JOURNAL OF COMPUTATIONAL PHYSICS 63, 20–32 (1986)

A Finite Difference 3-D Poisson–Vlasov Algorithm for Ions Extracted from a Plasma*

J. H. WHEALTON

Fusion Energy Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831

AND

R. W. MCGAFFEY AND P. S. MESZAROS

Computing and Telecommunications Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831

Received May 24, 1984; revised March 15, 1985

A Poisson–Vlasov algorithm has been constructed that solves three-dimensional (3-D) sheath problems. It can be used for accelerator design for intense ion beams extracted from a plasma. It has an advantage over an existing finite element algorithm in that it is much more accurate per unit time spent.

1. INTRODUCTION

Multidimensional sheath configurations are common in practice. However, computer modeling of even a collisionless sheath in more than one dimension is not yet common. A critical use for such an analysis is in the design of high-density ion accelerators in which the ions are extracted from a plasma. The analysis is not entirely straightforward partially because of the highly nonlinear structure of the equations,

$$\nabla^2\phi = \int f\, d\mathbf{v} - e^{-\phi}, \tag{1}$$

$$\nabla\phi \cdot \nabla_{\mathbf{v}} f + \mathbf{v} \cdot \nabla f = 0, \tag{2}$$

where ϕ is the electrostatic potential and f is the ion distribution function. These equations are fully coupled through the ion space charge term (the first term on the right side of Eq. (1)) and the acceleration term ($\nabla\phi$ in Eq. (2)). The nonlinearity of

* Research sponsored by the Office of Fusion Energy, U.S. Department of Energy, under Contract DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc.

0021-9991/86 $3.00

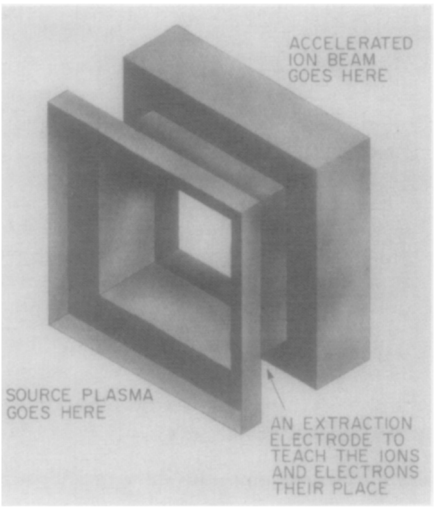

FIG. 1. A typical situation in which a multidimensional Poisson–Vlasov solution is important.

the equations is self-evident; however, the situation is exacerbated by the fact that the exponential nonlinear term on the right side of Eq. (1) is very large and almost canceling with the ion space charge term. This happens in the extraction plasma. In addition, the ions are traveling relatively slowly in this presheath region and are thus highly affected by the electric fields present. It is this combination of large sensitivity to small residuals from large nonlinear terms that has kept other workers from conquering this problem. Ion extraction codes either assume some sort of space-charge-limited flow and neglect explicit considerations of the plasma electrons altogether [1] or consider only the acceleration region, where there is no plasma, along with that part of the sheath up to where the electron density is only on the order of half the ion density [2]. Such ploys are hoped to be somewhat applicable where the sheath is one-dimensional (1-D) but can be expected to fail when the sheath is multidimensional [3].

A typical situation in which a multidimensional Poisson–Vlasov solution is important is shown in Fig. 1. The figure could refer to part of a device used for (1) ion implantation, (2) a plasma diagnostic beam, or (3) a neutral beam generator. In the source plasma, the ion and electron densities are approximately equal. Ions leaving the plasma are accelerated by the applied electric fields, leaving behind a sheath—a continuous transition between the region where the source plasma is and where it is not. (See Figs. 2 and 3.) Algorithms solving the full nonlinear equations generate the sheath automatically without any need to specify nontrivial boundary data. Such algorithms [4–6] exist in two dimensions. These algorithms have met with considerable experimental confirmation [7] and are now used as design tools [8]. A three-dimensional (3-D) algorithm that also solved the nonlinear plasma extraction problem [9] was constructed and enjoyed some preliminary application

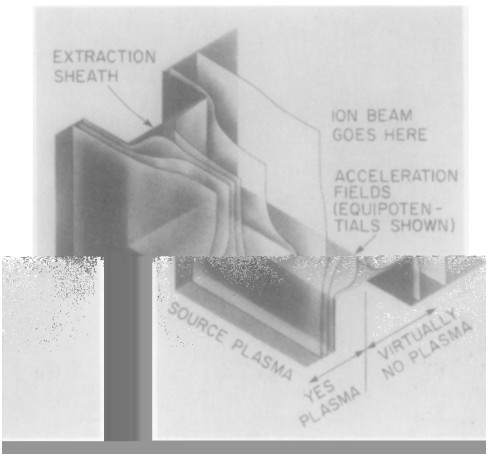

FIG. 2. A view of an extraction sheath showing equipotential surfaces.

[10]. However, the calculation of even the coarsest element configuration strained the memory resources of the CRAY-I computer [placed at the Lawrence Livermore National Laboratory (LLNL) for fusion work] and took the better part of an hour for execution. The algorithm described here is currently operated on a PDP-10 using an order of magnitude less memory while being an order of magnitude faster and solving a problem that is an order of magnitude more refined. Results using this algorithm will be illustrated.

2. THE ALGORITHM

Equation (1) is solved in a finite difference approximation. With reference to Fig. 3, first-order differences are as follows:

$$\frac{\partial\phi}{\partial x}=\frac{\phi_1 h_4^2-\phi_4 h_1^2-\phi_0(h_4^2-h_1^2)}{h_1 h_4(h_1+h_4)\,\Delta x},$$
$$\frac{\partial^2\phi}{\partial x^2}=\frac{2[\phi_1 h_4+\phi_4 h_1-\phi_0(h_1+h_4)]}{h_1 h_4(h_1+h_4)\,\Delta x^2}. \tag{3}$$

For Cartesian coordinates that are not near boundaries, h is unity. Boundaries are accounted for by adjusting the values of h near a boundary, as illustrated for an inhomogeneous Dirichlet boundary in Fig. 4 using the algorithm produced in Ref. [11].

Analogous expressions to Eq. (3) for $\partial^2\phi/\partial y^2$ and $\partial^2\phi/\partial z^2$ are easily obtainable. For the Laplace operator, we therefore have

$$\nabla^2\phi=f(h_i,\phi_0,\phi_i),\qquad i=1,2,\ldots,6,$$

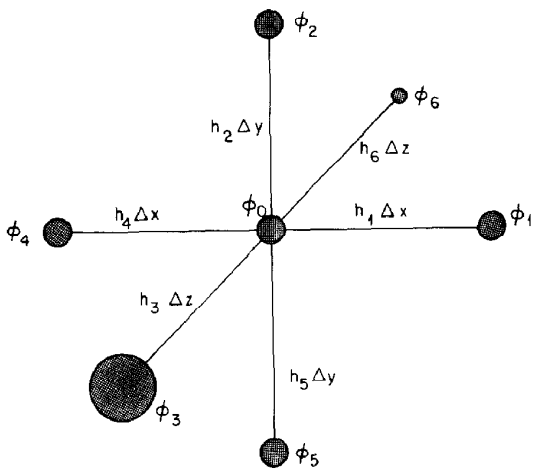

FIG. 3. Notation used for finite differences.

which when solved for ϕ_0 becomes

$$\phi_0 = k_i \phi_i, \qquad i = 1, 2, \ldots, 6,$$

where k_i is evaluated in Appendix A. For inhomogeneous terms, S, to the Laplace equation,

$$\phi_0 = k_i \phi + S/b_0, \qquad i = 1, 2, \ldots, 6. \tag{4}$$

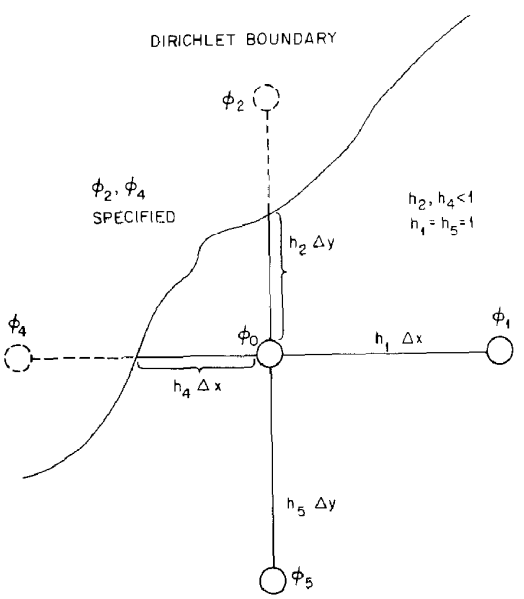

FIG. 4. Notation used for finite differences near a boundary.

In the absence of source terms, a standard expeditious technique is to iterate Eq. (4) over the region by a Gauss–Seidel [12, 13] implicit method. To apply this method for the nth iteration to ϕ_0, the latest iteration for ϕ_i is used so that

$$\phi_0^{(n)} = k_1\phi_1^{(n)} + k_2\phi_2^{(n)} + k_3\phi_3^{(n)} + k_4\phi_4^{(n-1)} + k_5\phi_5^{(n-1)} + k_0\phi_0^{(n-1)}. \tag{5}$$

Equation (5) expresses the notion that generally three of the neighboring values of ϕ are of the same order of iteration as the present value, and the other three are of the previous value. To use this method (relatively implicit) as opposed to the explicit method,

$$\phi_0^{(n)} = k_i\phi_i^{(n-1)}$$

generally converges faster. Using successive overrelaxation (SOR) to produce

$$\phi_0^{(n)} = \beta\phi_0^{(n)} + (1-\beta)\,\phi_0^{(n-1)}, \tag{6}$$

as in Ref. [11], makes the solution converge much faster for $\beta \sim 1.8$, as opposed to $\beta = 1$.

The iterative technique previously described uses only $7n$ words of memory, where n is the number of nodes (typically 10^3–10^4 when using a PDP-10). A straightforward inversion of the matrix of coefficients for the n equations would require n^2 words of memory. It is very difficult to quasi-diagonalize the matrix to such an extent that the memory requirements are competitive with $7n$. Frequently, the computational time is also larger with the direct inversion, since the number of computations is generally larger, even considering that there may be a hundred iterations in the process of Eqs. (5) and (6). This increase in computational time was certainly found to be the case in comparing the efforts of Ref. [9] with this work.

Techniques developed to solve the Poisson equation with the inhomogeneous terms in Eq. (1) are less standard. This is because both terms are very nonlinear, are almost equal, and have opposite signs, exacerbating any numerical difficulties. A Newton method [6] generally suffices to solve the Poisson equation,

$$\nabla^2\phi = -e^{-\phi},$$

for the cases under consideration [7, 8]. For a function as shown in Fig. 5, the root χ_0 is improved to χ_1 by the construction shown or

$$\chi_1 = \chi_0 - f(x_0)/f'(\chi_0).$$

This can be repeated as many times as desired. For the solution to Laplace's equation [Eq. (5)], the Newton method is implemented by

$$\phi_0^{(n,m)} = \frac{[1+\phi_0^{(n,m-1)}]\exp[-\phi_0^{(n,m-1)}] + b_0\phi_0^{(n,0)}}{b_0 + \exp[-\phi_0^{(n,m-1)}]}, \tag{7}$$

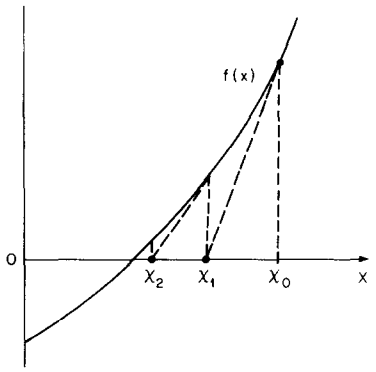

FIG. 5. Illustration of function for which the Newton method converges.

where m is the number of Newton iterations and $\phi_0^{(n,0)}$ is the solution to the homogeneous Laplace equation. In practice it is seldom necessary to resort to more than one Newton iteration since the number of iterations over the mesh, n, and the number of Vlasov iterations give plenty of opportunities to calculate to high accuracy all the nonlinearities of the behavior of the solution. Applying Eq. (6) after Eq. (7) corresponds to a Newton SOR iterative method about which a domain of convergence has been proven [14]. Unfortunately, we never operate in this domain (i.e., $\beta \leqslant 1$ for an $e^{-\phi}$ inhomogeneous term) in order to spare the resources. However, whenever we have chosen to check against the solutions in this established domain, the solutions have agreed within the limits of the machine precision. The domain of convergence is substantially larger than that established by Ref. [14].

It remains now to describe the algorithm for solving Eq. (2) to get the first inhomogeneous term to the Poisson equation [Eq. (1)]. Ion orbits are computed in the volume of interest, and from the properties of these orbits the desired quantities are obtained. We assume in each cell for which the Poisson equation is solved that the electric field vector is a constant, from which it follows that the ion trajectory is a parabola in each transverse coordinate. The algorithm is shown in Fig. 6. Since the acceleration interpolator [11, 13] computes the electric fields on each of eight nodes of a cell, the electric field is linearly varying within a cell. It is worthwhile to refine the trajectory computation (when warranted) to get a more accurate result without decreasing the mesh size. As in Ref. [15], the Vlasov solver is made self-regulating in accuracy, whereupon trajectory refinement is undertaken only in those places that need it. The trajectories are piecewise parabolas. Refining the trajectories and selecting the point at which such refinement is desirable are major factors in drastically reducing the computational time. The solution to the Vlasov equation [Eq. (2)] is generally more time consuming than that of Eq. (1) by a factor of 10. As denoted by z in Fig. 6, an axis (generally the axis parallel to the average acceleration fields) is incremented for the orbits in the fashion shown in Fig. 6.

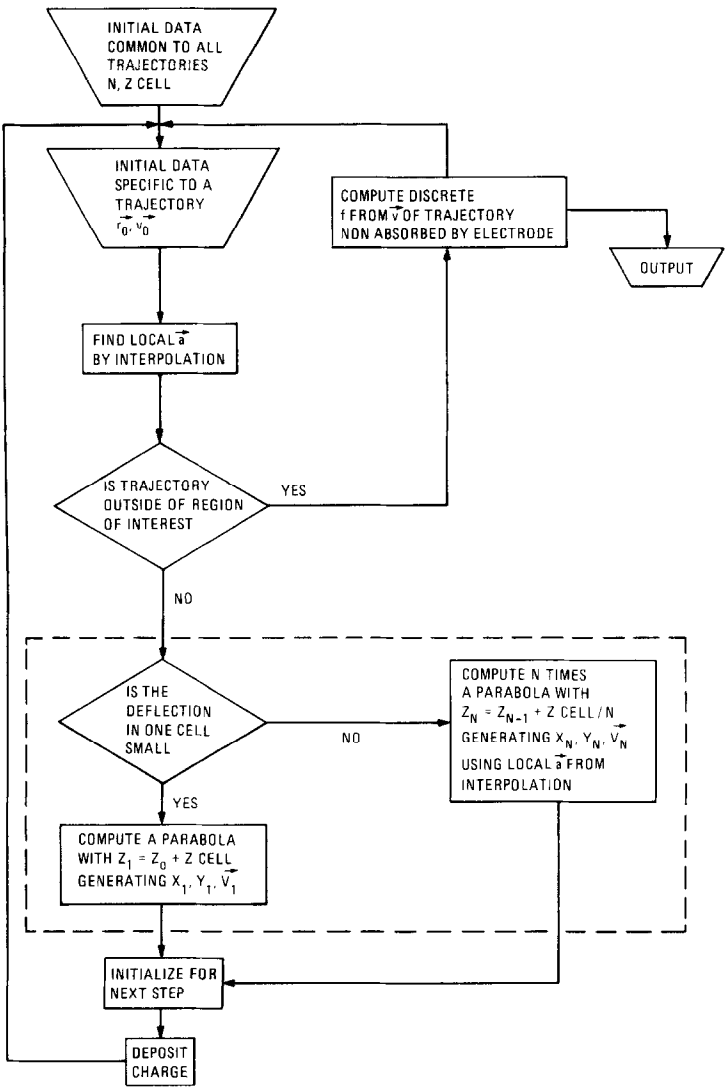

FIG. 6. Algorithm for Vlasov equation.

The trivial relationship between the coordinates inside an element and the global elements for the uniform Cartesian grid used in this algorithm allows orders-of-magnitude savings in the Vlasov solver over that employed in the irregular elements of Ref. [9]. The charge deposition is done in a straightforward manner, knowing the velocity of each trajectory and the charge it carries. Full coupling of the equation is done by iteration, as shown in Fig. 7. Typically, about 6 to 20 major iterations need to be performed before the solutions to Eqs. (1) and (2) converge.

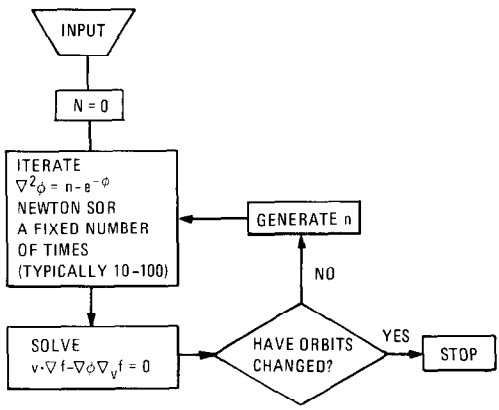

FIG. 7. Algorithm for Poisson–Vlasov iteration.

3. EXAMPLE

We consider a square aperture. A view along a symmetry semiplane, $x = 0$, is shown in Fig. 8. The dotted lines are the equipotentials of the electrostatic field; the solid lines are ion trajectories, which go from left to right. On the left, $y = 0$, is the source plasma; on the right is the acceleration column. The high density of equipotentials near the left is the sheath that was alluded to in the introduction.

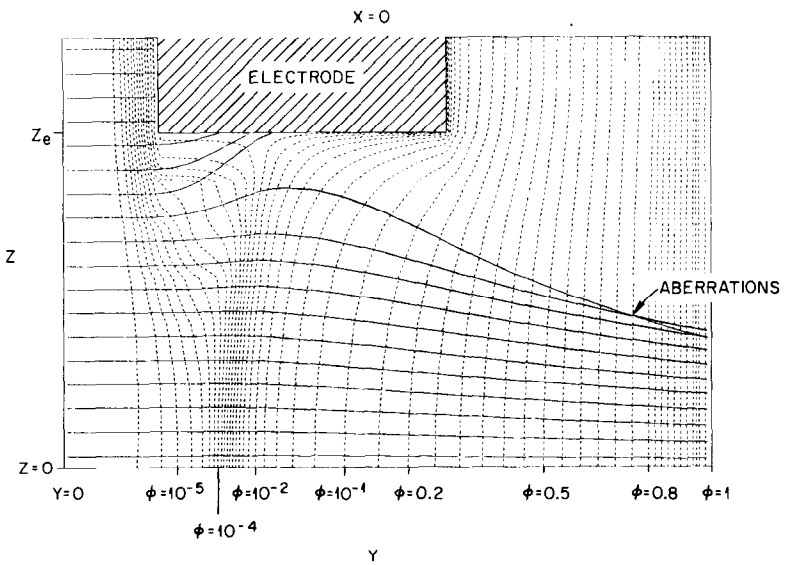

FIG. 8. A view of a symmetrical semiplane of a square aperture; the source plasma is located on the left side.

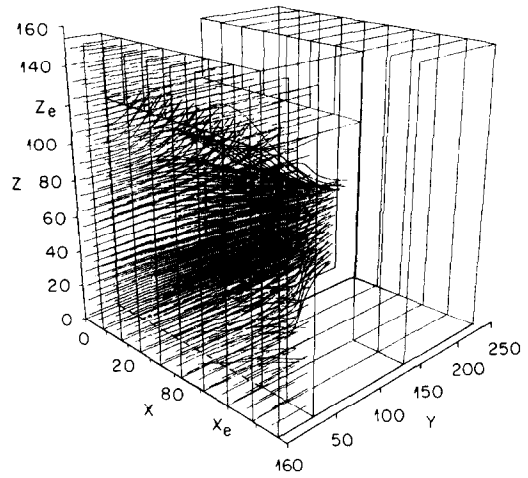

FIG. 9. A view of the entire square aperture extraction area.

Equipotentials near the plasma and through the sheath are spaced exponentially. The potential plunges by two orders of magnitude over only 5% of the axial distance y. Figure 9 is a view of the entire device, where the viewer's position is at large positive values of x and z and large negative values of y, similar to Fig. 1. The boundary data are shown, as well as the trajectories, for the final Vlasov iteration. Figure 8 corresponds to a plane at $x = 0$. A view similar to Fig. 8 but comprising all the planes (with the three closest to the viewer at the end of the electrode) is shown in Fig. 10 (viewed from large x). A view from the top of the device (large z) is

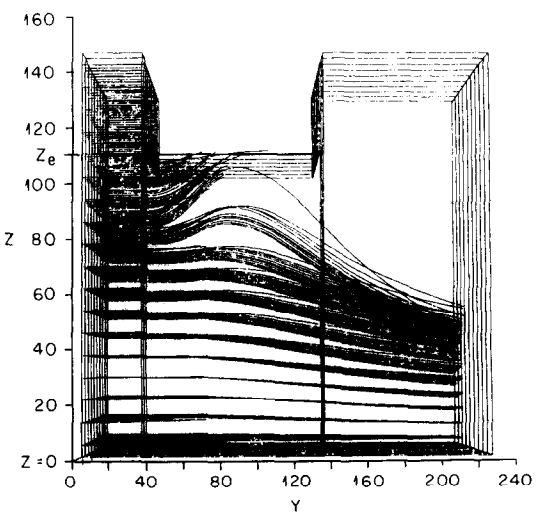

FIG. 10. A view of the device shown in Fig. 9 for large x.

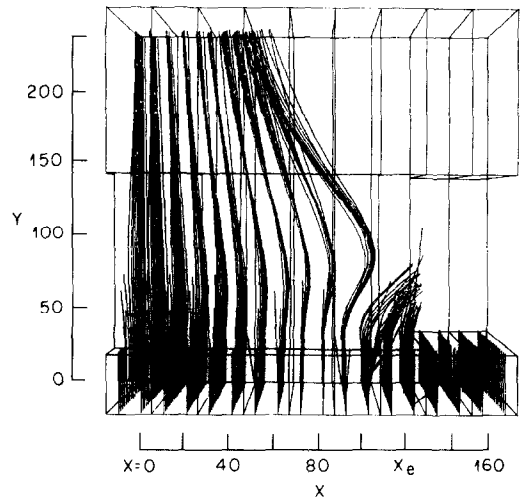

FIG. 11. A view of the device shown in Fig. 9 for large z.

shown in Fig. 11. A view from the plasma (large negative y) is shown in Fig. 12. The likeness of the solutions shown in Figs. 10 and 11 and the degree of diagonal symmetry in Fig. 12 indicate that the numerical procedures of the x and z directions give the same results, as they should owing to the symmetry of the boundary data about this diagonal. Furthermore, the solution for a very elongated position, $x_e \gg z_e$, along the symmetry plane gives the same result as the experimentally verified 2-D algorithm of Ref. [5]; this is as it must be since the algorithms in the limit $x_e \gg z_e$ are identical.

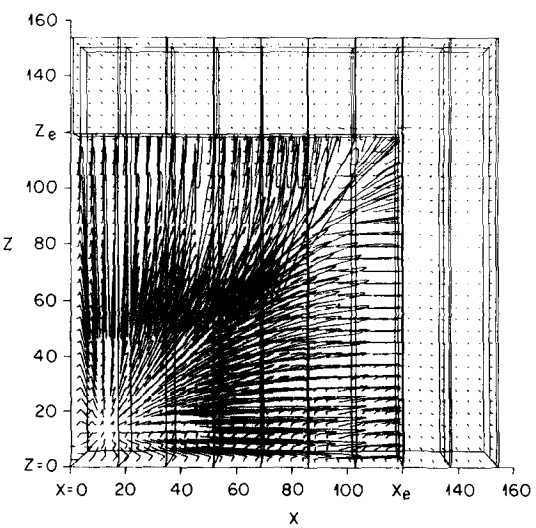

FIG. 12. A view of the device shown in Fig. 9 for large $-y$.

For the example shown in Figs. 8–12, 1500 trajectories were used to converge the Vlasov iteration over 12 iterations. The corresponding number of orbit parabolas was 310,000; the number of mesh points was 14,100, for which, in the endeavor to solve the Poisson equation, 45 sweeps over the volume were performed with an overrelaxation of 1.75. A one-step Newton iteration was used for the nonlinear exponential electron term for each of the 12 Vlasov iterations (8×10^6 algebraic local difference equations) solved. This example took $\sim$66 min on a PDP-10.

The example considered in Ref. [9] was reconsidered using this code [16], except that the number of nodal points was doubled to 3600. The calculation took one-tenth as much memory, and the computational time was reduced by a factor of $\approx$40 (13 min on a PDP-10 vs 40 min on a CRAY-I). The results were the same.

The most interesting phenomena, which are to be studiously avoided, are the aberrations present in the beam as seen by the crossing trajectories in Figs. 8, 10, and 11. Figure 13 shows an emittance diagram for the V_z, z occupation in phase space for the trajectories at the maximum value of y considered in Figs. 8–12. The population due to orbits with various initial x ($\equiv x_0$) is shown. An emittance diagram for V_x, x occupation in phase space would be identical except for labeling. In 2-D geometries much care has been devoted to reducing such aberrations. For circular holes [17], the matter ends there; however, for slots [18], the end of the slot requires care to avoid the situation of Fig. 11. It is not sufficient to make the slot much longer than it is wide, since the hideous aberrations at the end may cause electrode interception by the ion beam with the attendant annoyances of melting and breakdown.

One should not deduce from the preceding simple example that the type of electrode shape considered is limited; there is no limitation on electrode shapes or shapes of Dirichlet boundary conditions.

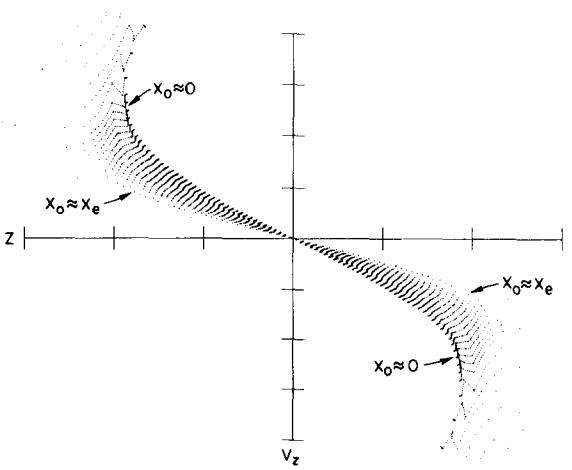

FIG. 13. An emittance of the V_z, z occupation in phase space for the trajectories at the maximum value of y of the device shown in Fig. 9. The ion temperature is zero.

APPENDIX A

Evaluation of k_i, $i = 1, 2, \ldots, 6$, follows:

$$b_0 k_1 = 2b_1 h_4,$$
$$b_0 k_2 = 2b_2 h_5,$$
$$b_0 k_3 = 2b_3 h_6,$$
$$b_0 k_4 = 2b_1 h_1,$$
$$b_0 k_5 = 2b_2 h_2,$$
$$b_0 k_6 = 2b_3 h_3,$$

where

$$b_0 = 2b_1(h_1 + h_4) + 2b_2(h_2 + h_5) + 2b_3(h_3 + h_6),$$
$$b_1^{-1} = h_1 h_4 (h_1 + h_4)\, \Delta x^2,$$
$$b_2^{-1} = h_2 h_5 (h_2 + h_5)\, \Delta y^2,$$
$$b_3^{-1} = h_3 h_6 (h_3 + h_6)\, \Delta z^2.$$

ACKNOWLEDGMENTS

We are especially indebted to D. E. Wooten for providing the algorithms that connect the output of the subject algorithm with the plotting package DISSPLA that we lease. We also thank E. F. Jaeger, J. C. Whitson, and J. W. Wooten for their help.

REFERENCES

1. P. T. KIRSTEIN AND J. S. HORNSBY, *IEEE Trans. Electron Devices* **11** (1964), 196; J. L. HARRISON, *J. Appl. Phys.* **39** (1968), 3827; B. I. VALKOV, A. G. SVESHNIKOV, AND N. N. SEMASHKO, *Sov. Phys.-Dokl.* **16** (1972), 1040; W. W. HICKS *et al.*, *Nucl. Instrum. Methods* **139** (1976), 25; Y. OHARA *et al.*, *Japan J. Appl. Phys.* **17** (1978), 423; Y. OHARA, *J. Appl. Phys.* **49** (1978), 4711; W. C. LATHEM, *J. Spacecraft* **5** (1968), 735; W. S. COOPER, K. H. BERKNER, AND R. V. PYLE, *Nucl. Fusion* **12** (1972), 263.
2. K. HALBACH, Lawrence Berkeley Laboratory Report LBL-4444, Berkeley, California, 1975 (unpublished); E. F. JAEGER AND J. C. WHITSON, ORNL/TM-4990, Oak Ridge Natl. Lab., 1975 (unpublished); K. ASAI *et al.*, *Japan J. Appl. Phys.* **15** (1976), 1343.
3. C. E. LEJEUNE, *Adv. Electron. Electron Phys.* (1983), 207.
4. J. H. WHEALTON, E. F. JAEGER, AND J. C. WHITSON, *J. Comput. Phys.* **27** (1978), 32; J. H. WHEALTON AND J. C. WHITSON, *Part. Accel.* **10** (1980), 235.
5. J. H. WHEALTON, *IEEE Trans. Nucl. Sci. NS* **28** (1981), 1358; J. H. WHEALTON, *J. Comput. Phys.* **40** (1981), 491; J. H. WHEALTON, *Nucl. Instrum. Methods* **189** (1981), 55.
6. J. C. WHITSON, J. SMITH, AND J. H. WHEALTON, *J. Comput. Phys.* **28** (1978), 408.

7. J. H. WHEALTON, E. F. JAEGER, AND J. C. WHITSON, *Rev. Sci. Instrum.* **48** (1977), 829; L. R. GRISHAM, C. C. TSAI, J. H. WHEALTON, AND W. L. STIRLING, *Rev. Sci. Instrum.* **48** (1977), 1037; J. KIM, J. H. WHEALTON, AND G. SCHILLING, *J. Appl. Phys.* **49** (1978), 517; J. H. WHEALTON, L. R. GRISHAM, C. C. TSAI, AND W. L. STIRLING, *J. Appl. Phys.* **49** (1978), 3091; J. H. WHEALTON *et al.*, *Appl. Phys. Lett.* **33** (1978), 278; M. M. MENON *et al.*, *Rev. Sci. Instrum.* **51** (1980), 1163; C. N. MEIXNER, M. M. MENON, AND C. C. TSAI, *J. Appl. Phys.* **52** (1981), 1167; W. L. GARDNER, J. H. WHEALTON, *et al.*, *Rev. Sci. Instrum.* **52** (1981), 1625; W. L. GARDNER *et al.*, *Rev. Sci. Instrum.* **53** (1982), 424.
8. J. H. WHEALTON AND C. C. TSAI, *Rev. Sci. Instrum.* **49** (1978), 495; J. H. WHEALTON, G. G. KELLEY, O. B. MORGAN, AND G. SCHILLING, *Nucl. Instrum. Methods* **154** (1978), 441; J. H. WHEALTON AND J. C. WHITSON, *J. Appl. Phys.* **50** (1979), 3964; W. L. STIRLING *et al.*, *Appl. Phys. Lett.* **35** (1979), 104; J. H. WHEALTON, R. W. MCGAFFEY, AND E. F. JAEGER, *Appl. Phys. Lett.* **36** (1980), 91; J. H. WHEALTON, R. W. MCGAFFEY, AND W. L. STIRLING, *J. Appl. Phys.* **52** (1981), 3787; J. H. WHEALTON, J. W. WOOTEN, AND R. W. MCGAFFEY, *J. Appl. Phys.* **53** (1982), 2806; J. H. WHEALTON, *J. Appl. Phys.* **53** (1982), 2811; J. H. WHEALTON AND R. W. MCGAFFEY, *Nucl. Instrum. Methods* **203** (1982), 377.
9. J. W. WOOTEN, J. H. WHEALTON, D. H. MCCOLLOUGH, R. W. MCGAFFEY, J. E. AKIN, AND L. J. DROOKS, *J. Comput. Phys.* **43** (1981), 95.
10. J. W. WOOTEN, J. H. WHEALTON, AND D. H. MCCOLLOUGH, *J. Appl. Phys.* **52** (1981), 6418.
11. J. S. HORNSBY, CERN-63-7, Conseil Européen des Recherches Geneva, 1963.
12. G. D. SMITH, "Numerical Solution of Partial Difference Equations: Finite Difference Methods," 2nd ed., Oxford Univ. Press, Oxford, 1978.
13. E. F. JAEGER AND J. C. WHITSON, ORNL/TM-4990, Oak Ridge Natl. Lab., 1975 (unpublished).
14. J. M. ORTEGA AND W. D. RHEINBOLDT, "Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, New York, 1970.
15. J. H. WHEALTON, *J. Comput. Phys.* **40** (1981), 491.
16. J. H. WHEALTON, R. W. MCGAFFEY, AND P. S. MESZAROS, "IEEE Conference on Plasma Science, San Diego, 1983" (unpublished).
17. G. W. HAMILTON, J. L. HILTON, AND J. S. LUCE, *Plasma Phys.* **10** (1968), 687; R. C. DAVIS, O. B. MORGAN, L. D. STEWART, AND W. L. STIRLING, *Rev. Sci. Instrum.* **43** (1972), 278; K. N. LEUNG *et al.*, *Rev. Sci. Instrum.* **49** (1978), 321; V. V. FOSNIGHT, T. R. DILLION, AND G. SOHL, *J. Spacecraft* **7** (1970), 226; L. D. STEWART, J. KIM, AND S. MATSUDA, *Rev. Sci. Instrum.* **46** (1975), 1193; J. R. COUPLAND AND T. S. GREEN, *Nucl. Instrum. Methods* **125** (1975), 197; T. SAGAWARA AND Y. OHARA, *Japan J. Appl. Phys.* **14** (1975), 1029; N. SAKUDO, K. TOKIGUCHI, H. KOIKE, AND I. KANONMATA, *Rev. Sci. Instrum.* **48** (1977), 762; E. THOMPSON, *Part. Accel.* **4** (1972), 69; J. R. COUPLAND *et al.*, *Rev. Sci. Instrum.* **44** (1973), 1258; T. S. GREEN, *J. Appl. Phys. D* **9** (1976), 1165; H. R. KAUFMAN, *Adv. Electron. Electron Phys.* **36** (1974), 265; T. S. GREEN, *Rep. Prog. Phys.* **37** (1974), 1257; T. S. GREEN, *IEEE Trans. Nucl. Sci. NS* **23** (1976), 918; Y. OHARA *et al.*, *Japan J. Appl. Phys.* **15** (1976), 135; G. SOHL AND V. V. FOSNIGHT, *J. Spacecraft* **6** (1969), 143.
18. K. W. EHLERS *et al.*, *J. Vac. Sci. Technol.* **10** (1973), 922; S. MATSUDA *et al.*, *Bull. Am. Phys. Soc.* **18** (1973), 132; K. H. BERKNER *et al.*, *Bull. Am. Phys. Soc.* **21** (1976), 1136; H. R. KAUFMAN, *Adv. Electron. Electron Phys.* **36** (1974), 265; T. S. GREEN, *Rep. Prog. Phys.* **37** (1974), 1257; T. S. GREEN, *IEEE Trans. Nucl. Sci. NS* **23** (1976), 918; Y. OHARA *et al.*, *Jpn. J. Appl. Phys.* **15** (1976), 135; W. C. LATHEM, *J. Spacecraft* **5** (1968), 735; X. P. GRIGOREV, *Sov. Phys.-Tech. Phys.* **16** (1971), 909; K. ASAI *et al.*, *Japan J. Appl. Phys.* **15** (1976), 1343.