I2M1))
            X2XI1 = 0.5*(X2(I1P1,I2)-X2(I1M1,I2))
            X2XI2 = 0.5*(X2(I1,I2P1)-X2(I1,I2M1))
            ALPHA = X1XI2*X1XI2 + X2XI2*X2XI2
            BETA  = X1XI1*X1XI2 + X2XI1*X2XI2
            GAMMA = X1XI1*X1XI1 + X2XI1*X2XI1

            C     = -2.0*(ALPHA+GAMMA)
            AMU   = -2.0/C *  (  ALPHA * COS(PI/(I1MX-1))
     .                         + GAMMA * COS(PI/(I2mx-1)) )
            OMEGA =  2.0/(1.0+SQRT(1.0-AMU*AMU))

            C1 = -ALPHA/C
            C2 =  BETA/(2.0*C)
            C3 = -GAMMA/C

            X1N =     C1*(X1(I1P1,I2)+X1(I1M1,I2))
     .            +  C2*(X1(I1P1,I2P1)-X1(I1M1,I2P1)
     .                   -X1(I1P1,I2M1)+X1(I1M1,I2M1))
     .            +  C3*(X1(I1,I2P1)+X1(I1,I2M1))
            X2N =     C1*(X2(I1P1,I2)+X2(I1M1,I2))
     .            +  C2*(X2(I1P1,I2P1)-X2(I1M1,I2P1)
     .                   -X2(I1P1,I2M1)+X2(I1M1,I2M1))
     .            +  C3*(X2(I1,I2P1)+X2(I1,I2M1))

            X1N = (1.0-OMEGA)*X1(I1,I2) + OMEGA*X1N
            X2N = (1.0-OMEGA)*X2(I1,I2) + OMEGA*X2N
            DXMX = MAX(ABS(X1(I1,I2)-X1N),ABS(X2(I1,I2)-X2N),DXMX)

            X1(I1,I2) = X1N
            X2(I1,I2) = X2N

            ENDIF
   50    CONTINUE

C***  COMPUTATION OF THE RESIDUAL VALUES
C
         RSX1N = 0.0
         RSX2N = 0.0
         DO 70 I1=2,I1MX-1
            I1P1 = I1 + 1
            I1M1 = I1 - 1
            DO 70 I2=2,I2MX-1
            IF(IGEW(I1,I2)) THEN
            I2P1 = I2 + 1
            I2M1 = I2 - 1
            X1XI1 = 0.5*(X1(I1P1,I2)-X1(I1M1,I2))
            X1XI2 = 0.5*(X1(I1,I2P1)-X1(I1,I2M1))
            X2XI1 = 0.5*(X2(I1P1,I2)-X2(I1M1,I2))
            X2XI2 = 0.5*(X2(I1,I2P1)-X2(I1,I2M1))
            ALPHA = X1XI2*X1XI2 + X2XI2*X2XI2
            BETA  = X1XI1*X1XI2 + X2XI1*X2XI2
            GAMMA = X1XI1*X1XI1 + X2XI1*X2XI1
            C     = -BETA*0.5

            RSX1 = ALPHA * (X1(I1P1,I2)-2.0*X1(I1,I2)+X1(I1M1,I2))
     .               +C * (X1(I1P1,I2P1)-X1(I1P1,I2M1)
     .                                  -X1(I1M1,I2P1)+X1(I1M1,I2M1))
     .           +GAMMA * (X1(I1,I2P1)-2.0*X1(I1,I2)+X1(I1,I2M1))
            RSX2 = ALPHA * (X2(I1P1,I2)-2.0*X2(I1,I2)+X2(I1M1,I2))
     .               +C * (X2(I1P1,I2P1)-X2(I1P1,I2M1)
     .                                  -X2(I1M1,I2P1)+X2(I1M1,I2M1))
     .           +GAMMA * (X2(I1,I2P1)-2.0*X2(I1,I2)+X2(I1,I2M1))

            RSX1N = MAX(RSX1N,ABS(RSX1))
            RSX2N = MAX(RSX2N,ABS(RSX2))

            ENDIF
   70    CONTINUE

         IF(DXMX.LT.DIFFMX) GOTO 200
         IF(RSX1N.LT.DEFTMX.AND.RSX2N.LT.DEFTMX) GOTO 200
  100 CONTINUE

      ITER = ITERMX

C***  NO CONVERGENCE
      ICONV = 0
      RETURN

C***  CONVERGENCE
  200 ICONV = 1
      RETURN

      END
```

ACKNOWLEDGMENTS

The author acknowledges the stimulating discussions with V. Fedorov and W. Schmidt. He deeply regrets the demise of W. Schmidt, who significantly supported this work. Thanks also to K. May for the programming of the new modules and D. Seldner for carefully reading the manuscript.

REFERENCES

1. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
2. C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw–Hill, New York, 1985).
3. J. P. Quintenz, *J. Appl. Phys.* **49**, 4377 (1978).
4. D. B. Seidel, M. L. Kiefer, R. S. Coats, T. D. Pointon, J. P. Quintenz, and W. A. Johnson, in *Proceedings, Europhysics Conference on Computational Physics, Amsterdam, Sept. 10–13, 1990*, edited by A. Tenner (World Scientific, Singapore, 1990), p. 475.
5. A. Mankofsky, J. L. Seftor, C. L. Chang, K. Ko, A. A. Mondelli, A. T. Drobot, J. Moura, W. Aimonetti, S. T. Brandon, D. E. Nielsen, Jr., and K. M. Dyer, *Comput. Phys. Commun.* **48**, 155 (1988).
6. M. E. Jones, in *Proceedings, 12th Int. Conf. Numer. Simul. Plasmas, San Francisco, IM3, Sep. 20–24, 1987.*
7. T. Westermann, *Nucl. Instrum. Methods A* **263**, 2711 (1988).
8. M. Matsumoto and S. Kawata, *J. Comput. Phys.* **87**, 488 (1990).
9. R. Loehner and J. Ambrosiano, *J. Comput. Phys.* **91**, 22 (1990).

10. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *J. Comput. Phys.* **47**, 1 (1982).
11. T. Westermann, *Int. J. Numer. Modelling, Electron. Networks, Devices and Fields* **7**, 43 (1994).
12. J. U. Brackbill and J. S. Saltzman, *J., Comput. Phys.* **46**, 342 (1982).
13. J. U. Brackbill and H. M. Ruppel, *J. Comput. Phys.* **65**, 314 (1986).
14. J. F. Thompson, *Appl. Numer. Math.* **1**, 3 (1985).
15. C. Cook, M. P. Desjarlais, S. A. Slutz, T. R. Lockner, D. J. Johnson, S. E. Rosenthal, J. E. Bailey, R. S. Coats, R. J. Leeper, J. E. Maenchen, T. A. Mehlhorn, T. D. Pointon, J. P. Quintenz, C. L. Ruiz, R. W. Stinnett, W. A. Stygar, and J. P. VanDevender, in *Proceedings, 7th Conf. on High-Power Particle Beams, Kernforschungszentrum Karlsruhe, 1988*, edited by W. Bauer and W. Schmidt, p. 35.
16. J. P. VanDevender and D. L. Cook, *Science* **232**, 831 (1986).
17. H. J. Bluhm, P. Hoppé, H. P. Laqua, and D. Rusch, *IEEE Trans. Plasma Sci.* **80**, 995 (1992).
18. E. Halter, *Die Berechnung von elektrostatischen Feldern in Pulsleistungsanlagen*, KfK4072 (Kernforschungszentrum Karlsruhe GmbH, Karlsruhe, 1986).
19. L. W. Ehrlich, *Iterative Methods for large Linear Systems* (Academic Press, New York, 1990), p. 81.
20. D. Seldner and T. Westermann, *J. Comput. Phys.* **79**, 1 (1988).
21. T. Westermann, *J. Comput. Phys.* **101**, 307 (1992).
22. V. Fedorov, private communication, 1992.
23. L. Feher, W. Schmidt, and T. Westermann, *DRIFTPIC, a Computer Program for the Calculation of Ion Trajectories in the Drift Section of Externally Applied-B Diodes*, KfK5207 (Kernforschungszentrum Karlsruhe GmbH, Karlsruhe, 1993).