NASA CONTRACTOR
REPORT

NASA CR-144159

COMPUTATION OF COLLISIONLESS STEADY-STATE PLASMA FLOW PAST A CHARGED DISK

By Lee W. Parker
Lee W. Parker, Inc.
Concord, Massachusetts

February 1976

Prepared for

NASA - GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA CR-144159		

4. TITLE AND SUBTITLE	5. REPORT DATE
Computation of Collisionless Steady-State Plasma Flow Past a Charged Disk	February 1976
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT #
Lee W. Parker	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
Lee W. Parker, Inc. Concord, Massachusetts	11. CONTRACT OR GRANT NO. NASA Order H-16314B
	13. TYPE OF REPORT & PERIOD COVERED
12. SPONSORING AGENCY NAME AND ADDRESS	Contractor Report
National Aeronautics and Space Administration Washington, D. C. 20546	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

A computer method is presented, using the "inside-out" approach, for predicting the structure of the disturbed zone near a moving body in space. The approach uses fewer simplifying assumptions than other available methods, and is applicable to large ranges of the values of body and plasma parameters. Two major advances concerning 3-dimensional bodies are that (a) thermal motions of ions as well as of electrons are treated realistically by following their trajectories in the electric field, and (b) the technique for achieving self-consistency is promising for very large bodies.

Three sample solutions are obtained for a disk-shaped body, charged negatively to a potential 4kT/e. With ion Mach number 4, and equal ion and electron temperatures, the wakes of a relatively small body (radius 5 Debye lengths) and a relatively large body (radius 100 Debye lengths) both begin to fill up between 2 and 3 body radii downstream. For the large body there is in addition a potential well (about 6kT/c deep) behind the body. Increasing the ion Mach number to 8 for the large body causes the potential well to become wider and longer but not deeper. For the large body, the quasineutrality assumption is validated outside of a cone-shaped region in the very near wake. For the large as well as the small body, the disturbed zone behind the body extends transversely no more than 2 or 3 body radii, a result of significance for the design of spacecraft boom instrumentation.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
	Unclassified - Unlimited for James T. Murphy, Director Program Development

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	105	NTIS

FOREWORD

This report presents the results of a study to access the plasma disturbance created by a large body of Space Shuttle dimensions traversing the ionosphere. This study was commissioned by the Plasma Flow and Interaction Section of the Atmospheric Magnetospheric and Plasmas in Space (AMPS) science definition working group. The study was performed by Lee Parker, Inc. of Concord, Mass. for the NASA, Marshall Space Flight Center under the direction of W. R. Roberts of the AMPS Task Team.

CHAPTER 1

INTRODUCTION

The problem of theoretically calculating the structure of the disturbed plasma (frequently referring to the wake and/or sheath) around a moving body in space is equivalent to that of solving a complicated system of coupled nonlinear partial differential/integral equations. The equations consist of the Vlasov (collisionless Boltzmann) equations for the ions and electrons, and the Poisson equation relating the electric field to the distributions of ions and electrons. The difficulty is essentially a numerical one because analytic solutions are not possible (for cases of interest), and there is no unique approach. In cases of stationary bodies (Parker, 1973 and 1975), as well as moving bodies (other theoretical references of this report), combinations of numerical techniques (finite differences, iteration, quadratures, etc.) are required for treating various parts of the problem. For either stationary or moving bodies, the choices of techniques and their use to achieve consistent solutions for any given set of physical parameters (defining body and plasma) have never been obvious. Innovations are frequently required. The purpose of this report is to review some of the available techniques for a moving body (with emphasis on the wake), to describe in detail a new combination of techniques which appear to be reasonably successful over a large range of the physical parameters, and to present sample solutions as well as the implementing computer program.

Various approaches which have been used for this type of problem are summarized in Chapter 2. In all such calculations simplifying assumptions are made. The customary ones are:

- Collisions negligible.
- Geomagnetic field negligible.
- Simple geometry (sphere, disk, cylinder, etc.)
- Simple surface reactions (usually, charged particles are neutralized).
- Prescribed surface emission (usually none, but simplified

photoelectron and secondary-electron emission are includable).

- Conducting body (usually perfectly conducting, but finite conductivities are includable).
- Steady state.

These assumptions may be questioned (for example the neglect of time-dependent phenomena), but they may be at least partially relaxed by employing known techniques to generalize the calculations. In the interest of achieving reasonably economical calculations within the limits of available computers, the above assumptions in their usual form are adopted in the present work.

The techniques and computer program described in Chapters 2 and 3 and in the Appendices have been developed to solve the coupled Poisson-Vlasov system of equations to obtain distributions of ion and electron density, and potential, about 3-dimensional bodies (with axial symmetry about the direction of plasma flow). The program uses the "inside-out" method developed by the author in 1964, which follows ion and electron trajectories backward in time from the point in space at which it is desired to know the velocity distribution, to their origin in the undisturbed plasma where the distribution is known.

Briefly, the present approach (see Chapter 2) differs from that of Call (1969) and Martin (1974) by including both the ion and the electron thermal motions, whereas Call and Martin represent the distribution of ions by a cold beam, and of electrons by the Boltzmann factor. The approach differs from that of Taylor (1967) in that (a) it is applied to 3-dimensional bodies whereas Taylor treats an infinitely-long cylinder, and (b) the Poisson and Vlasov calculations are cycled until self-consistency is achieved, whereas Taylor's calculation is terminated after the first cycle. The approach differs from that of Grabowski and Fischer (1975) because they (a) assume that quasineutrality holds throughout space (which is invalid in the very near wake), and (b) apply their method to an infinitely-long cylinder. Differences with other methods are outlined in Chapter 2. The most similar calculation previously done was for an infinitely-long cylinder by Fournier

(1971), using the inside-out method. In general, the present approach uses fewer simplifying assumptions and is thus applicable to a larger range of parameters than other available methods.

Two major advances are represented by the present program, as opposed to previous approaches, particularly with regard to wakes of 3-dimensional bodies:

(1) Thermal motions of ions as well as of electrons are treated realistically by following their trajectories in the electric field.

(2) The technique for achieving self-consistency is promising for large bodies many orders-of-magnitude larger than the Debye length (the Shuttle-Orbiter or the moon, for example).

Solutions may be obtained with reasonable amounts of computer time by judicious choices of grid points and other numerical parameters. The method can be extended to include an arbitrarily-shaped body (presently a body of revolution).

The structure of this report is as follows.

Chapter 2 comprises a review and summary of previous approaches, classified on the basis of how they treat the Vlasov problem (calculation of ion and electron densities and currents). In particular, the inside-out method is treated in detail. The computational method for number and current density quadratures is given in Appendix A. (Throughout the report the words "orbit" and "trajectory" are used interchangeably.)

In Chapter 3, the method of self-consistent solution by Poisson-Vlasov iteration is treated. The method of solution of the Poisson problem by finite differences is described in Appendix B. The "ion-density option" discussed in Chapter 3 is appropriate for the large-body problem (see also Appendix B and Chapter 4).

The FORTRAN listing for the computer program, and the descriptions of input and output data, are given in Appendix C.

Chapter 4 presents numerical solutions for three sample disk problems, showing the effects of changes in body size and in ion Mach number. The results are presented in the form of transverse profiles of ion density,

electron density, and potential, in the wake. (Data on the sheath in front of the disk are available but are not given in this report.) The key results are the following.

For a relatively small body (radius = 5 Debye lengths), at potential = -4 kT/e and ion Mach number = 4, there is no prominent wake structure such as large-amplitude "bumps" (enhancements/depletions) in the ion or electron density. The wake becomes filled in between 2 and 3 body radii downstream, and there is no potential well. The wake disturbance is essentially confined to a region of length in the axial dimension about 4 radii, and transverse radial dimension about 1.5 radii.

For a large body (radius = 100 Debye lengths, i.e., larger than has been previously treated realistically), and for the same body potential and ion Mach number as above, the wake begins to fill up again between 2 and 3 body radii downstream. The wake disturbance extends more than 6 body radii downstream, but transversely only between 2 and 3 radii. There is a potential well near the wake surface of the disk, and quasineutrality is valid outside of a cone-shaped region near the wake surface (with the disk forming the base of the cone).

For the same large body, but with ion Mach number 8 instead of 4, the dimensions of the wake-disturbance region are not significantly changed, but the filling-up occurs further downstream. The potential well becomes wider and longer, although the depth is similar. In addition, there seems to be a central core of essentially ambient density along the axis, for both ions and electrons.

To more comprehensively establish the practical applicability of the present computer method to future AMPS/Spacelab missions, it would be of interest to compare theory and experiment for cases where in-situ and laboratory simulation data are available. At present there are more laboratory results (Oran et al., 1975; Fournier and Pigache, 1975), than in-situ results (Henderson and Samir, 1967; Samir et al., 1973). However, it is presently still difficult to simulate ion transverse velocity distributions in the laboratory, and the effective ion temperature is frequently low. Since various ratios of ion temperature to electron temperature may be treated by the

program with relatively minor modifications, computations should be made with "cool" ions to facilitate lab-theory comparisons. The present theoretical model should also be compared, using selected ionosphere-magnetosphere example problems, with other available theoretical models. The other models may be less realistic but they may have advantages of relative simplicity and economy; they may also be "calibrated" through such comparisons.

The present computer method gives information regarding the dimensions of the disturbed zone about a body. Information of this kind should be useful for estimating the lengths of booms to be deployed, for example, on the Spacelab to keep outboard instrumentation outside the disturbance created by various structures on the spacecraft. In this sense, the computations may be regarded as a phase of a feasibility study.

CHAPTER 2

APPROACHES: REVIEW AND SUMMARY

All of the approaches to the body-in-a-plasma problem have in common the following elements. The quantities to be computed include (a) the potential distribution and (b) the ion and electron density distributions. One may also include the associated surface current densities. The equations to be solved simultaneously are (a) the Vlasov equation for ions, (b) the Vlasov equation for electrons, and (c) the Poisson equation. The solutions of the Vlasov equations (velocity-distribution functions) are used to compute number densities (and surface current densities). The number-density distributions become input to the (right-hand side of the) Poisson equation which yields the potential distribution. Finally, an iterative procedure is used for self-consistency, wherein the density and potential distributions are successively cycled until satisfactory convergence has been achieved.

The steady-state Vlasov equations for ions and electrons state that the velocity-distribution functions remain constant along particle trajectories. With the electric field assumed given (numerically in terms of a spatial grid about the body), solving the Vlasov equations means formally that one determines, from the shapes of the trajectories, the ion and electron velocity distributions at the grid points. The trajectories relate local velocities at a given grid point to those at infinity. Through these relationships, the ion or electron number density at the point may be evaluated by a velocity-integral over the local velocity distribution. Similarly, the current density may be evaluated at desired locations (usually the body surface).

It is convenient to classify the various theoretical approaches on the basis of how they treat the trajectory part of the Vlasov problem. "Inside-out" methods follow the trajectories backward in time into the undisturbed plasma, while "outside-in" methods follow the trajectories forward, in the direction of physical motion of the particles. (In an outside-in method, the velocity-distribution function is not calculated;

rather, the density is evaluated directly.) "Other" methods denote approximations where trajectories are not followed at all. The three approaches are discussed below.

There exists as yet no systematic comparison of the results of the various approaches with one another.

Before discussing the various approaches, we may define here the parameters of interest:

Plasma Parameters

n_0 = unperturbed number density at infinity

T_i, T_e = ion, electron temperatures

m_i = ion mass (electron mass not required)

λ_D = electron Debye length

Body Parameters

r_0 = characteristic dimension

v_0 = velocity

Φ_0 = body potential

$\phi_0 = e\Phi_0/kT_e$ = dimensionless body potential

$M = v_0 \sqrt{m_i/2kT_i}$ = ion Mach number (electron Mach number assumed negligible)

$\lambda_D = \lambda_D/r_0$ = Debye number

Henceforth all lengths are to be considered normalized by r_0. Thus, λ_D will denote the dimensionless Debye number. Potentials are normalized by kT_e/e, so that $\phi(\vec{r})$ denotes the dimensionless potential at the spatial point $\vec{r}$. Number densities are normalized by n_0, so that $n(\vec{r})$ denotes the dimensionless density at $\vec{r}$. In the calculations involving integrations over velocities, $\vec{v}$ will denote a velocity normalized by the value of $\sqrt{2kT/m}$ associated with the particles of interest. Similarly, E will denote total energy normalized by kT. Velocity-distribution functions (denoted by f) will be normalized by n_0. For a given body geometry, there

are four dimensionless physical parameters of interest, namely, λ_D, ϕ_o, and M, along with the temperature ratio T_i/T_e. Table 2-1 shows a sampling of the parameters used in various previous calculations.

2A. INSIDE-OUT METHOD

Consider a single species of (charged) particle, i.e., ions or electrons. The electric field is assumed to be known. In order to compute the number density $n(\vec{r})$ at the point $\vec{r}$, one must evaluate the triple integral over velocity space:

$$n(\vec{r}) = \int\int\int f(\vec{r}, \vec{v})\, dv_x dv_y dv_z \qquad (2\text{-}1)$$

where $f(\vec{r}, \vec{v})$ is the distribution function which satisfies the Boltzmann equation for the given species of particle, $\vec{r}$ is the radius vector of the space point of interest, and $\vec{v}$ is the local velocity of a particle at $\vec{r}$. The velocity-volume element is written as if cartesian coordinates were being used, but the product $dv_x dv_y dv_z$ is intended to symbolize an arbitrary coordinate system. Similarly, in order to compute the collected current density at points on the surface of a body, one must evaluate at each point a triple integral over velocity space of the form

$$j(\vec{r}) = \int\int\int f(\vec{r}, \vec{v})\, v_n dv_x dv_y dv_z \qquad (2\text{-}2)$$

where v_n is the component of the particle velocity normal to the surface at the point $\vec{r}$.

The problem is thus to evaluate f. Since the problems of interest are assumed to be collisionless and constant in time, the distribution function f satisfies the steady-state Vlasov (or collisionless Boltzmann) equation, namely,

$$\vec{v} \cdot \nabla f + \vec{a} \cdot \nabla_v f = 0 \qquad (2\text{-}3)$$

TABLE 2-1

PARAMETERS ADOPTED IN PREVIOUS WAKE CALCULATIONS

	Mach Number	Debye Number	Ion-Attractive Body Potential
Fournier (1971)	1, 6, 10	$\frac{1}{10}$, $\frac{2}{3}$	-3, 0, 1, 2.75, 6, 40
Davis and Harris (1961)	6 → 7	$\frac{1}{25}$, $\frac{1}{10}$	20 → 1000
Call (1969)	1 → 8	$\frac{1}{25}$ → 5	0 → 40
McDonald and Smetana (1969)	0 → 3	1, $3\frac{1}{3}$, 10	10, 25
Maslennikov and Sigov (1964)	2, 7	1, 12	0 → 40
Liu and Jew (1968, 1969)	4, 8	$\frac{1}{20}$, $\frac{1}{5}$, 1	1, 5
Kiel et al. (1968)	5, 8	$\frac{1}{1000}$, $\frac{1}{100}$, $\frac{1}{10}$	3
Martin (1974)	4 → 10	$\frac{1}{9}$ → 1	5, 9
Grabowski and Fischer (1975)	0 → 1.4	0 (quasineutral)	irrelevant
Taylor (1967) (first order only)	6	2/3	0.25 → 20

where $\vec{a}$ is the vector acceleration of a particle passing with velocity $\vec{v}$ through the point $\vec{r}$. The gradient operators ∇ and ∇_v operate on the components of $\vec{r}$ and of $\vec{v}$, respectively. Equation (2-3) states that f is constant along a particle orbit, which is characterized by the constants of the motion. In a general electrostatic field (here assumed given) whose sources are volume and surface charges, the total energy E is conserved, where the dimensionless E is defined by

$$E = v^2 + \phi(\vec{r}) \qquad (2\text{-}4)$$

and $\phi(\vec{r})$ is the dimensionless potential energy of the particle at $\vec{r}$.

With $\phi(\vec{r})$ a known function of $\vec{r}$, one may evaluate the integrals in Eqs. (2-1) and (2-2) by following orbits backward in time with trajectory calculations to a point where f is known. For example, in the case of a body immersed in a plasma, f is assumed to be known at infinity (where ϕ vanishes), and is assumed to have at infinity a prescribed energy distribution, such as a Maxwellian with drift, or a more general distribution. Also, f is assumed to be known on the surfaces of electrodes. If a surface emits particles, its distribution function must be prescribed. If the surface absorbs without re-emitting charged particles, the distribution functior (of emitted particles) is prescribed to be zero. Thus, f is discontinuous in velocity space: That is, the physically-possible velocity space (at the point $\vec{r}$) is divided into two domains, namely, the domain of orbits which have come to $\vec{r}$ from infinity, and the domain of orbits which have come to $\vec{r}$ from electrode surfaces. In the latter domain, f vanishes if there is no emission. Therefore, f is discontinuous on the boundary between the two domains in velocity space. The shape of the boundary between the two domains depends, of course, on the geometry and the potential function ϕ, and it is the heart of the problem (a) to determine the boundary of the domain of orbits coming from infinity, and (b) to evaluate the integrals Eqs. (2-1) and (2-2) over that domain of velocity space.

In practice, one need not in general determine explicitly the boundary of the domain in velocity space of orbits coming from infinity.

Rather, one may follow a large number of orbits backward in time (computationally), and evaluate the moment integrals, Eqs. (2-1) and (2-2), automatically from the results of the orbit-following. It may, however, under some circumstances be more accurate and efficient to determine this boundary. To do so would complicate the computer programming.

For a Maxwellian distribution drifting with Mach number M, the dimensionless velocity-distribution function at infinity may be written:

$$f_\infty = \frac{1}{\pi^{3/2}} e^{-(v_\infty^2 + M^2 - 2Mv_{z\infty})} = \frac{1}{\pi^{3/2}} e^{-(\phi + v^2 + M^2 - 2Mv_{z\infty})} \qquad (2\text{-}5)$$

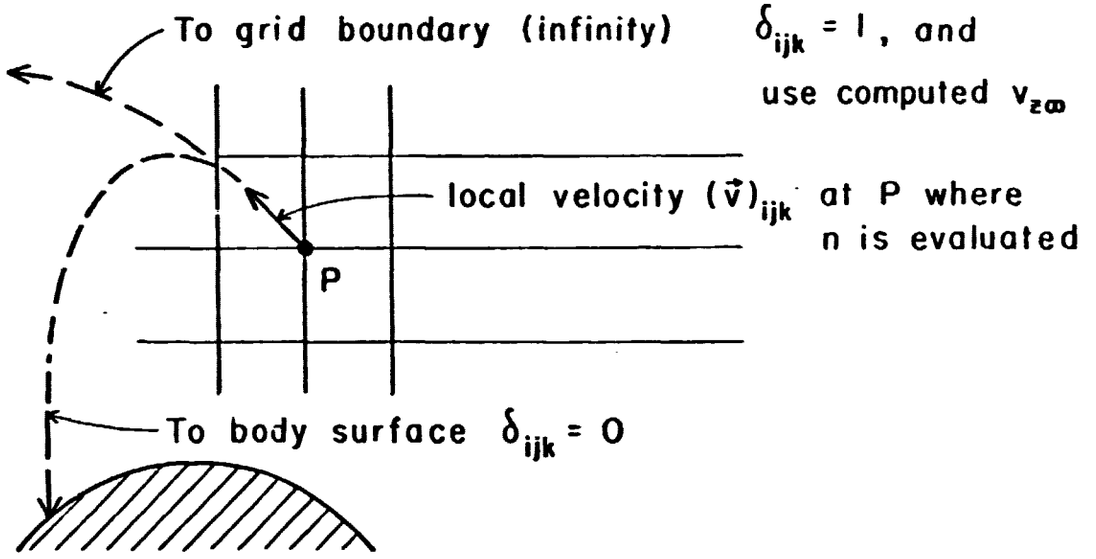
(velocities in units of $\sqrt{2kT/m}$,
v_z = axial component of velocity)

where $v_\infty^2 = v^2 + \phi$ may be identified with the total energy E, and $v_{z\infty}$ with $\sqrt{E}$ times the cosine of the angle between $\vec{v}_\infty$ and the axis. The moment integral (2-1) for number density may be approximated by a quadrature sum as follows:

$$n = \int\int\int \delta f_\infty \, d^3\vec{v} = \sum_i^I \sum_j^J \sum_k^K A_{ijk} \, \delta_{ijk} \, (f_\infty)_{ijk} \qquad (2\text{-}6)$$

where $d^3\vec{v}$ is a short-hand notation for the element $dv_x dv_y dv_z$, and δ is a cut-off (or step) function, equal to unity or zero according as the trajectory is found to come from infinity or the body surface, respectively. In the sum, the three indices refer to discrete values of three components of velocity, where the values are chosen in accord with a quadrature scheme (Gaussian), and the coefficients A_{ijk} are proportional to the associated weights. Each term in the sum represents an individual trajectory. A similar sum is obtained for the current density (see Appendix A).

Figure 2-1 indicates schematically how one of the trajectories (with indices i, j, k) from the sum in Eq. (2-6) is traced backward from the



Evaluation of δ_{ijk} for (i,j,k)-th trajectory by following (reversible) trajectories backward in time.

FIG. 2-1. INSIDE-OUT METHOD

point P (usually, a grid point), and found either to reach the body surface, or to reach "infinity" at the boundary of the grid.

This constitutes the inside-out method of solution of the Vlasov problem. Further details including the discrete velocities and coefficients of the sum are given in Appendix A.

The advantages and disadvantages of the method are:

Advantages of Inside-Out Method

1. Density points can be chosen individually and at random. Hence the method is flexible.

2. Suitable for electrons as well as ions.

Disadvantage (relatively minor)

Information carried by trajectories is lost upon moving to another density point. Hence the calculation tends to be time-consuming.

The inside-out method was developed by Parker (1964), and has subsequently been used by Fournier (1971) and by Grabowski and Fischer (1975) to calculate the wake of an infinitely-long moving circular cylinder. Grabowski and Fischer also assumed complete quasineutrality, thus restricting the generality of their method. It was also used by Taylor (1967) for the wake of an infinitely-long cylinder of rectangular cross-section (a "thick strip"), but the calculation was not carried beyond the first iteration, and is therefore not self-consistent. Parker and Whipple (1967, 1970) have used the method for two-electrode probes on a satellite, and Parker (1970, 1973) has used the method for two-electrode rocket-borne and laboratory probe systems, and for the problem of a small probe in the sheath of a large electrode.

2B. OUTSIDE-IN METHODS

Outside-in methods may be divided into two types, the method of flux-tubes and the method of superparticles or weighted deposition. The two

types of outside-in methods are illustrated in Fig. 2-2. The trajectories are injected from the outer boundary of a grid.

In the method of flux tubes, the flux of particles in a tube is constant. The tube is defined by two neighboring trajectories. Since the cross-sectional area of the tube is known from the trajectory calculation, and the particle velocity is also known, the particle density may be determined at any point in the tube (as indicated in the figure). The density is usually assigned to the nearest grid point along the path of the tube.

The advantages and disadvantages of the flux-tube outside-in method are:

Advantages of Method

A relatively fast calculation.

Disadvantages

1. Invalid if trajectories cross or reverse direction. Hence the region near the body's wake surface cannot be treated.
2. Suitable only for an axisymmetric body with cold ions in a beam.

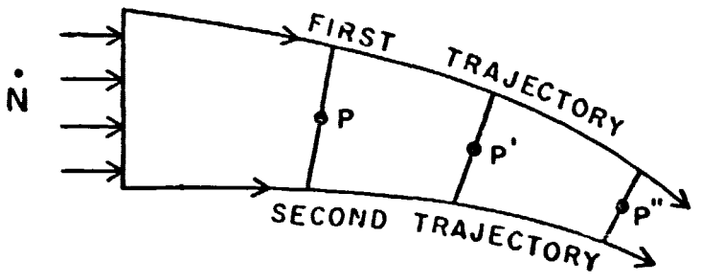
The flux-tube technique has been used by Davis and Harris (1961) for the cold-ion wake of a sphere, by Call (1969) for the cold-ion wakes of a strip, disk, infinitely-long cylinder, and sphere, by Martin (1974) for the cold-ion wakes of a strip and a disk, and by McDonald and Smetana (1969) for the wake of an infinitely-long cylinder in a monoenergetic-ion plasma with drift.

In the method of weighted deposition, the space is divided into cells, with each cell associated with one of the grid points. The contribution of a trajectory to the density in the cell is proportional to the time spent in passing through the cell.

The advantages and disadvantages of the method of weighted deposition are:

METHOD OF FLUX TUBES

Tube defined by two neighboring trajectories



$\dot{N}$ = number entering tube per second at outer boundary of grid.

$= nAv$ = constant

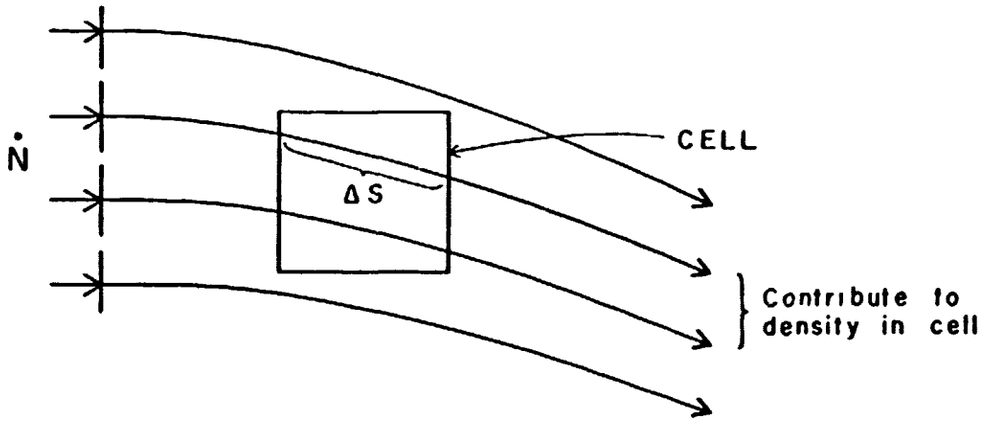
A, A', A" = cross-sectional area at points P, P', P"

v, v', v" = local speed at P, P', P"

$$n_P = \frac{\dot{N}}{Av}, \quad n_{P'} = \frac{\dot{N}}{A'v'}, \quad n_{P''} = \frac{\dot{N}}{A''v''}$$

(Method assumes n = constant on cross-section of tube)

METHOD OF SUPERPARTICLES OR WEIGHTED DEPOSITION



Contribution to density in cell proportional to time ($\Delta t = \Delta s/v$) spent in cell.

$$\Delta n = \frac{\dot{N}\,\Delta t}{(\text{volume of cell})} = \frac{\dot{N}\,\Delta s}{v \times (\text{volume of cell})}$$

(method assumes n = constant within cell)

FIG. 2-2. OUTSIDE-IN METHODS

Advantages of Method

1. No difficulty with trajectory crossings.
2. Related to and adaptable to time-dependent computer simulation.

Disadvantage

Many trajectories needed for good statistics within cells.

This method was studied by Parker (1964) for a monoenergetic-ion distribution with drift, and was used by Maslennikov and Sigov (1965) for the cold-ion wake of a sphere.

2C. OTHER METHODS

Other methods include approximate treatments which avoid trajectory calculations. Liu and Jew (1968, 1969) assumed that the ion axial component of velocity is constant. They then determined limiting trajectories for the density integral by further approximations, namely, an additional assumed approximate constant of the motion, evaluated using the local field in the vicinity of the point in question. They applied their method to the wakes of a sphere and a cylinder. Kiel, Gey, and Gustafson (1968) treated the wake of a sphere, assuming neutral ion trajectories (straight-line paths, neglecting the electric field). They also assumed that the electron densities were given by approximate formulas designed to include the effect of the potential barrier in the wake. For the wake of a strip, a disk, and an infinitely-long cylinder, Gurevich et al. (1969) assumed quasineutrality, with ion and electron densities both equated to the Boltzmann factor. In addition, they assumed that the ion axial component of velocity is constant and that the ion Mach number is large.

CHAPTER 3

SELF-CONSISTENT SOLUTIONS BY POISSON-VLASOV ITERATION

The "inside-out" method for obtaining ion and electron densities, in a given electric field defined by the values of the electric potential at a chosen set of grid points, has been described in Chapter 2, with computational details given in Appendix A. This constitutes the "Vlasov problem." The Vlasov problem must be solved separately for the electrons and each species of ions (when there is more than one). In going from one species to another, or to electrons, the potentials are multiplied by the appropriate scale factor.

How the electric field is obtained when the ion and electron densities are given is discussed in detail in Appendix B. Here, the Poisson equation is replaced by a set of difference equations based on the chosen set of grid points, with one equation for each unknown potential at a grid point. The derivation of the coefficients of the unknown potentials in the difference equations, and the method of solution, are given in Appendix B. The system of simultaneous equations for the unknown potentials is solved by a relaxation procedure. This constitutes the "Poisson problem."

The boundary conditions for the potentials in the Poisson problem are as follows. At points representing the body surface, the normalized potential is fixed at the chosen value ϕ_0. At the external (boundary) points of the grid, where "infinity" is represented on the computer, a "floating" condition is optionally used, namely, a linear relation between ϕ and $\partial\phi/\partial n$, the normal component of $\nabla\phi$. The exact relation of ϕ to $\partial\phi/\partial n$ is not important when the external boundary of the grid is sufficiently far away. (For the calculations to be reported, the assumed relation was the same as for a Coulomb potential.) In any case, either the fixed condition $\phi = 0$ or the floating condition will give the same results, provided the grid boundary is moved sufficiently far out. The effects of various types of boundary conditions representing "infinity" have been studied by Taylor (1967), and by Parker and Sullivan (1974). In general, the floating condition appears to be computationally more efficient than the fixed one. Of

course, the floating condition becomes ideal when the true relation between ϕ and $\partial\phi/\partial n$ is used, but this requires that the asymptotic form of the solution be known in advance. (See, for example, Parker and Whipple (1970).)

An iteration method is used for computing self-consistent charged-particle and potential distributions. This is herein referred to as the "Poisson-Vlasov iteration." Two principal options are employed for this procedure. In one of the options, the "charge-density" option, the space charge is initially and arbitrarily assumed to be zero. For this case, one obtains the Laplace (space-charge-less) electric field from the Poisson problem. This is the "zero-order" potential distribution, which becomes input to the Vlasov problem. The resulting solution of the Vlasov problem yields the ion and electron densities at the grid points, which are combined to make "zero-order" charge densities. These become input to the next Poisson problem, which then yields the "first-order" potentials, and so on. In this procedure one usually "mixes" successive charge-density iterates to improve stability. Otherwise the process can "blow up." One can also mix potential iterates rather than densities if desired. The dependence of the stability and convergence of the above procedure on the mixing parameter have been studied analytically by Parker (1970) and Parker and Sullivan (1974). (No other analysis of this type has been published to the author's knowledge.) This (charge-density) option is most effective when the spatial region of interest is not many Debye lengths across. The analysis shows that one can (probably always) choose a mixing parameter sufficiently small to ensure convergence, but at the expense of additional iterations.

In the other option, the "ion-density option," the ion density distribution alone is assumed initially. Initial guesses which can be employed include (a) zero ion density everywhere, (b) unit ion density (the ambient value) everywhere, and (c) the neutral ion density which obtains when there are no forces. Whichever choice is made for the initial guess is designated the "zero-order" ion density. Now if one assumes the electron density to be given by the Boltzmann factor $\exp(\phi)$, the Poisson equation may

solved, holding the ion densities fixed, but regarding both the potentials and the electron densities at the grid points as unknowns. This is a nonlinear problem, which is solved by a modification of the relaxation procedure used for the "charge-density" option. The new procedure is an important advance since the iteration is not as sensitive (tending to blow up) to small Debye numbers as in the charge-density option. Thus, very large bodies (in multiples of the Debye length) can be treated. This has been the method used to obtain the results reported in Chapter 4. Similar ideas have been used by Call (1969) and Fournier (1971), but these workers have not treated large bodies.

The assumption that the electron density is given by the Boltzmann factor becomes invalid when the body surface potential is near zero, or when there is a potential barrier in the wake such that the wake potentials are more negative than the surface potential (causing electrons to be attracted to the surface rather than repelled from it). In this case it is still possible to use the ion-density option, with its large-body capability, provided that, within each cycle, where the ion densities are held fixed, a "minor iteration" is carried out such that the electron densities are computed by trajectory calculations.

CHAPTER 4

SAMPLE RESULTS

Calculations were made for three sample problems, using the computer program listed in Appendix C, based on the theory of this report. The results presented here are preliminary in the sense that they are intended as an illustration of the capability of the program, rather than a systematic study. The body is assumed to be a circular disk with its plane normal to the flow, and the problem is specified by the dimensionless physical parameters ϕ_o, M, and λ_D, defined by (Chapter 2):

$$\phi_o \equiv e\Phi_c/kT$$

$$M \equiv mv_o^2/2kT$$

$$\lambda_D \equiv \underline{\lambda}_D/r_o$$

where T is either the ion or electron temperature (assuming equal temperatures), Φ_o is the disk potential, v_o is the disk velocity, m is the ion mass (M is assumed to be negligible for the electrons), r_o is the disk radius, and $\underline{\lambda}_D$ is the dimensional Debye length.

Numerical parameters for the calculations include 89 grid points, distributed mostly in the wake region, and in most cases 512 trajectories per grid point (8 values each for the polar and azimuthal angles, and 8 values for the energy; see Appendix C).

The potential was set to zero on the downstream computational boundary, and was allowed to "float" on the upstream and side boundaries. (The boundary conditions at the various outer grid surfaces can be either fixed or floating.) The downstream boundary for the present calculations was set at 6 radii, i.e., beyond the Mach number of radii for the two problems with M=4. The electron density n_e was assumed to be given by the Boltzmann factor $\exp(\phi)$. This is reasonable for ϕ_o=-4 on the surface, and

leads of course to computer economy (by avoiding trajectory calculations for electrons). However, it must be emphasized that this does not represent an essential restriction; the program is specifically designed to compute n_e, as well as n_i, realistically when necessary, by trajectory calculations. Moreover, in cases where a potential well occurs in the wake near the surface, the Boltzmann-factor assumption becomes invalid and the trajectories must then be computed, at least for points near the surface.

With the Boltzmann-factor assumption, an option is available in the Poisson-solution part of the program. This is the "ion-density" option (Chapter 3) which includes the Boltzmann factor in the equations for the potential distribution; the equations thereby become nonlinear rather than linear. This is a simplified case of a possible general technique where, during each "major" iteration cycle in which the ion densities are held fixed, self-consistent potentials and electron densities are simultaneously determined. This technique is as yet in an experimental stage, but it seems promising in that it may produce solutions with reasonable costs for large-body problems; in such problems, the conventional Poisson-Vlasov iteration based on the "charge-density" option (Chapter 3) becomes expensive (Parker and Sullivan, 1974). A disadvantage of the ion-density option, however, is that its convergence properties are not understood; therefore its costs are difficult to predict. This is in contrast to the case of the charge-density option where an analysis is available (Parker and Sullivan, 1974).

The three calculations to be described next were all made with the ion-density option. The cases are:

(a) $\phi_o = -4$, $M = 4$, $\lambda_D = 1/5$

(b) $\phi_o = -4$, $M = 4$, $\lambda_D = 1/100$

(c) $\phi_o = -4$, $M = 8$, $\lambda_D = 1/100$

Transverse profiles of normalized ion density (n_i), electron density (n_e), and potential (ϕ) in the wake region downstream of the disk are shown below for the three cases. The profiles are in transverse planes

at various distances downstream, and all lengths are normalized by the disk radius. (z denotes axial distance downstream, in radii, with z=0 defined as the plane of the disk, and r denotes transverse, i.e., radial, distance from the axis.) The profiles are presented in Figs. 4-1 to 4-3, and in Tables 4-1 to 4-9.

4.1 Case $\phi_0 = -4$, $M = 4$, and $\lambda_D = 1/5$

For this case (Fig. 4-1, and Tables 4-1 to 4-3), the parameter values illustrate a class of problems of physical interest, such as a small TAD at high altitudes, or a probe mounted on or near a spacecraft.

Twelve major iterations (Poisson-Vlasov cycles) were computed, in which successive ion-density iterates were mixed, with a mixing parameter 0.5, starting with uniform ambient ion density as an initial guess. In the last four iterations, 8192 trajectories were used at points at and near the wake axis. This increase of the number of trajectories by a factor 16 was made to increase the accuracy in the investigation of possible structure (enhancements or depletions in ion or electron density) near the axis.

Figure 4-1 shows three sets of profiles, one for n_i, one for n_e, and one for ϕ. Within each set, the profiles are arranged vertically in order of increasing axial distance z. There are eight values of z, namely, z = 0.2, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0 and 6.0. Each profile is constructed using nine values of r, namely, r = 0, 0.1, 0.3, 0.6, 0.8, 1.0, 1.2, 1.5, and 2.0, with straight-line segments connecting the values of the functions (n_i, n_e, or ϕ) computed at these points. (The nine values of r and eight values of z are the coordinates of the 72 grid points chosen to represent the wake region of this problem, for which a total of 89 points were used.)

The ϕ-profiles (right side of figure) and the n_e-profiles (middle of figure) are the 12th-order iteration values. These ϕ-values are also given in Table 4-3, and the n_e-values are given, in parentheses, in Table 4-1. On the left side of the figure, there are two sets of n_i-profiles, one labelled "A", and the other unlabelled. The "A"-profiles for n_i are the 11th-order values (Table 4-2), from which the ϕ-profiles (and n_e-profiles) in the figure are derived. The unlabelled profiles for n_i are the 12th-order n_i-values (Table 4-1) resulting from trajectory calculations using the ϕ-profiles (Table 4-3). The juxtaposition of the "A" and unlabelled n_i-profiles indicates the extent to which the Poisson-Vlasov

iteration has converged. At z = 1 and below, the two profiles are sufficiently close to be considered well converged for the present purposes. Tables 4-1 and 4-2 give the numerical values of the 12th and 11th orders of n_i (unlabelled and "A"), respectively.

The convergence is more complete at some points than at others. The non-convergence at z = 2 and beyond seems to be small-amplitude numerical "noise," but the over-all solution is sufficiently well defined for the present purposes. The difficulty of the convergence at and beyond z = 2 may be associated in part with insufficiencies in the numbers of grid points, numbers of trajectories, and individual trajectory accuracy; improvements in these parameters requires more computer time. At points along the axis more trajectories were used than at points off the axis, so that the accuracy is relatively high along the axis. In contrast to off-axis points, the convergence is clearly excellent at the axial points beyond z = 1. The accuracy of n_i at off-axis points far downstream where ϕ is small is estimated to be about 10 percent.

Tables 4-1 and 4-2 also give the dimensionless ion and electron current density (j_i and j_e) at the center of the wake side of the disk. (j_e is seen to have the value exp(-4).) The electron density (n_e) profiles are given in Fig. 4-1 (middle profiles) and in Table 4-1 (in parentheses). n_e is assumed to be unity (ambient value) at the downstream boundary z = 6 where ϕ is assumed to be zero. Figure 4-1 and Table 4-1 indicate that quasineutrality ($n_i = n_e$) is roughly valid within the accuracy of the calculation at and beyond z = 4.

The features of the wake structure are as follows: There is no prominent structure such as large-amplitude ion or electron density bumps. The apparent structure in the ion profiles at z = 2 and beyond suggests the possibility of small-amplitude ion structure near the axis, beyond z = 2. The apparent ion structure near the axis at z = 5 is clearly not associated with the local potential profile (since ϕ is essentially zero in this region); hence this structure must be associated with upstream perturbations, i.e., the deflection of ion trajectories passing near the edge of the disk. It is also evident that the downstream boundary at z = 6

has been chosen sufficiently for downstream. This evidence is based on the smoothness with which the potential has already fallen off to negligible values at $z = 5$.

The reality of the small-amplitude structures (as opposed to iterative noise) can be verified by more accurate calculations, with changes in the numerical parameters such as numbers of grid points and trajectories. Persistence of the structure despite changes in the numerical parameters may be taken as an indication of its reality. In spite of this uncertainty, the important gross features clearly indicated by the profiles are that (a) the wake becomes filled in by electrons and ions somewhere between 2 and 3 radii downstream, i.e., less than the Mach number of radii, and (b) the wake disturbance is essentially confined within a region extending to about $z = 4$ downstream, and outward to about $r = 1.5$ in the transverse dimension.

4.2 Case $\phi_0 = -4$, $M = 4$, and $\lambda_D = 1/100$

For this case (Fig. 4-2 and Tables 4-4 to 4-6), the parameter values differ from those of the preceding problem only in λ_D, which is very small so that the problem applies to a large body, namely 100 Debye lengths in radius. This size of moving body is larger than has been previously treated by trajectory-following, i.e., realistic, calculations. (In the large-body calculations of Kiel et al. (1968), the particle trajectories were not treated realistically.) The results show what may be expected for the wake structure of large bodies in general. This case requires more effort (computer time and judicious selection of numerical parameters) than that of a smaller body. The solutions shown, therefore, are intended to be illustrative rather than accurate.

Six iterations, or Poisson-Vlasov cycles, were computed in which successive iterates were used without mixing, starting with the neutral ion density as an initial guess. The nominal number of trajectories, 512, was used at all grid points.

The profiles of n_i, n_e, and ϕ in Fig. 4-2 are constructed in the same way and at the same grid points as in Fig. 4-1. The wake is essentially "empty" of both ions and electrons between $z = 0$ and $z = 1$, and begins to fill up between $z = 2$ and $z = 3$. In this way, the wake is qualitatively similar to that in Fig. 4-1.

Again, two sets of ion-density profiles are shown on the left side of Fig. 4-2, the unlabelled profiles for the 6th order (6th iteration, Table 4-4), and the profiles labelled "A" for the 5th order (Table 4-5). The 6th-order potentials are given in Table 4-6, and the 6th-order n_e-values are given, in parentheses, in Table 4-4. Comparison of the n_e-values in Table 4-4 with the 5th-order n_i-values (labelled "A") in Table 4-5 indicate that the quasineutrality assumption is valid everywhere outside a cone-shaped region near the wake surface; the cone height along the axis is between one and two radii. This is in accord with expectation for a large body. Near the wake surface, however, quasineutrality is violated because the effective Debye length is large. The similarity of the n_i-profiles (labelled "A") and the n_e profiles in Fig. 4-2 is a consequence of near-quasineutrality.

Comparison of the 5th and 6th order n_i-profiles (labelled "A" and unlabelled) in Fig. 4-2 show that the solution is reasonably converged for z = 1 and below, but that there is incomplete convergence at z = 2 and beyond. The incomplete convergence and apparent structure at z = 2 and beyond may be artifactual due to insufficient numerical accuracy. (No attempt was made to achieve high accuracy since this was regarded as a trial run.) The structure and lack of convergence are seen to extend past z = 5, so that the downstream boundary should be placed further than at z = 6.

Compared with the previous case (and despite any inaccuracies), one may infer additional physical conclusions indicated by Fig. 4-2, namely, (a) the suggestion of a core of high (approximately ambient) density of ions and electrons on the axis, and (b) the occurrence of a potential well in the near wake, defined as a region with ϕ-values below -4. The shading in the two lowest ϕ-profiles of Fig. 4-2 denote cross-sections of this well. The wake-surface current densities (Table 4-4) are less than in the previous case; the electron current density is less than exp(-4), as would be expected in the presence of a potential well.

The region of wake disturbance is not as well defined as in the previous case of the smaller body, but it probably extends more than 6 radii downstream, and between 2 and 3 radii in the transverse direction.

4.3 Case $\phi_0 = -4$, M = 8, and $\lambda_D = 1/100$

This case (Fig. 4-3 and Tables 4-7 to 4-9) is another large-body case similar to the previous large-body case except that the Mach number is increased from M = 4 to M = 8. Ten iterations were computed in which successive iterates were used without mixing, starting with uniform ambient ion density. (The latter starting condition was inadvertently different from that of the M = 4 calculation which was started with the neutral ion density, but this difference should become unimportant after many iterations.) Similar statements may be made about the incompleteness of the convergence as in the M = 4 case. The 9th and 10th-order ion densities are labelled "A" (Table 4-8) and unlabelled (Table 4-7), respectively. On comparing these, the convergence seems fairly good at z = 0.5 and z = 1. Again, the disturbance extends beyond z = 5, so that the downstream boundary should be moved further than z = 6.

Despite inaccuracies, the consistency is such that physical conclusions may be drawn as follows. In this case the wake is seen to remain empty further downstream than in the M = 4 case. In addition, the suggestion is much stronger that there is a central core of ambient density for both ions and electrons along the axis. Moreover, the potential well is wider and longer than in the M = 4 case, although the depth is about the same. The disk-wake-surface electron current density (Table 4-7) is slightly less than the M = 4 value, and is again less than exp(-4).

The conical region behind the disk where quasineutrality breaks down is now longer than in the M = 4 case, extending to between z = 4 and z = 5 along the axis.

The region of wake disturbance is probably longer than 6 radii downstream, as in the M = 4 case, but may not extend beyond about 2 radii in the transverse direction.

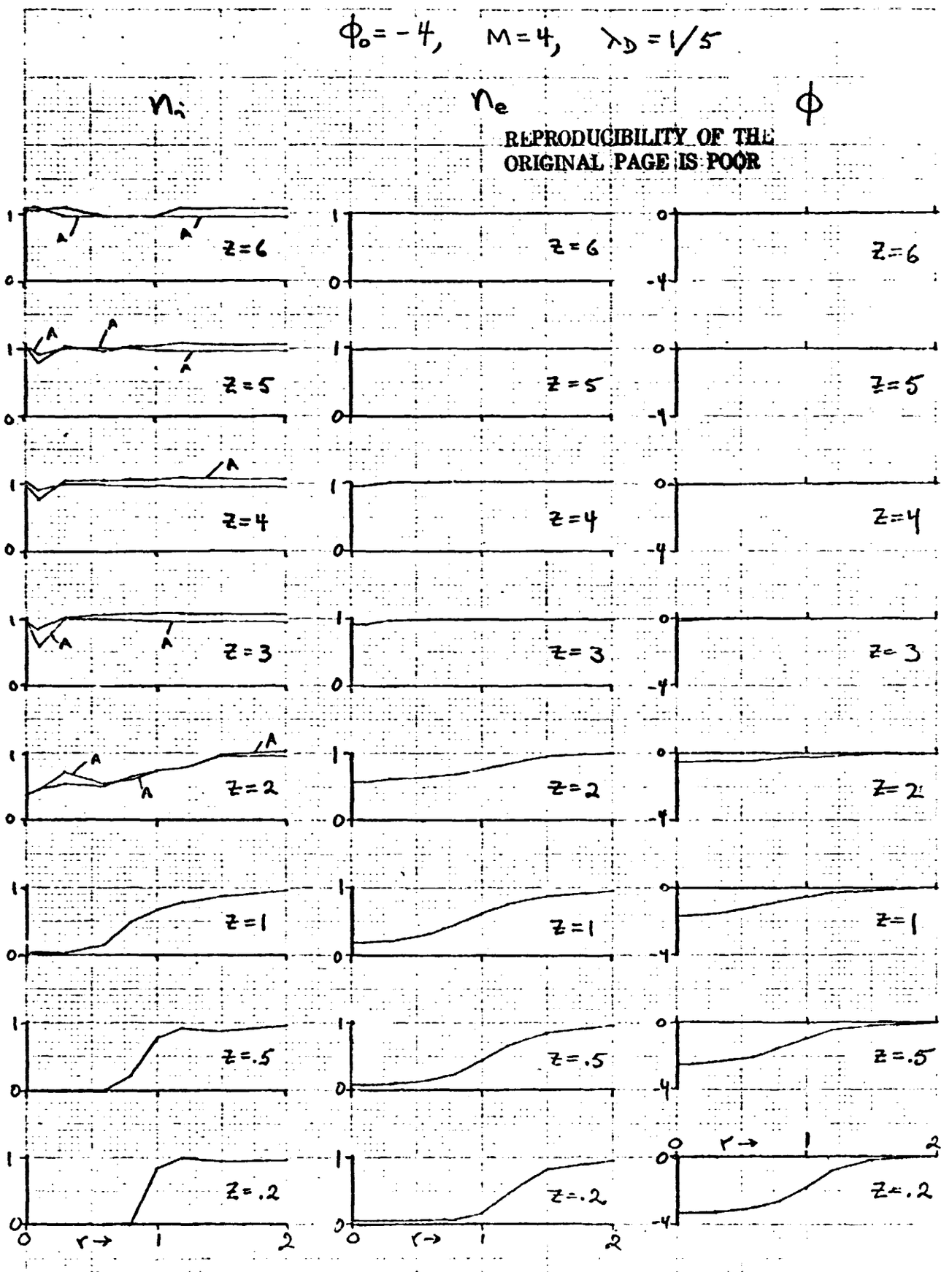


FIG. 4-1. DISK WAKE PROFILES

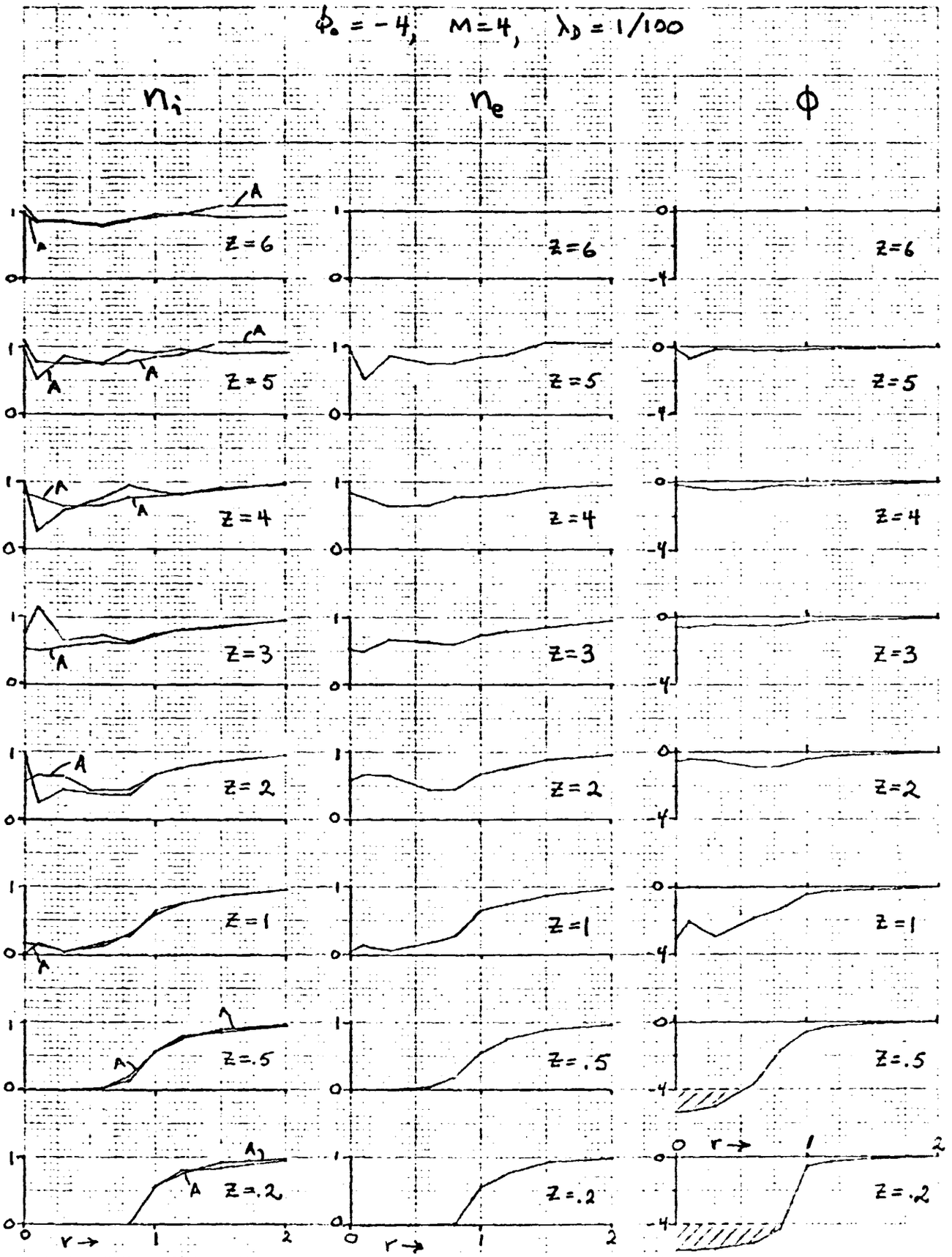


FIG. 4-2. DISK WAKE PROFILES

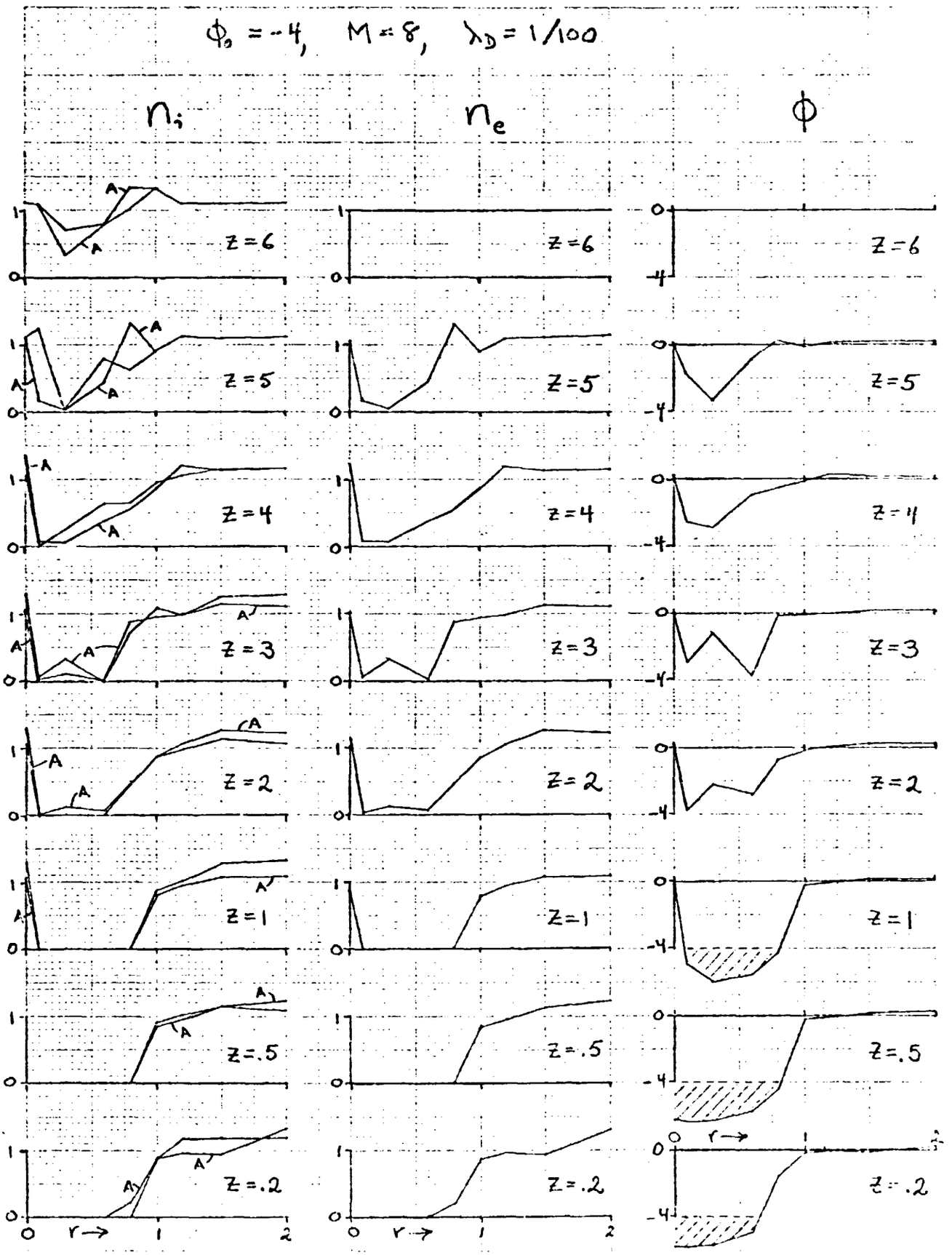


FIG. 4-3. DISK WAKE PROFILES

TABLE 4-1

NORMALIZED ION (AND ELECTRON) DENSITY IN WAKE
($\phi_0 = -4$, $M = 4$, $\lambda_D = 1/5$)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	1.03 (1.00)	1.03 (1.00)	1.09 (1.00)	0.95 (1.00)	0.95 (1.00)	0.95 (1.00)	1.09 (1.00)	1.09 (1.00)	1.09 (1.00)
5	1.03 (0.98)	0.79 (0.98)	1.02 (1.00)	0.95 (1.00)	1.01 (1.00)	1.02 (1.00)	1.08 (1.00)	1.06 (1.00)	1.06 (1.00)
4	1.01 (0.95)	0.89 (0.95)	1.00 (1.01)	0.99 (1.02)	0.96 (1.03)	0.97 (1.03)	0.95 (1.03)	0.95 (1.03)	0.95 (1.03)
3	0.96 (0.91)	0.82 (0.91)	1.01 (0.97)	1.06 (0.99)	1.07 (0.99)	1.09 (0.99)	1.09 (0.99)	1.07 (0.99)	1.07 (0.99)
2	0.39 (0.58)	0.46 (0.58)	0.53 (0.61)	0.51 (0.63)	0.64 (0.69)	0.73 (0.75)	0.79 (0.83)	0.96 (0.96)	0.95 (1.00)
1	0.016 (0.19)	0.018 (0.19)	0.016 (0.21)	0.15 (0.31)	0.49 (0.45)	0.68 (0.61)	0.79 (0.74)	0.88 (0.87)	0.98 (0.96)
0.5	0.00034 (0.084)	0.00037 (0.085)	0.00047 (0.094)	0.0035 (0.14)	0.20 (0.23)	0.77 (0.42)	0.91 (0.65)	0.88 (0.84)	0.97 (0.95)
0.2	2.3×10^{-5} (0.036)	2.2×10^{-5} (0.036)	2.6×10^{-7} (0.038)	2.0×10^{-5} (0.049)	0.00039 (0.071)	0.82 (0.15)	1.00 (0.46)	0.93 (0.81)	0.97 (0.94)

r, z in units of disk radius

IT = 12, 9/8/75, $\alpha = .5$

$j_i = 9.6\times10^{-6}$, $j_e = 0.0183$

$\phi = 0$ at z_N

TABLE 4-2

NORMALIZED ION DENSITY IN WAKE (A)
($\phi_o = -4$, $M = 4$, $\lambda_D = 1/5$)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	1.04	1.04	0.95	0.95	0.95	0.95	0.95	0.95	0.95
5	1.04	0.91	1.00	1.00	1.00	0.99	0.97	0.97	0.97
4	1.00	0.77	1.05	1.05	1.08	1.06	1.09	1.07	1.07
3	0.96	0.59	0.99	0.99	0.98	0.97	0.95	0.96	0.94
2	0.40	0.45	0.71	0.54	0.60	0.72	0.79	0.98	1.03
1	0.017	0.018	0.016	0.15	0.50	0.67	0.79	0.88	0.98
0.5	0.00035	0.00037	0.00048	0.0035	0.19	0.77	0.91	0.90	0.98
0.2	0.000022	0.000022	2.6×10^{-7}	0.000020	0.00039	0.82	1.00	0.93	0.97

$j_i = 9.4\times10^{-6}$, $j_e = 0.0183$

IT = 11, 9/8/75, $\alpha = .5$

$\phi = 0$ at z_N

TABLE 4-3

NORMALIZED POTENTIAL IN WAKE
(ϕ_o = -4, M = 4, λ_D = 1/5)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	-0.016	-0.019	-0.00046	0.0016	0.0016	7.5×10^{-5}	-0.0037	-0.0041	-0.0027
4	-0.046	-0.048	0.0075	0.022	0.030	0.029	0.033	0.030	0.026
3	-0.096	-0.099	-0.027	-0.011	-0.0073	-0.0058	-0.0064	-0.0053	-0.0083
2	-0.54	-0.54	-0.49	-0.47	-0.37	-0.28	-0.18	-0.037	0.0013
1	-1.66	-1.64	-1.54	-1.17	-0.79	-0.50	-0.30	-0.14	-0.045
0.5	-2.48	-2.46	-2.36	-1.98	-1.49	-0.86	-0.42	-0.17	-0.055
0.2	-3.33	-3.32	-3.26	-3.03	-2.65	-1.87	-0.78	-0.21	-0.059

TABLE 4-4

NORMALIZED ION (AND ELECTRON) DENSITY IN WAKE
($\phi_0 = -4$, $M = 4$, $\lambda_D = 1/100$)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	1.08 (1.00)	0.87 (1.00)	0.87 (1.00)	0.80 (1.00)	0.88 (1.00)	0.91 (1.00)	0.98 (1.00)	0.91 (1.00)	0.92 (1.00)
5	1.09 (0.91)	0.79 (0.52)	0.75 (0.85)	0.78 (0.75)	0.95 (0.75)	0.91 (0.84)	0.97 (0.89)	0.90 (1.07)	0.91 (1.06)
4	1.00 (0.81)	0.28 (0.76)	0.58 (0.63)	0.76 (0.65)	0.95 (0.78)	0.85 (0.79)	0.81 (0.81)	0.89 (0.91)	0.99 (0.94)
3	0.76 (0.52)	1.12 (0.50)	0.65 (0.66)	0.72 (0.62)	0.63 (0.60)	0.75 (0.72)	0.81 (0.80)	0.88 (0.85)	0.95 (0.94)
2	0.96 (0.57)	0.24 (0.65)	0.44 (0.63)	0.36 (0.42)	0.36 (0.43)	0.67 (0.67)	0.76 (0.76)	0.85 (0.87)	0.95 (0.95)
1	0.14 (0.044)	0.12 (0.13)	0.039 (0.049)	0.11 (0.16)	0.30 (0.27)	0.59 (0.64)	0.75 (0.74)	0.86 (0.86)	0.96 (0.97)
0.5	0.00020 (0.0047)	0.00098 (0.0050)	0.0037 (0.0063)	0.019 (0.024)	0.13 (0.19)	0.57 (0.55)	0.79 (0.73)	0.84 (0.89)	0.94 (0.97)
0.2	0.0095 (0.0038)	2.7×10^{-5} (0.0038)	1.3×10^{-5} (0.0041)	2.8×10^{-5} (0.0059)	0.00054 (0.013)	0.56 (0.56)	0.80 (0.75)	0.83 (0.91)	0.94 (0.98)

$j_i = 2.4\times10^{-7}$, $j_e = 0.0043$

IT = 6, 9/19/75

$\phi = 0$ at z_N

TABLE 4-5

NORMALIZED ION DENSITY IN WAKE (A)
(ϕ_o = -4, M = 4, λ_D = 1/100)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	0.94	0.84	0.84	0.77	0.86	0.96	0.94	1.09	1.09
5	0.94	0.52	0.86	0.75	0.75	0.84	0.89	1.07	1.06
4	0.81	0.76	0.63	0.65	0.78	0.79	0.81	0.91	0.94
3	0.52	0.50	0.66	0.62	0.60	0.72	0.80	0.85	0.94
2	0.56	0.65	0.63	0.42	0.43	0.67	0.76	0.87	0.95
1	0.00063	0.14	0.046	0.16	0.27	0.64	0.74	0.86	0.97
0.5	0.0017	0.0025	0.0033	0.021	0.19	0.56	0.73	0.89	0.97
0.2	0.000035	5.4×10^{-7}	8.9×10^{-6}	0.000033	0.00050	0.57	0.75	0.91	0.98

$j_i = 1.1\times10^{-8}$, $j_e = 0.0044$

IT = 5, 9/19/75

$\phi = 0$ at z_N

TABLE 4-b

NORMALIZED POTENTIAL IN WAKE
($\phi_o = -4$, $M = 4$, $\lambda_D = 1/100$)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	-0.090	-0.65	-0.16	-0.29	-0.29	-0.17	-0.12	0.065	0.057
4	-0.21	-0.28	-0.46	-0.44	-0.24	-0.24	-0.21	-0.095	-0.057
3	-0.65	-0.68	-0.42	-0.48	-0.51	-0.33	-0.22	-0.17	-0.058
2	-0.56	-0.44	-0.46	-0.87	-0.83	-0.40	-0.28	-0.14	-0.047
1	-3.12	-2.03	-3.01	-1.86	-1.30	-0.45	-0.30	-0.15	-0.031
0.5	-5.35	-5.30	-5.07	-3.72	-1.67	-0.59	-0.32	-0.12	-0.032
0.2	-5.57	-5.56	-5.49	-5.14	-4.32	-0.58	-0.29	-0.095	-0.022

TABLE 4-7

NORMALIZED ION (AND ELECTRON) DENSITY IN WAKE
($\phi_0 = -4$, $M = 8$, $\lambda_D = 1/100$)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	1.11 (1.00)	1.09 (1.00)	0.71 (1.00)	0.80 (1.00)	1.01 (1.00)	1.32 (1.00)	1.10 (1.00)	1.10 (1.00)	1.11 (1.00)
5	1.11 (1.04)	1.24 (0.18)	0.045 (0.038)	0.80 (0.44)	0.63 (1.30)	0.91 (0.90)	1.12 (1.10)	1.11 (1.11)	1.12 (1.12)
4	1.04 (1.21)	0.0019 (0.080	0.24 (0.057)	0.63 (0.38)	0.64 (0.54)	0.96 (0.87)	1.04 (1.20)	1.14 (1.13)	1.15 (1.14)
3	1.27 (0.93)	0.015 (0.055)	0.095 (0.31)	0.0065 (0.023)	0.71 (0.87)	1.08 (0.95)	0.99 (0.99)	1.24 (1.13)	1.27 (1.10)
2	1.04 (1.12)	0.00064 (0.022)	0.00081 (0.11)	0.0070 (0.061)	0.46 (0.47)	0.85 (0.85)	0.98 (1.05)	1.12 (1.26)	1.06 (1.21)
1	1.27 (0.85)	1.5×10^{-5} (0.0071)	9.7×10^{-13} (0.0025)	0.00019 (0.0039)	0.0014 (0.014)	0.87 (0.79)	1.01 (0.94)	1.28 (1.07)	1.32)1.08)
0.5	5.0×10^{-16} (0.0018)	4.4×10^{-15} (0.0017)	3.0×10^{-6} (0.0018)	4.6×10^{-13} (0.0034)	8.6×10^{-5} (0.013)	0.90 (0.82)	1.01 (0.95)	1.12 (1.11)	1.08 (1.21)
0.2	6.9×10^{-7} (0.0032)	1.6×10^{-7} (0.0032)	9.4×10^{-16} (0.0036)	1.3×10^{-10} (0.0077)	3.6×10^{-6} (0.20)	0.87 (0.86)	1.16 (0.96)	1.18 (0.93)	1.18 (1.30)

$j_i = 4.2\times10^{-30}$, $j_e = 0.0037$

IT = 10, 9/19/75

$\phi = 0$ at z_N

TABLE 4-8

NORMALIZED ION DENSITY IN WAKE (A)
(ϕ_o = -4, M = 8, λ_D = 1/100)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	1.11	1.07	0.33	0.79	1.33	1.33	1.10	1.10	1.11
5	1.11	0.18	0.032	0.44	1.31	0.90	1.10	1.11	1.12
4	1.31	0.075	0.052	0.38	0.54	0.87	1.20	1.13	1.14
3	1.04	0.039	0.31	0.011	0.87	0.95	0.99	1.13	1.10
2	1.27	0.0038	0.11	0.056	0.47	0.86	1.05	1.26	1.21
1	1.04	0.000017	0.000037	0.00095	0.0044	0.80	0.94	1.07	1.08
0.5	2.1×10^{-16}	4.6×10^{-15}	3.0×10^{-6}	3.8×10^{-5}	0.0014	0.83	0.95	1.11	1.21
0.2	6.9×10^{-7}	6.2×10^{-7}	4.3×10^{-7}	8.2×10^{-7}	0.21	0.87	0.96	0.93	1.31

$j_i = 7.4\times10^{-30}$, $j_e = 0.0029$

IT = 9, 9/19/75

$\phi = 0$ at z_N

TABLE 4-9

NORMALIZED POTENTIAL IN WAKE
(ϕ_b = -4, M = 8, $_D$ = 1/100)

z	r=0	r=.1	r=.3	r=.6	r=.8	r=1.0	r=1.2	r=1.5	r=2.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.039	-1.71	-3.26	-0.82	0.26	-0.10	0.094	0.11	0.12
4	0.19	-2.52	-2.87	-0.96	-0.62	-0.14	0.19	0.13	0.13
3	-0.073	-2.90	-1.18	-3.76	-0.14	-0.047	-0.015	0.12	0.094
2	0.11	-3.80	-2.24	-2.79	-0.75	-0.16	0.051	0.23	0.19
1	-0.16	-4.95	-5.98	-5.55	-4.30	-0.23	-0.059	0.070	0.081
0.5	-6.33	-6.38	-6.31	-5.69	-4.38	-0.19	-0.053	0.11	0.19
0.2	-5.75	-5.74	-5.63	-4.86	-1.60	-0.15	-0.044	-0.077	0.27

APPENDIX A

THE VLASOV PROBLEM: DENSITY CALCULATION

For the purpose of evaluating density and current-density integrals, it is convenient to transform to energy and angle variables in velocity space. Since we will be interested primarily in Maxwellian energy distributions (with drift), we adopt the following units in terms of which dimensionless variables may be defined:

kT = unit of energy, where T is the temperature of the Maxwellian distribution

$\sqrt{2kT/m}$ = unit of velocity, namely, the most probable velocity

n_0 = unit of particle density, namely, the unperturbed density

The energy and angle variables are:

E = energy in multiples of kT

α = polar angle with respect to z-axis (Fig. A-1)

β = azimuthal angle with respect to the plane containing the z-axis and the point $\vec{r}$ (Fig. A-1)

z-axis = axis of symmetry of body as well as direction of plasma flow

The definitions of the angles of α and β, which define the orientation of the velocity-vector $\vec{v}$, are illustrated in Fig. A-1. The potential energy ϕ will also be a multiple of kT.

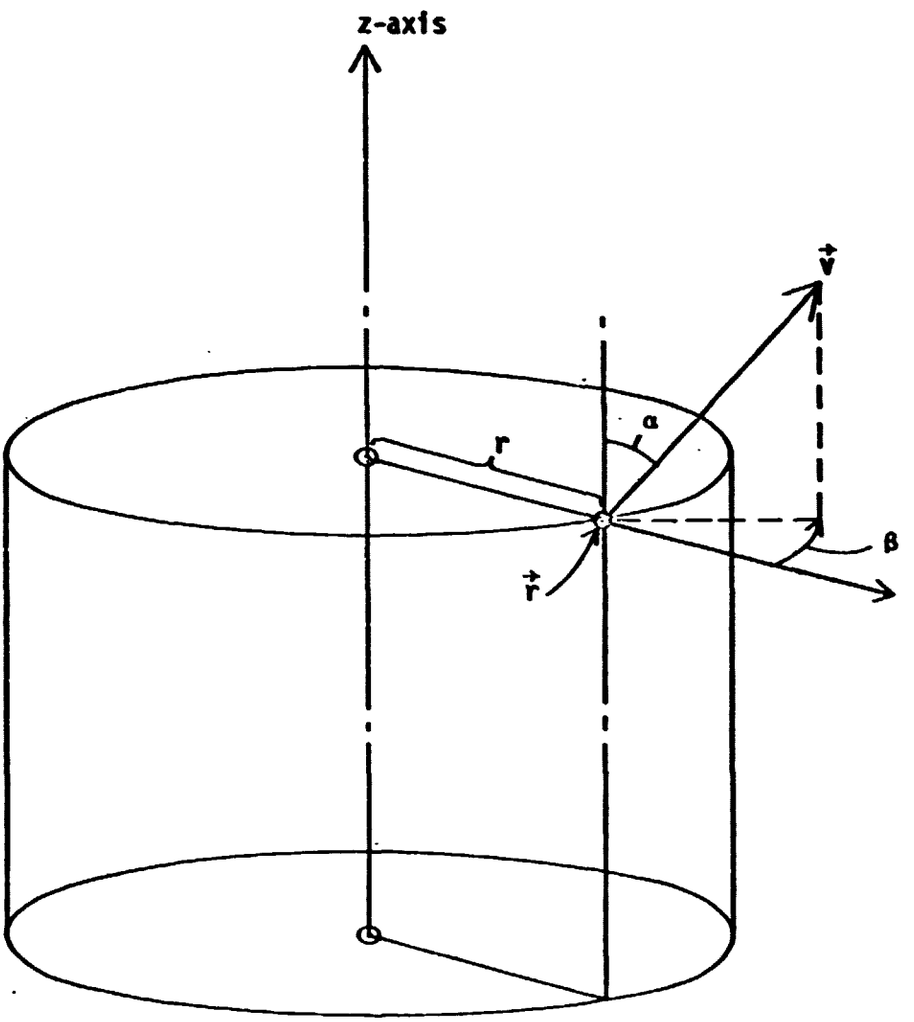


FIG. A-1. ANGLE VARIABLES IN VELOCITY SPACE

The integral for the particle density follows from appropriate transformation of the volume element in velocity space of Chapter 2, namely,

$$n = \iiint f\, v^2\, dv \sin\alpha\, d\alpha\, d\beta$$

$$= (\text{const}) \iiint f \cdot \sqrt{E - \phi}\, dE \sin\alpha\, d\alpha\, d\beta \qquad \text{(A-1)}$$

where there is a step-function δ (a "cut-off" factor) which is defined as unity if the orbit connects with infinity (an "escaping" orbit), and as zero otherwise. The step-function thus automatically takes care of the restriction to the domain of escaping orbits. However, whether δ is unity or zero is decided only after the orbit has been followed backward in time sufficiently far (by performing an orbit calculation in the given potential distribution).

For the Maxwellian distribution with drift, we have

$$n = \frac{1}{2\pi^{3/2}} \int_{\text{Max}\,(0,\phi)}^{\infty} \sqrt{E - \phi}\, dE \int_0^{\pi} \sin\alpha\, d\alpha \int_0^{2\pi} e^{-U(E,\alpha,\beta)}\, \delta\, d\beta \qquad \text{(A-2)}$$

where

$$U \equiv E + M^2 + 2M\sqrt{E}\cos\alpha_\infty \qquad \text{(A-3)}$$

with M denoting the Mach number (plasma-flow velocity divided by $\sqrt{2kT/m}$), and α_∞ denoting the value of the polar angle α of the velocity-vector at infinity.

The limits on the integral correspond to the full ranges of the variables. The lower limit, "Max(0,ϕ)," on the E-integral is defined to be zero if $\phi < 0$ (attractive potential), and to be ϕ if $\phi > 0$ (repulsive potential). For $\phi = 0$ and $\delta = 1$, α_∞ is equal to α and n becomes unity (the unperturbed value).

The current density integral may similarly be written

$$j = \int\int\int f\, v^3\, dv \cos\alpha \sin\alpha\, d\alpha\, d\beta$$

$$= (\text{const}) \int\int\int f \cdot (E - \phi) dE \cos\alpha \sin\alpha\, d\alpha\, d\beta \qquad \text{(A-3)}$$

where the normal component of velocity v_n is identified with the z-component of velocity, if the surface of interest on which the current is to be calculated is perpendicular to the z-axis. The constant in front will now be adjusted. Specializing Eq. (A-3) to the Maxwellian distribution (with drift), we may write

$$\frac{j}{j_0} = \frac{1}{\pi} \int_{\text{Max}(0,\phi)}^{\infty} (E - \phi) dE \int_0^{\pi/2} \cos\alpha \sin\alpha\, d\alpha \int_0^{2\pi} e^{-U(E,\alpha,\beta)}\, \delta d\beta \qquad \text{(A-4)}$$

The current density is expressed as the ratio j/j_0 where j_0 denotes the "random" current density (namely, $n_0\sqrt{kT/2\pi m}$) which would be collected in the absence of plasma flow and if there were no electric fields, i.e., such that $\phi = 0$ and $\delta = 1$ over the ranges of integration. In the remainder of the report it will be convenient to take j to mean j/j_0 ($j_0 \equiv 1$).

The integrals in Eqs. (A-2) and (A-4) represent infinite numbers of orbits (continuous distributions in E, α, and β). In approximating the integrals by quadratures, we replace the infinite sums by finite numbers of terms, each corresponding to an orbit. Thus, transforming the ranges of integration into intervals between -1 and +1 in preparation for the Gaussian quadrature, we may write

$$E(c) = \frac{1 + c}{1 - c} + \text{Max}(0,\phi) \qquad \text{(A-5)}$$

$$\left.\begin{aligned} \alpha &= \cos^{-1} a \qquad \text{for density} \\ \alpha &= \sin^{-1}\sqrt{\frac{1+a}{2}} \quad \text{for current density} \end{aligned}\right\} \tag{A-6}$$

$$\beta = \frac{\pi}{2}(1+b) \tag{A-7}$$

Then the transformed density and current density integrals become

$$n = \frac{1}{\sqrt{\pi}} \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} e^{-U(c,a,b)} \sqrt{E(c) - \phi} \cdot \delta \cdot \frac{dc\ da\ db}{(1-c)^2} \tag{A-8}$$

and

$$j = \frac{1}{2} \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} e^{-U(c,a,b)} [E(c) - \phi] \cdot \delta \cdot \frac{dc\ da\ db}{(1-c)^2} \tag{A-9}$$

In a case where a potential barrier exists, "Max $(0,\phi)$" in Eq. (A-5) should be replaced by the barrier height.

We now have the integrals in a form suitable for Gaussian quadratures, where the new variables (c, a, b) all lie in the range -1 to +1. For flexibility, we now divide the c-range into M_e sub-intervals, and apply a Gaussian quadrature of order 2 to each of these sub-intervals. Similarly, we divide the a-range into M_a sub-intervals, and the b-range into M_b sub-intervals, with Gaussian quadratures of order 2 in each sub-interval. Then both Eqs. (A-8) and (A-9) may be put in the form

$$I = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} T(E) \cdot \delta(E, \alpha, \beta) \cdot dc\ da\ db \tag{A-10}$$

which may be approximated by the sum:

$$I \approx S \equiv \frac{1}{M_e M_a M_b} \sum_{K_e=1}^{M_e} \sum_{K_a=1}^{M_a} \sum_{K_b=1}^{M_b} [T(E', \alpha', \beta') \cdot \delta(E', \alpha', \beta')$$

$$+ T(E'', \alpha'', \beta'') \cdot \delta(E'', \alpha'', \beta'')] \qquad \text{(A-11)}$$

where T is defined by

$$T(E) \equiv \frac{e^{-U(c,a,b)}}{(1-c)^2} \cdot \begin{cases} \frac{\sqrt{E(c) - \phi}}{\sqrt{\pi}} & \text{for density} \\ \frac{E(c) - \phi}{2} & \text{for current density} \end{cases} \qquad \text{(A-12)}$$

with

$$E' \equiv E(c'), \quad E'' \equiv E(c'')$$

$$\alpha' \equiv \alpha(a'), \quad \alpha'' \equiv \alpha(a'') \qquad \text{(A-13)}$$

$$\beta' \equiv \beta(b'), \quad \beta'' \equiv \beta(b'')$$

and with $E(c)$, $\alpha(a)$ and $\beta(b)$ defined by Eqs. (A-5) through (A-7).

The special values c', c", a', a", b', and b" are defined by formulas based on the abscissas ($\pm 3^{-1/2}$) for the Gaussian (Order-2) quadrature applied to the multiple sub-intervals, namely,

$$c' = \frac{1}{M_e}\left(-\frac{1}{\sqrt{3}} + 2K_e - 1 - M_e\right)$$

$$c'' = \frac{1}{M_e}\left(+\frac{1}{\sqrt{3}} + 2K_e - 1 - M_e\right)$$

$$a' = \frac{1}{M_a}\left(-\frac{1}{\sqrt{3}} + 2K_a - 1 - M_a\right)$$

$$a'' = \frac{1}{M_a}\left(+\frac{1}{\sqrt{3}} + 2K_a - 1 - M_a\right)$$

$$b' = \frac{1}{M_b}\left(-\frac{1}{\sqrt{3}} + 2K_b - 1 - M_b\right)$$

$$b'' = \frac{1}{M_b}\left(+\frac{1}{\sqrt{3}} + 2K_b - 1 - M_b\right) \qquad \text{(A-14)}$$

Again, the function δ is the unit step-function which is unity for escaping orbits. Note that the Gaussian coefficients are (conveniently) all unity for this quadrature scheme.

In the absence of plasma flow ($M = 0$), or for electrons, one can consider I to be a sum of monoenergetic contributions, which becomes evident by rearranging the sum, namely:

$$S = \frac{4}{M_e}\sum_{K_e=1}^{M_e}\left[T(E') \cdot F(E') + T(E'') \cdot F(E'')\right] \qquad \text{(A-15)}$$

where F(E) is a new quantity defined by:

$$F(E) \equiv \frac{1}{4M_aM_b}\sum_{K_a=1}^{M_a}\sum_{K_b=1}^{M_b}\left[\delta(E, \alpha', \beta') + \delta(E, \alpha'', \beta'')\right] \qquad \text{(A-16)}$$

Note that, since δ is either unity or zero according as the orbit escapes or is absorbed, the sum of Eq. (A-16) is the quotient of two integers, namely, the number of orbits escaping divided by the total number of orbits ($4M_aM_b$), for the given energy E. Thus, F is a fraction between zero and unity, and becomes unity if all orbits escape. Note also that information regarding the energy distribution resides in T. Thus, a non-Maxwellian distribution may be treated by suitably modifying T. In particular, for a monoenergetic distribution we simply set E equal to unity and replace S, Eq. (A-15), by the single term:

$$S_{mono} = \left\{ \begin{array}{c} 1 - \phi \\ \hline \sqrt{1 - \phi} \end{array} \right\} \cdot F(1) \text{ for } \left\{ \begin{array}{l} \text{current density} \\ \\ \text{density} \end{array} \right. \qquad \text{(A-17)}$$

where F(1) is given by Eq. (A-16) with E = 1 and represents the fraction of orbits which escape. The dimensionless potential ϕ is now a multiple of the singular energy of the particles.

The equations derived here are suitable for a computer program and have been incorporated into the program used for the results discussed in Chapter 4.

The method of computation of orbits involves integration of the equations of motion, with the forces given by the components of the gradient of potential. These components are obtained by interpolation between values of potential defined at the points of a grid in space as described in the next appendix. The criterion for "escape" of an orbit (i.e., evaluation of δ) depends on the geometry of the problem and of the grid. The equations of motion are integrated step-by-step until the orbit either passes out of the outer boundary of the grid ("escapes" so that $\delta = 1$) or returns to one of the metal surfaces (is "absorbed" so that $\delta = 0$). The orbit computation time-step is not of physical importance in these time-independent problems where only the shape of the orbit matters. The time-step is kept as large as possible consistent with maintaining the energy loss or gain within desired limits. The method of integrating the equations

of motion, the interpolation method to find the forces, and the control of step size, are discussed in Appendix C.

An important consideration is the accuracy of the quadrature-sum. Naturally, the accuracy is related to the number of terms used, that is, the number of orbits where each term corresponds to a unique orbit. The total number of orbits involved in Eq. (A-11) or Eq. (A-15) is given by $8M_eM_aM_b$. In a test of the energy quadrature alone, and with M = 0, the unperturbed value of density (unity) was computed for values of M_e = 1, 2, 4, 8, 16, and 32. The corresponding numerical errors were -6%, -7%, +1.5%, -0.05%, +0.013%, and +0.003%. This test was independent of geometry (the α and β integrations were numerically exact). Thus, M_e = 4 (8 values of E) is taken to represent sufficient accuracy (within a few percent) for the purposes of computing density for a Maxwellian distribution without drift (or, for electrons).

A device for improving the accuracy of the quadratures at large Mach numbers, without increasing the total number of orbits and therefore computer time, is to suitably weight the integrand in the domains of importance. Thus, one modification is to multiply the term "(1 + c)/(1 - c)" in Eq. (A-5) by M^2 when M exceeds a suitable value, say unity. This emphasizes the contributions of orbits having E in the vicinity of M^2. Another modification is to replace "a" in Eq. (A-6) by the function "$-1 + 2((1 + a)/2)^M$". This emphasizes the contributions of orbits having α in the vicinity of π, that is, directed upstream. When these modifications are used, the quadrature sums must also be multiplied by additional corresponding factors, namely, "M^2" and "$M((1 + a)/2)^{M-1}$", respectively.

APPENDIX B

THE POISSON PROBLEM: POISSON DIFFERENCE EQUATIONS

In the problems treated in this report the electrostatic field is axially symmetric and is defined on a mesh of spatial grid points, such that at any point (including grid points) the potential and electric field are obtained by interpolation.

Assume that the space charge density is known at the grid points. Consider a group of interior grid points, forming a portion of the overall grid as shown in Fig. B-1. In this figure, the vertical and horizontal directions are the z and r directions, respectively, where z and r denote the cylindrical axial and cylindrical radial coordinates, respectively. Three horizontal grid lines, of constant z-values z_{i-1}, z_i, and z_{i+1}, and three vertical grid lines, of constant r-values r_{j-1}, r_j, and r_{j+1}, are shown in the figure. (Note that the index (i) of z increases as z decreases.) The set of grid lines intersect at 9 grid points, or nodes, as shown. Each point may be considered to be associated with a volume of space, and to have a group of four neighboring points which "interact" with it. Thus, consider the central point of the group, labelled C in the figure. Associated with this point is a volume of revolution (a torus) whose cross-section is rectangular and is shown by the rectangular shaded area surrounding Point C. The shaded area is defined by connecting the mid-points of the surrounding mesh rectangles. Let τ denote the volume of the torus, and let the neighboring points (above, below, to the right of, and to the left of C) be labelled N, S, E and W (north, south, east and west, respectively).

Let the Poisson equation be written in dimensionless form as

$$\nabla^2\phi = -\rho \equiv (n_e - n_i)/\lambda_D^2 \qquad \text{(B-1)}$$

where n_e, n_i, λ_D, ϕ and ρ denote the dimensionless electron density, ion density, Debye length, electrostatic potential and space-charge density, respectively; and all lengths are in units of the body radius. Now integrate Eq. (B-1) over the volume τ of the torus associated with point C:

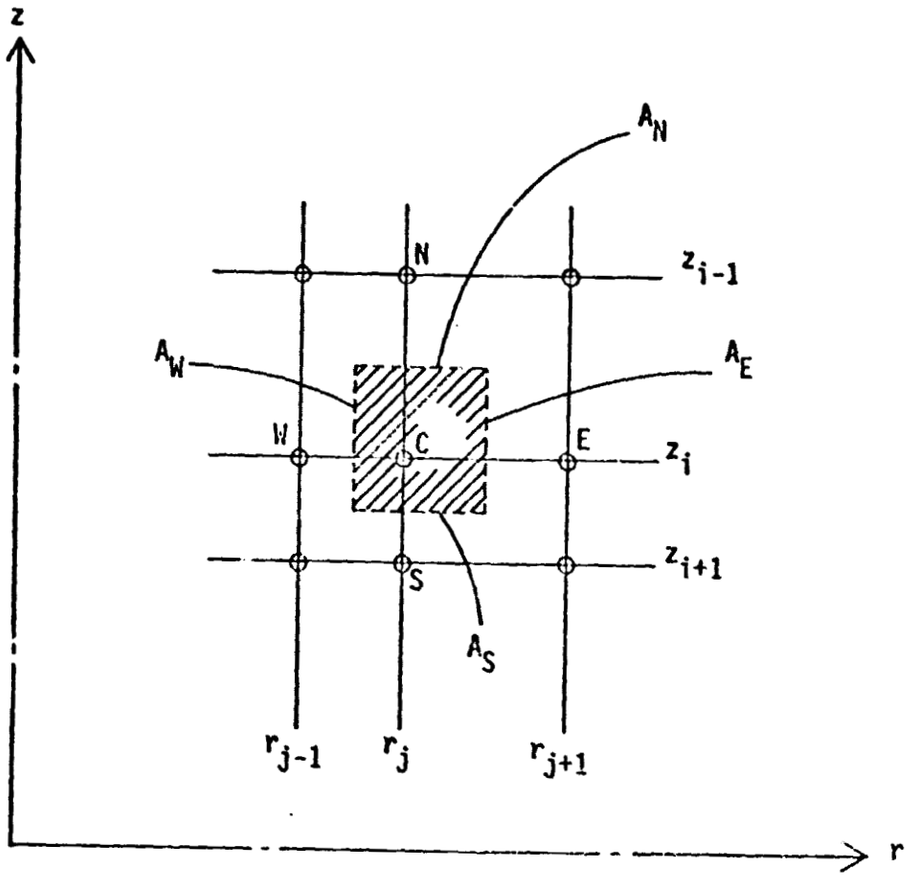


FIG. B-1. GROUP OF INTERIOR GRID POINTS IN r,z GRID

$$\iiint_{\tau} \nabla^2\phi \, d\tau = - \iiint_{\tau} \rho d\tau \cong - \rho_C \tau \tag{B-2}$$

where ρ_C is known at the grid point C. The right-hand side has been approximated as shown since τ is small in principle, and ρ_C is the value of ρ at Point C. By the divergence theorem, the left-hand side becomes

$$\iint_{\Sigma} \frac{\partial\phi}{\partial n} \, d\Sigma \cong \tag{B-3}$$

$$A_N \left(\frac{\partial\phi}{\partial n}\right)_N + A_S \left(\frac{\partial\phi}{\partial n}\right)_S + A_E \left(\frac{\partial\phi}{\partial n}\right)_E + A_W \left(\frac{\partial\phi}{\partial n}\right)_W$$

where Σ denotes the surface of the torus; $\partial\phi/\partial n$ is the component of $\nabla\phi$ in the outward normal direction at the surface; A_N, A_S, A_E, and A_W denote the areas of the north, south, east, and west surfaces, respectively; and the quantities $(\partial\phi/\partial n)_{N,S,E,W}$ denote values of $\partial\phi/\partial n$ taken to be constant on the corresponding surfaces.

$(\partial\phi/\partial n)_{N,S,E,W}$ may be approximated by difference quotients, namely,

$$\left(\frac{\partial\phi}{\partial n}\right)_N \cong \frac{(\phi_N - \phi)}{(z_{i-1} - z_i)} \tag{B-4}$$

$$\left(\frac{\partial\phi}{\partial n}\right)_S \cong \frac{(\phi_S - \phi)}{(z_i - z_{i+1})} \tag{B-5}$$

$$\left(\frac{\partial\phi}{\partial n}\right)_E \cong \frac{(\phi_E - \phi)}{(r_{j+1} - r_j)} \tag{B-6}$$

$$\left(\frac{\partial\phi}{\partial n}\right)_W \cong \frac{(\phi_W - \phi)}{(r_j - r_{j-1})} \tag{B-7}$$

where ϕ denotes the potential at Point C and ϕ_N, ϕ_S, ϕ_E, ϕ_W denote the neighboring potentials. The areas A_N, A_S, A_E, and A_W are given by

$$A_N = \frac{\pi}{4}\left[(r_{j+1} + r_j)^2 - (r_j + r_{j-1})^2\right] \tag{B-8}$$

$$A_S = A_N \tag{B-9}$$

$$A_E = \frac{\pi}{2}(r_{j+1} + r_j)(z_{i-1} - z_{i+1}) \tag{B-10}$$

$$A_W = \frac{\pi}{2}(r_j + r_{j-1})(z_{i-1} - z_{i+1}) \tag{B-11}$$

and the volume τ is given by

$$\tau = \frac{A_N}{2}(z_{i-1} - z_{i+1}) \tag{B-12}$$

Thus, equating Eq. (B-3) with Eq. (B-2), and substituting the foregoing, we obtain the difference equation in the form

$$C_N \phi_N + C_S\phi_S + C_E\phi_E + C_W\phi_W - C\phi = -\rho_C\tau \tag{B-13}$$

where

$$C = C_N + C_S + C_E + C_W \tag{B-14}$$

and

$$C_N = \frac{A_N}{(z_{i-1} - z_i)} \tag{B-15}$$

$$C_S = \frac{A_S}{(z_i - z_{i+1})} \tag{B-16}$$

$$C_E = \frac{A_E}{(r_{j+1} - r_j)} \tag{B-17}$$

$$C_W = \frac{A_W}{(r_j - r_{j-1})} \tag{B-18}$$

This shows how to form the difference equations used for the Poisson problems of this report. Equation (B-14) holds only for an "interior" point of the grid, that is, a point surrounded by neighbors on all four sides.

If Point C is not an interior point, one or more of the terms on the left-hand side of Eq. (B-13) may vanish. To see how this happens, consider an example grid shown in Fig. B-2 where there are 16 grid points, but only 13 of these correspond to unknown potentials to be solved for.* The grid points where the potentials are unknown are numbered and indicated by circles. The three solid circles labelled by the letters a, b, and c denote electrode surface points where the potentials are known. Points No. 1, 2, 4, 5, 9, 10, and 12 are special points all of which have different properties, and are indicated by small crosses within the circles. Among these points, the only interior point is No. 10.

Consider Point No. 10, which has a known potential, namely $\phi_b = \phi_S$, as its southern neighbor. The equation for this point is given by a modification of Eq. (B-13) in that the term $C_S\phi_S$ is now known and is transferred to the right-hand side. For this point, Eq. (B-14) still holds.

Consider Point No. 5, which is on the axis. Its equation is given by the modification of Eq. (B-13) in which C_W vanishes, and the remaining coefficients are evaluated with $r_j = r_{j-1} = 0$. Equation (B-14) still holds, but with C_W not appearing.

Consider Point No. 9, which is both on the axis and has a known neighbor ($\phi_a = \phi_S$). The modification of Eq. (B-13) includes both the modifications for Points No. 5 and 10. Equation (B-14) still holds, but with C_W not appearing.

*This grid illustrates only the space behind the body; at points elsewhere around the body the formulas are similar.

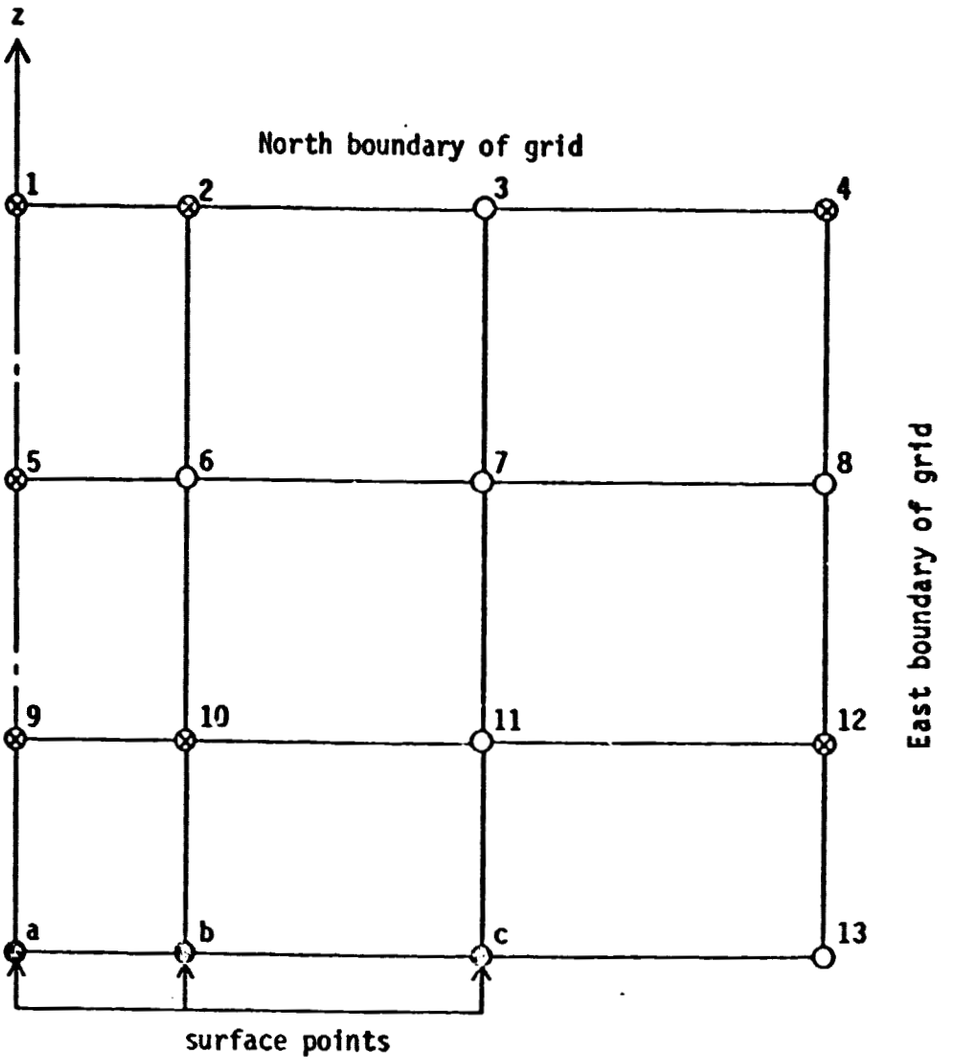


FIG. B-2. SPECIAL GRID POINTS

The modifications for Points No. 1, 2, 4, and 12 depend on the fact that these points are on the outer boundary of the grid, where a "floating" boundary condition is used. Namely, the potentials are assumed to satisfy the linear laws

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial z} = - A' \phi \tag{B-19}$$

and

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial r} = - B' \phi \tag{B-20}$$

on the z-boundary (north, or south), and on the r-boundary (east), respectively. The formulas chosen for A' and B' depend on the assumed potential model. As a result of using Eq. (B-19) or (B-20) (or both for a corner point), corresponding terms do not appear on the left-hand side of Eq. (B-13), and the coefficient C is suitably modified so that Eq. (B-14) no longer holds. That is, for Points 1, 2, 4, and 12, C is given, respectively, by

$$C = A' A_N + C_S + C_E \text{ (Point 1)} \tag{B-21}$$

$$C = A' A_N + C_S + C_E + C_W \text{ (Point 2)} \tag{B-22}$$

$$C = A' A'_N + C_S + B' A'_E + C_W \text{ (Point 4)} \tag{B-23}$$

$$C = C_N + C_S + B' A'_E + C_W \text{ (Point 12)} \tag{B-24}$$

where it is understood that the coefficients in the above equations depend on the location of the point in question. (A'_N and A'_E denote modifications of A_N and A_E; see examples given in Parker (1968).)

Once the coefficients of all of the equations (corresponding to the grid points where the potentials are unknown) are computed, the system of linear equations may be solved by iteration. Point-successive over-relaxation has been found to work well (Parker, 1968).

The relaxation procedure may be illustrated by re-writing Eq. (B-13) in the form

$$C\phi^{n+1} = \left(C_N\phi_N + C_S\phi_S + C_E\phi_E + C_W\phi_W \right)^{n,\, n+1} + \rho_C\tau \qquad \text{(B-25)}$$

The superscripts n and n+1 denote the n-th and (n+1)-th iterates, respectively. At the beginning of each "sweep" through the equations, all ϕ^n are considered known. Then new values of ϕ are obtained by putting on the right-hand side of Eq. (B-25) the most recently updated values of ϕ at the neighboring grid points. The superscripts "n, n+1" on the right-hand side of Eq. (B-25) imply that one or more of the quantities ϕ_N, ϕ_S, ϕ_E, or ϕ_W might have been already updated by appearing on the left-hand side of a previous equation during the sweep. Thus far, the iteration scheme indicated by Eq. (B-25) is known as the "Gauss-Seidel" iteration. Convergence usually requires many iterations for problems involving many grid points.

A more efficient procedure which will reduce the number of iterations is to modify the ϕ^{n+1} obtained from Eq. (B-25) before going on to the next equation in the sweep. The modification is given by the "relaxation" equation

$$(\phi^{n+1})_{\text{modified}} = \omega(\phi^{n+1})_{\substack{\text{Gauss-}\\ \text{Seidel}}} + (1-\omega)\phi^n \qquad \text{(B-26)}$$

where ω is a relaxation parameter and the first term on the right-hand side of Eq. (B-26) involves as a factor the ϕ^{n+1} resulting from Eq. (B-25). The parameter ω may be less than unity ("under-relaxation") or greater than unity ("over-relaxation"). In practice it is found that the number of iterations is minimized dramatically when ω is close to but less than 2. For example, simply increasing the value of ω from 1.0 (Gauss-Seidel iteration)

to 1.9 (over-relaxation) typically reduces the numbers of iterations, in the problems of this report, from the order of thousands to the order of hundreds (for convergence to within a fractional change of one part in 10^5 in all potentials).

A simple modification of the foregoing procedure "forces" the potential to have arbitrary desired values at any of the grid points. This means that one may, for example, set the potential to zero at the outer grid points if for some reason this is felt to give a better representation of "infinity" than the floating one. Or it allows one to arbitrarily specify the shape of the body surface, which may be of value for three-dimensional problems. The modification consists of re-defining (overriding) the appropriate values of potential as soon as they are updated on the left-hand side of Eq. (B-25). This procedure is valid since the convergence of the iteration is not significantly affected.

Another modification which is of value for large bodies is afforded when the potential distribution is such that n_e is approximable by the Boltzmann factor $\exp(\phi)$. In this case, one may replace Eq. (B-25) by the equation

$$C\phi^{n+1} + (\tau/\lambda_D^2)\exp(\phi^{n+1}) = (C_N\phi_N + C_S\phi_S + C_E\phi_E + C_W\phi_W)^{n,n+1} + n_i\tau/\lambda_D^2 \qquad (B\text{-}27)$$

and then solve this nonlinear equation for ϕ^{n+1}, with n_i considered to be held fixed.

APPENDIX C

COMPUTER PROGRAM

This program implements the calculational procedures described in Chapters 2 and 3, and in Appendices A and B. The main controlling program name is DWAKE (for "Disk Wake") or TDWAKE (for "Thick-Disk Wake," referring to modifications for a thick disk or short cylinder). There are two principal subprograms (FIELD and DENSTY), and eight subroutines serving these programs. The control hierarchy is listed as follows.

DWAKE or TDWAKE (main)

FIELD (Poisson problem)	DENSTY (Vlasov problem)
LEFT	INTERP
MIDDLE	TRACK
RIGHT	
PRINT	
SEIDEL	
ROOT	

The program operation is as follows.

FIELD solves the Poisson problem for the potentials when (optionally) either the ion or charge densities are given. DENSTY solves the Vlasov problem for the number and current densities when the potentials are given. The subroutines LEFT, MIDDLE, RIGHT, PRINT, and SEIDEL are called by FIELD, while INTERP and TRACK are called by DENSTY. FIELD and its subroutines constitute one overlay, while DENSTY and its subroutines constitute the other overlay. SEIDEL in turn calls ROOT if the ion-density option is selected.

DWAKE reads the input data (described below) consisting of the positions of the grid lines, the values of the potential at points on the metal surfaces, the Debye number, the Poisson-Vlasov major-iteration mixing parameter α, the maximum number of major iterations, the array of input ion or charge densities, and various parameters affecting the options, orbit calculations, quadrature orders, single space-point coordinates, and single-orbit initial conditions.

A Poisson-Vlasov iteration cycling loop is set up in which DWAKE calls FIELD and DENSTY, in each iteration cycle. DENSTY is called twice, once to compute the ion or charge densities and then to compute the current density. (In the appended listing the current density is computed only at the central wake point.) The cycling is continued for the desired maximum numbe of iterations, with the potentials or densities (normally, the potentials) being punched out on cards at every cycle for possible use in continuation.

C.1 Operation of FIELD and Its Subroutines

FIELD sets up the coefficients of the potential-unknowns in the linear equations resulting from the differencing of the Poisson equation as described in Appendix B. Subroutine LEFT is called if the point in question is on the axis; subroutine RIGHT is called if the point is on the right-hand grid-boundary line; subroutine MIDDLE deals with an interior point. For each point corresponding to an unknown value of potential, FIELD obtains the values of the variables C, CN, CS, CE, and CW corresponding to the coefficients used in the equations of Appendix B. The values of these coefficients, as well as the volume of the volume element (V) associated with the point in question, are printed out and saved (in the order in which they would appear as matrix elements) by subroutine PRINT. The right-hand sides of the Poisson difference equations are also set up using the input ion or space-charge densities. When the setting-up has been completed, FIELD calls SEIDEL to obtain the solution, which is accomplished by point-successive over-relaxation. If the option is selected in which the nonlinear equation at the end of Appendix B is to be solved (appropriate for large bodies and where the Boltzmann factor can be used for the electrons), then SEIDEL calls ROOT within its over-relaxation loop. In ROOT, solution of each nonlinear equation is achieved by Newton's method for root evaluation. If the thick-disk option is selected, the body potential overrides the calculated potentials at grid points covered by the body. After SEIDEL has obtained the potentials in the form of a one-dimensional solution-vector X, the main program converts these to the two-dimensional potential arrays PHIN and PHIS used by DENSTY for the Vlasov problem.

C.2 Operation of DENSTY and Its Subroutines

DENSTY sets up an outermost loop in which one or more space points (or all grid points) are to be processed. First, the positive ions are treated, and then the electrons are treated by changing the sign of the potentials. There is a number density option and a current density option. The orbit quadrature loops are then set up, in which the energy and two velocity-angles are the variables of integration (summation). One option is that of a single orbit; another option is that of a single energy (mon-energetic) in which only the angle sums are carried out. The quadratures are slightly different for the current and number density options. Each orbit is traced step-by-step backward in time until it terminates either on the metal surfaces or at "infinity" (outer boundary of the grid). At each step of the orbit, DENSTY calls INTERP and TRACK. It calls INTERP to find the potential and the components of the potential gradient (by interpolation within the grid) for use in the equations in motion. It calls TRACK to update the position and velocity components through the Newton equations of motion. In INTERP z and r are located by means of a table search, either in the North region or in the South region. In order to save time, after the first step in which the whole table is searched, the search is confined only to nearest-neighbor table entries. After the "box" containing z and r is identified, the potentials at the 4 corners of the box are used for a double-linear interpolation.

TRACK, where the Newton equations of motion are used, monitors the "non-conservation" of energy, that is, the relative deviation between the assigned total energy E and the sum of the kinetic plus potential energies, where the latter two are subject to numerical errors as the orbit is propagated. The degree of energy non-conservation is kept under control by means of a time-step control governed by an input variable (STEP). Since the time-step control logic is probably not self-evident, this logic is explained as follows.

During each time-step, the acceleration a is assumed to be constant. Therefore, if a is approximately constant within each grid zone (i.e., between two adjacent grid points in r and in z), the error (and, therefore,

energy loss or gain) would be that incurred upon crossing a boundary between two grid zones during a time step. Let a_1 and a_2 be the constant accelerations in the first and second grid zones, respectively. Then the absolute value of the energy loss or gain, $|\Delta E|$, would be given by

$$|\Delta E| = |a_2 - a_1| \, S_2$$

where S_2 is the path length in the second zone. Now we know a_1 and not a_2. However, we estimate $|a_2 - a_1|$ to be of the same order as $|a_1|$. Moreover, we estimate S_2 to be the larger of the two quantities

$$|v_1|\Delta t \qquad \text{or} \qquad |a_1|(\Delta t)^2$$

where Δt is the time-step interval. Hence, $|\Delta E|$ is estimated as the larger of the two quantities

$$|a_1 v_1|\Delta t \qquad \text{or} \qquad a_1^2(\Delta t)^2$$

Thus, if $|\Delta E| \equiv fE$ is considered as known, the appropriate Δt may be estimated as the smaller of the two quantities

$$\frac{fE}{|a_1 v_1|} \qquad \text{or} \qquad \frac{\sqrt{fE}}{|a_1|}$$

In the program, the fraction f is identified with the input variable STEP. There is an additional control on Δt, to avoid taking too many steps, namely, a minimum value $\Delta t = \Delta S/|v_1|$ where $\Delta S = .01$ somewhat arbitrarily, representing of the order of 100 steps per unit length. For no electric field, Δt is taken to be the minimum value, with STEP replacing ΔS.

C.3 FORTRAN Input Variables

The following variables are read by DWAKE as input:

Card No. 1

DATE = any identification (alphanumeric, 80 columns).

Card No. 2 (integers only, 5 columns each)

NCOLSN = number of r-values on North (wake) face of disk.
NCOLSE = number of r-values East of disk (between disk and grid boundary).
NCOLSS = number of r-values on South (front) face of disk.
NROWSN = number of z-values, North region (between disk wake surface and grid boundary).
NROWSS = number of z-values, South region (between disk front surface and grid boundary).

Card No. 3 (8 fields of 10 columns each, per card)

RHON1 = r-values on North face of disk.

Card No. 4 (8 fields of 10 columns each, per card)

RHOE = r-values East of disk.

Card No. 5 (8 fields of 10 columns each, per card)

RHOS1 = r-values on South face of disk.

Card No. 6 (8 fields of 10 columns each, per card)

ZN = z-values, North region.

Card No. 7 (8 fields of 10 columns each, per card)

ZS = z-values, South region.

Card No. 8 (8 fields of 10 columns each, per card)

PHI = potential values at the grid points on North and South faces of disk.

Card No. 9 (7 fields of 10 columns each)

DEBYE = Debye number.

ALPH = iteration parameter α.

RBOUND = r-value of East boundary of grid. (If this is greater than unity, r-values East are proportionally scaled between unity and RBOUND.)

ZNBOUND = z-value of North boundary of grid. (If this is greater than zero, z-values North are proportionally scaled between zero and ZNBOUND.)

ZSBOUND = z-value of South boundary of grid. (If this is less than zero, z-values South are proportionally scaled between zero and ZSBOUND.)

RWAKE = radius in wake such that additional iterations are applied to grid points lying within RWAKE, if option (ITALL > 1 on Card No. 10) is chosen.

ZFRONT = negative z-value of front face of thick disk. If this is not zero, the disk is treated as having thickness equal to -ZFRONT. (Program TDWAKE.)

Card No. 10 (integers only, 5 columns each)

ITS = number of iteration cycles allowed (beyond the initial one). If this is zero, only one iteration is performed.

IT = initial iteration cycle number (zero for new problem; greater than zero if continuing from previous problem).

NEWPHI = zero for charge-density option; greater than zero for ion-density option.

MAME = nonzero only if additional accuracy (more trajectories) desired for points in wake near axis.

ITALL = nonzero if it is desired to apply more iterations to points in wake than to other points; indicates that all points are to be computed once every "ITALL-th" iteration. If this is zero, all points are computed every iteration.

Card No. 11 (if IT = 0) Normally applies to number densities.

NPRINT = printout option regarding orbit details (NPRINT = 0 for minimum normal printout). (Column 1.)

MD=0,>0 indicates one, or more than one, space point to be processed (normally greater than zero on Card No. 11). (Column 2.)

MC=0,>0 indicates whether charge- or current-density is to be calculated (normally equal to zero on Card No. 11). (Column 3.)

MA = number of quadrature intervals M_a for polar angle α. MA=0 indicates single orbit. (Columns 4 and 5.)

MB = number of quadrature intervals M_b for azimuthal angle β. (Columns 6-10.)

ME = number of quadrature intervals M_e for energy E. ME=0 indicates single energy. (Columns 11-15.)

STEP = orbit step-size control, for control of accuracy. (Columns 16-20.)

RSAVE = initial value of r for single orbit or for single space point. (Columns 21-30.)

ZSAVE = initial value of z for single orbit or for single space point. (Columns 31-40.)

ALPHA = initial value of polar angle α for single orbit. (Columns 41-50.)

BETA = initial value of azimuthal angle β for single orbit. (Columns 51-60.)

EE = value of energy E for single orbit or for monoenergetic velocity distribution. (Columns 61-70.)

XMSAVE = drift Mach number for Maxwellian with drift. (Columns 71-80.)

Card No. 12 (if IT=0) Normally applies to current densities.

Same as above, except that MD=0 and MC>0.

Additional Data Cards for Continuation

If IT (iteration cycle number) is greater than zero on Card No. 10, that is, if the calculation is a continuation, then an array (normally of potentials) punched from the previous calculation is read in, after Card No. 10 but before Cards No. 11 and 12.

C.4 FORTRAN Output Variables (Normally Generated by Program)

Output generated by DWAKE (main program)

Geometric and other input data, potential arrays, and iteration cycle number.

Output generated by PRINT (called by FIELD)

In the following, CN, CS, CE, CW, and C are the non-vanishing coefficients in the Poisson difference-equation matrix: (initial cycle only)

INDX = index of unknown potential (equation number)
INDXN = index of North-neighbor potential
CN = coefficient of North neighbor in equation number given by INDX
INDXS = index of South-neighbor potential
CS = coefficient of South neighbor in equation number given by INDX
INDXE = index of East-neighbor potential
CE = coefficient of East neighbor in equation number given by INDX
INDXW = index of West-neighbor potential
CW = coefficient of West neighbor in equation number given by INDX
C = coefficient of central unknown in equation associated with INDX
V = volume of space associated with central unknown in equation associated with INDX

Output generated by SEIDEL (called by FIELD)

X = solution of Poisson problem
= one-dimensional array of potentials

Output generated by DENSTY

PHIN = two-dimensional array of potentials, North region
PHIS = two-dimensional array of potentials, South region

NUMBER = number of trajectories (orbits) for each energy. (Off the axis, this is given by $4 M_a M_b$; on the axis, this is given by $2M_a$ since only one value of β is necessary by symmetry.)

If the number-density option is chosen (MC=0), the following output is given on a single line:

N, RSAVE, ZSAVE, PHISAV = point number, r-coordinate, z-coordinate, and potential, of point where density is calculated.

DENSA, DENST, CD = ion number-density, electron number-density, and charge (net) density.

If computations are done at a group of space points (MD>0, usually for all the grid points), the output includes the ion or charge density summary tabulation:

RSV, ZSV, CDSV = r-coordinate of point, z-coordinate of point, and ion or charge density at that point.

To monitor which trajectory takes the most steps, the following output is given on a single line:

MOSTPS, NSAVE, KES, JES, KBS, JBS, KAS, JAS = the largest number of steps taken by a trajectory, the index of the associated space point, and the velocity-coordinate indices of the trajectory (in the quadrature sum) taking the largest number of steps.

In addition, the ion densities are printed out as two-dimensional arrays, DNORTH and DSOUTH.

If the current-density option is chosen (MC>0), the following output is given:

For every value of energy E calculated by the quadrature formula, a line is printed, consisting of

NOESC = number of trajectories escaping

NUMBER = total number of trajectories at the point of interest

FRACT = the fraction escaping (NOESC/NUMBER)

E = the energy E

DENS = the partial density in the summation associated with energy E

The following output is also given, as a single line:

RSAVE, ZSAVE, PHISAV = r-coordinate, z-coordinate, and potential, of point where current density is calculated.

PARTCL, DENST = "ion" or "electron," and value of current density.

The listing for the TDWAKE program is given in the following pages.

```
      OVERLAY(DISKUS,0,0)
**********  MODIFICATION FOR FINITE DISK THICKNESS
      PROGRAM TDWAKE(INPUT,OUTPUT,TAPE60=INPUT,TAPE61=OUTPUT,PUNCH)
C
C     SATELLITE WAKE PROBLEM  (UNSYMMETRIC DISK = DISKUS)
C     MIXING OF POTENTIALS RATHER THAN DENSITIES
C     OPTIONAL ITERATION BY SWEEPING GROUPS OF POINTS (UPSTREAM TO DOWNSTREAM
C     MODIFICATION TO TREAT PILLBOX OR THICK DISK (-ZFRONT=THICKNESS)
C
      COMMON JJN,IIN,JJS,IIS,NTOT,RNT(500),ZNT(500),RST(500),ZST(500),
     1 RZ(500,2),PHIN(20,20),PHIS(20,20),CD(500),IFIRST,M
      COMMON/FLD/NCOLSN,NCOLSE,NCOLSS,NROWSN,X(500),NROWSS,DEBYE,DEBYE2,
     1 RHON1(500),RHOE(500),RHOS1(500),ZN(500),   ZS(500),PHI(500),NGAP,NDISK
      COMMON/DEN/NPRINT,MD,MC,MA,MB,ME,STEP,RSAVE,ZSAVE,ALPHA,BETA,EE,
     1 XMSAVE ,RADIUS
      COMMON/DEN1/IT,NAME,NEWPHI,ISAVE,NGR,NGROUP(500),DSAVE(500)
**********  MODIFICATION FOR FINITE DISK THICKNESS
     1 , ZFRONT
      DIMENSION DATE(20), X1(500)
C
C  FOR DENSITY AND CURRENT CALCULATIONS
C  INTEGER INPUTS ARE NPRINT,MD,MC,MA,MB,ME. NPRINT=0,1,2,3 FOR PRINTING
C  NO INTERMEDIATE DATA, ESCAPING INDICES, FIRST AND LAST STEPS, AND ALL
C  STEPS.  MD=0,1 FOR ONE SPACE POINT (RSAVE,ZSAVE) OR SEVERAL SPACE
C  POINTS (READ IN R,Z VALUES).  MC=0,1 FOR CHARGE DENSITY OR CURRENT DEN-
C  SITY. MA=0 MEANS ONE TRAJECTORY (READ RSAVE,ZSAVE,ALPHA,BETA,EE).  ME=0
C  MEANS ONE ENERGY (READ EE).  OTHERWISE, MA,MB,ME, DENOTE THE NUMBER O
C  ALPHA-INTERVALS, BETA-INTERVALS, AND ENERGY-INTERVALS.
C
      L=60
      M=61
      IPUNCH=0
      IFUNCH=1
      ISAVE=0
      DEBYE2=0.
    1 READ(L,9998) DATE
 9998 FORMAT(20A4)
      IF(EOF(L)) 99,15
   15 WRITE(M,9999) DATE
 9999 FORMAT(31H1UNSYMMETRIC DISK FIELD PROBLEM ,20A4)
      READ(L,111)  NCOLSN,NCOLSE,NCOLSS,NROWSN,NROWSS
      NROWSE=1
      JJN=NCOLSN+NCOLSE
      IIN=NROWSN+1
      JJS=NCOLSS+NCOLSE
      IIS=NROWSS+1
      NDISK=NCOLSN+NCOLSS
      NGAP=NDISK+1
      NTOT=JJN*NROWSN+NCOLSE+JJS*NROWSS
      READ(L,222) (RHON1(J),J=1,NCOLSN)
      READ(L,222) (RHOE(J),J=1,NCOLSE)
      READ(L,222) (RHOS1(J),J=1,NCOLSS)
      READ(L,222) (ZN(I),I=1,NROWSN)
      READ(L,222) (ZS(I),I=1,NROWSS)
      READ(L,222) (PHI(J),J=1,NDISK)
      PHI(NGAP)=.5*(PHI(NCOLSN)+PHI(NCOLSN+1))
      RADIUS=RHON1(NCOLSN)
```

```
********** MODIFICATION FOR FINITE DISK THICKNESS
      READ(L,222) DEBYE,ALPH,RBOUND,ZNBOUND,ZSBOUND,RWAKE,ZFRONT
      WRITE(M,220)DEBYE,ALPH,RBOUND,ZNBOUND,ZSBOUND,RWAKE,ZFRONT
      READ(L,111)  ITS,IT,NEWPHI,MAME,ITALL
      WRITE(M,110) ITS,IT,NEWPHI,MAME,ITALL
      ITMAX=ITS+IT
      DO 17 N=1,NTOT
      X(N)=X1(N)=0.
      OSAVE(N)=0.
      CD(N)=0.
17    CONTINUE
      NTOTAL=NTOT
      IF(ISAVE.GT.0) READ(L,52)NTOT,NTOT,(OSAVE(N),N=1,NTOT)
      IF(IT.EQ.0) GO TO 21
      IF(DEBYE.GT.0.) READ(L,52)IT,NTOTAL,( X(N),N=1,NTOTAL)
      IF(DEBYE.GT.0..AND.NTOTAL.NE.NTOT) GO TO 18
      GO TO 21
18    WRITE(M,668)
668   FORMAT(////1X,41HTROUBLE - INCORRECT NUMBER OF POINTS READ)
      GO TO 99
C
21    IF(NEWPHI.EQ.0.AND.DEBYE.GT.0.)
     1 WRITE(M,230) DEBYE,ALPH,ITMAX,IT,(N, X(N),N=1,NTOT)
      IF(NEWPHI.GT.0.AND.DEBYE.GT.0.)
     1 WRITE(M,2301)DEBYE,ALPH,ITMAX,IT,(N, X(N),N=1,NTOT)
C
C RESCALE GRID LINE POSITIONS TO FIT WITHIN INPUT BOUNDS.
C
      RATIO=(RBOUND-RADIUS)/(RHOE(NCOLSE)-RADIUS)
      DO 23 J=1,NCOLSE
      IF(RATIO.GT.0.)  RHOE(J)=RADIUS + RATIO*(RHOE(J)-RADIUS)
23    CONTINUE
      RATIO=ZNBOUND/ZN(1)
      DO 24 I=1,NROWSN
      IF(RATIO.GT.0.)   ZN(I)=RATIO*ZN(I)
24    CONTINUE
      RATIO=ZSBOUND/ZS(NROWSS)
      DO 25 I=1,NROWSS
      IF(RATIO.GT.0.) ZS(I)=RATIO*ZS(I)
25    CONTINUE
C
      WRITE(M,112) NCOLSN,NCOLSE,NCOLSS,NROWSN,NROWSS
      WRITE(M,223) (J,RHON1(J),J=1,NCOLSN)
      WRITE(M,224) (J,RHOE(J),J=1,NCOLSE)
      WRITE(M,225) (J,RHOS1(J),J=1,NCOLSS)
      WRITE(M,226) (I,ZN(I),I=1,NROWSN)
      WRITE(M,228) (I,ZS(I),I=1,NROWSS)
      WRITE(M,229) (J,PHI(J),J=1,NGAP)
110   FORMAT(1X,30HNO. OF ITS, IT, NEWPHI, MAME =, 4I5,5X,
     1 16HALL POINTS EVERY , I3,24H-TH ITERATION AFTER IT=2)
111   FORMAT(16I5)
112   FORMAT(//1X,I3,25H COLUMNS (R-VALUES) NORTH/
     1         1X,I3,25H COLUMNS (R-VALUES) EAST /
     2         1X,I3,25H COLUMNS (R-VALUES) SOUTH/
     3         1X,I3,22H ROWS (Z-VALUES) NORTH/
     4         1X,I3,22H ROWS (Z-VALUES) SOUTH)
220   FORMAT(1X,6HDEBYE=,F10.5,5X,5HALPH=,F10.5,
```

```
     1 10X,7HRBOUND , F7.2,5X,8HZNBOUND=, F7.2,5X,8HZSBOUND=, F7.2,
**********  MODIFICATION FOR FINITE DISK THICHNESS
     2 5X, 6HRWAKE=, F7.2/1X,7HZFRONT=, F7.2)
  222 FORMAT(8F10.5)
  223 FORMAT(//15H R-VALUES NORTH/(I3,1PE15.4))
  224 FORMAT(//14H R-VALUES EAST/(I3,1PE15.4))
  225 FORMAT(//15H R-VALUES SOUTH/(I3,1PE15.4))
  226 FORMAT(//15H Z-VALUES NORTH/(I3,1PE15.4))
  228 FORMAT(//15H Z-VALUES SOUTH/(I3,1PE15.4))
  229 FORMAT(//26H POTENTIALS ON DISK SURFACES/(I3,1PE15.4))
 230  FORMAT(/26HC  CONVENTIONAL ITERATION,,
     1 30H  POTENTIALS WITH DEBYE NUMBER, F10.5,5X,6HALPH =,
     2 F10.5,5X,6HITMAX=,I4,5X,3HIT=,I4/(1X,I3,1PE15.4))
 2301  FORMAT(/26HCEXP-IN-POISSON ITERATION,,
     1 30H  POTENTIALS WITH DEBYE NUMBER, F10.5,5X,6HALPH =,
     2 F10.5,5X,6HITMAX=,I4,5X,3HIT=,I4/(1X,I3,1PE15.4))
 5050 FORMAT(1H0,I4,16H ORDER POTENTIAL)
 5060 FORMAT(1H0,I4,21H ORDER DENSITIES     )
 5070 FORMAT(1H0,I4,22H ORDER CURRENT DENSITY)
   52 FORMAT(2I5,1P7E10.3/(  8E10.3))
C
      DO 2 J=1,NCOLSN
    2 RNT(J)=RHON1(J)
      DO 3 J=1,NCOLSE
      JPN=J+NCOLSN
    3 RNT(JPN)=RHOE(J)
      DO 4 I=1,NROWSN
    4 ZNT(I)=ZN(I)
      ZNT(IIN)=0.
      DO 5 J=1,NCOLSS
    5 RST(J)=RHOS1(J)
      DO 6 J=1,NCOLSE
      JPS=J+NCOLSS
    6 RST(JPS)=RHOE(J)
      ZST(1)=0.
      DO 7 I=1,NROWSS
      IPS=I+1
    7 ZST(IPS)=ZS(I)
      WRITE (M,231) (J,RNT(J),J=1,JJN)
      WRITE (M,232) (J,RST(J),J=1,JJS)
      WRITE (M,233) (I,ZNT(I),I=1,IIN)
      WRITE (M,234) (I,ZST(I),I=1,IIS)
  231 FORMAT (//1X,27HAUGMENTED   R-VALUES, NORTH/(I3,1PE15.4))
  232 FORMAT (//1X,27HAUGMENTED   R-VALUES, SOUTH/(I3,1PE15.4))
  233 FORMAT (//1X,27HAUGMENTED   Z-VALUES, NORTH/(I3,1PE15.4))
  234 FORMAT (//1X,27HAUGMENTED   Z-VALUES, SOUTH/(I3,1PE15.4))
C     OUTPUT    RHO  AND  Z   ARRAYS
      DO 71 I=1,NROWSN
      DO 71 J=1,JJN
      JRZ=(I-1)*JJN+J
      RZ(JRZ,1)=RNT(J)
   71 RZ(JRZ,2)=ZNT(I)
      DO 72 J=1,NCOLSE
      JRZ=(NROWSN*JJN)+J
      RZ(JRZ,1)=RNT(NCOLSN+J)
   72 RZ(JRZ,2)=ZNT(IIN)
      NE=NROWSN*JJN+NCOLSE
```

```
      DO 73 I=1,NROWSS
      DO 73 J=1,JJS
      JRZ=NE+(I-1)*JJS+J
      RZ(JRZ,1)=RST(J)
   73 RZ(JRZ,2)=ZS(I)
C
      NFPP=(NTOT/300) +1
      DO 85 IP=1,NFPP
      WRITE (N,80)
   80 FORMAT (1H1,6X,1HI,4X,6H  R(I),4X,4HZ(I)  )
      DO 85 I=1,60
      K1=I+300*(IP-1)
      K2=K1+60
      K3=K2+60
      K4=K3+60
      K5=K4+60
      IF(K5.LE.NTOT)WRITE(N,8)K1,RZ(K1,1),RZ(K1,2),K2,RZ(K2,1),RZ(K2,2),
     1 K3,RZ(K3,1),RZ(K3,2),K4,RZ(K4,1),RZ(K4,2),K5,RZ(K5,1),RZ(K5,2)
      IF (K5.LE.NTOT) GO TO 85
      IF(K4.LE.NTOT)WRITE(N,8)K1,RZ(K1,1),RZ(K1,2),K2,RZ(K2,1),RZ(K2,2),
     1 K3,RZ(K3,1),RZ(K3,2),K4,RZ(K4,1),RZ(K4,2)
      IF (K4.LE.NTOT) GO TO 85
      IF(K3.LE.NTOT)WRITE(N,8)K1,RZ(K1,1),RZ(K1,2),K2,RZ(K2,1),RZ(K2,2),
     1 K3,RZ(K3,1),RZ(K3,2)
      IF (K3.LE.NTOT) GO TO 85
      IF(K2.LE.NTOT)WRITE(N,8)K1,RZ(K1,1),RZ(K1,2),K2,RZ(K2,1),RZ(K2,2)
      IF (K2.LE.NTOT) GO TO 85
      IF(K1.LE.NTOT)WRITE(N,8)K1,RZ(K1,1),RZ(K1,2)
   85 CONTINUE
    8 FORMAT (5(I8,F10.3,F8.3))
      RZ(NTOT+1,1)=0.
      RZ(NTOT+1,2)=0.
      NGRPS=1
      IF(ITALL.LE.1) ITALL=1
      IF(ITALL.LE.1) GO TO 1201
C
C DEFINE GROUPS 1, 2, ETC., IN THE WAKE, IN ORDER OF AXIAL DISTANCE FROM DI
      ZGROUP=ZN(NROWSN)
      NGR=1
      DO 12 N=1,NTOT
      NREV=NTOT-N+1
      NGROUP(NREV)=0
      IF(RZ(NREV,2).LE.0..OR.NREV.LE.JJN) GO TO 12
      IF(RZ(NREV,2).NE.ZGROUP) GO TO 13
      IF(RZ(NREV,1).GT.RWAKE) GO TO 12
      NGROUP(NREV)=NGR
      GO TO 12
13    ZGROUP=RZ(NREV,2)
***** TEMPORARY JUMP TO FORCE NGR=1 FOR ALL WAKE PTS.
      JUMP=1
      IF(JUMP.EQ.1) GO TO 12
      NGR=NGR+1
12    CONTINUE
      NGRPS=NGR
C
1201  CONTINUE
      READ(L,666)NPRINT1,MD1,MC1,MA1,MB1,ME1,STEP1,RSAVE1,ZSAVE1,
```

```
     1 ALPHA1,BETA1,EE1,XMSAVE1
   666 FORMAT(3I1,I2,2I5,F5.3,6F10.5)
       READ(L,666)NPRINT2,MD2,MC2,MA2,MB2,ME2,STEP2,RSAVE2,ZSAVE2,
     1 ALPHA2,BETA2,EE2,XMSAVE2
C
       IFIRST=0
C
C DO POTENTIALS
C
 10    DO 30 NG=1,NGRPS
       NGR=NG
       IF(IFIRST.EQ.0.OR.IT.LE.2.OR.ITALL.LE.1.OR.MOD(IT,ITALL).EQ.0)
      1  NGR=0
       CALL OVERLAY(6LDISKUS,1,0,0)
       IF(NGR.LE.1) WRITE(M,5050) IT
       NGO=1
 11    IF(NGO.EQ.1) GO TO 45
       DO 20 N=1,NTOTAL
       IF(IFIRST.EQ.0)  X1(N)= X(N)
        X1(N)=ALPH* X(N)+(1.-ALPH)* X1(N)
   20   X(N)= X1(N)
 45    DO 50 I=1,NROWSN
       DO 50 J=1,JJN
       INDX=J+(I-1)*JJN
   50  PHIN(I,J)=X(INDX)
       DO 51 J=1,NCOLSN
       PHIN(IIN,J)=PHI(J)
       IF(IT.EQ.0) PHIN(IIN,J)=0.
 51    CONTINUE
       JPLUSN=NCOLSN+1
       DO 53 J=JPLUSN,JJN
       INDX=INDX+1
   53  PHIN(IIN,J)=X(INDX)
       INDX=INDX-NCOLSE
       JPLUSS=NCOLSS+1
       DO 54 J=1,NCOLSS
       NSUBVR=NDISK-(J-1)
       PHIS(1,J)=PHI(NSUBVR)
       IF(IT.EQ.0) PHIS(1,J)=0.
 54    CONTINUE
       DO 55 J=JPLUSS,JJS
       INDX=INDX+1
   55  PHIS(1,J)=X(INDX)
       DO 56 I=2,IIS
       DO 56 J=1,JJS
       INDX=INDX+1
       PHIS(I,J)=X(INDX)
   56  CONTINUE
       IF(NGO.EQ.2) GO TO 500
       NGO=2
       WRITE(M,120) IT,NGR
 120   FORMAT(///,1X,23HPOTENTIAL ARRAY - NORTH,5X,4HIT =,I3,3X,5HNGR =,
      1 I3)
       WRITE(M,2004) (RNT(J),J=1,JJN)
       DO 91 I=1,IIN
       WRITE (M,123)  I,ZNT(I),(PHIN(I,J),J=1,JJN)
   91  CONTINUE
```

```
      WRITE (M,122)
  122 FORMAT (///,1X,35H POTENTIAL ARRAY - SOUTH.          //)
      WRITE(M,2004) (RST(J),J=1,JJS)
 2004 FORMAT(/1X,2HR=,16F8.4//(/3X,16F8.4))
      DO 93 I=1,IIS
      WRITE (M,123)  I,ZST(I),(PHIS(I,J),J=1,JJS)
   93 CONTINUE
 123  FORMAT(/5H LINE,I4,5X,2HZ=,F8.4//(/7F16.8))
C
      GO TO 11
 500  CONTINUE
      IF(IPUNCH.GT.0)              PUNCH 52, IT,NTOTAL,(X(N),N=1,NTOTAL)
      IF(IT.GT.ITMAX) GO TO 1
C
C FIRST DO DENSITIES
C
      NPRINT=NPRINT1
      MD=MD1
      MC=MC1
      MA=MA1
      MB=MB1
      ME=ME1
      STEP=STEP1
C STOP IN DENSTY IF STEP LE ZERO.
      RSAVE=RSAVE1
      ZSAVE=ZSAVE1
      ALPHA=ALPHA1
      BETA=BETA1
      EE=EE1
      XMSAVE=XMSAVE1
      CALL OVERLAY(6LDISKUS,2,0,0)
      IF(IFIRST.EQ.0.OR.NGR.EQ.0) GO TO 31
 30   CONTINUE
 31   CONTINUE
      IFIRST =IFIRST+1
      WRITE(M,664) NPRINT,MD,MC,MA,MB,ME,STEP,RSAVE,ZSAVE,ALPHA,BETA,EE,
     1 XMSAVE
  664 FORMAT(1X,22HNPRINT,MD,MC,MA,MB,ME-,6I4/
     1 1X,37HSTEP,RSAVE,ZSAVE,ALPHA,BETA,EE,XMACH=,7F10.5)
      WRITE(M,5060) IT
C
C THEN DO CURRENTS
C
      NPRINT=NPRINT2
      MD=MD2
      MC=MC2
      MA=MA2
      MB=MB2
      ME=ME2
      STEP=STEP2
C STOP IN DENSTY IF STEP LE ZERO.
      RSAVE=RSAVE2
      ZSAVE=ZSAVE2
      ALPHA=ALPHA2
      BETA=BETA2
      EE=EE2
      XMSAVE=XMSAVE2
```

```
      CALL OVERLAY(6LDISKUS,2,0,6HRECALL)
      WRITE(M,664) NPRINT,MD,MC,MA,MB,ME,STEP,RSAVE,ZSAVE,ALPHA,BETA,EE,
     1 XMSAVE
      IF(MC2.GT.0) WRITE(M,5070) IT
      IT=IT+1
      GO TO 10
   99 STOP
      END
```

```
      SUBROUTINE ROOT(A,B,X)
      COMMON JJN,IIN,JJS,IIS,NTOT,RNT(50),ZNT(50),RST(50),ZST(50),
     1 RZ( 500,2),PHIN(20,20),PHIS(20,20),CO( 500),IFIRST,N
C FIND ROOT OF  X+B*EXP(X)=A.  BY NEWTON METHOD.
      KPRA=0
      KPRB=0
      EPS=1.E-6
      XOLD=X=0.
      KMAX=1000
      DO 100 K=1,KMAX
      XOLD=X
      KPR=K
      F=X + B*EXP(X) - A
      FP=1. + B*EXP(X)
      DX=0.
      IF(FP.GT.0.) DX=-F/FP
      X=XOLD + DX
      DELTA=DX
      IF(ABS(X).GT.1.E-8) DELTA=DX/X
      IF(KPRA.GT.0) WRITE(M,1000) K,A,B,X,DX,DELTA,F,FP
 1000 FORMAT(1X,22HK,A,B,X,DX,DELTA,F,FP=,I5,1P7E14.4)
      IF(ABS(DELTA).LT.EPS) GO TO 200
  100 CONTINUE
      WRITE(M,9999) KMAX
 9999 FORMAT(/////1X, 9HMORE THAN,I5,
     1 40H ITERATIONS IN ROOT. HENCE PROGRAM STOP.)
      STOP
C
  200 CONTINUE
      PDELTA=100.*DELTA
      IF(KPRB.GT.0) WRITE(M,2000) EPS,X,PDELTA,KPR
 2000 FORMAT(1X,35HCONVERGENCE IN ROOT WITHIN EPSILON=,1PE9.1,1H.,10X,
     1 3HX =,E12.4,7H WITHIN,E10.2,11H PERCENT IN,I4,12H ITERATIONS.)
      RETURN
      END
```

```
      OVERLAY (DISKUS,1,0)
      PROGRAM FIELD
C
C     UNSYMMETRIC DISK FIELD
C      SEIDEL METHOD
C
C
C
      COMMON JJN,IIN,JJS,IIS,NTOT,RNT(50),ZNT(50),RST(50),ZST(50),
     1 RZ( 500,2),PHIN(20,20),PHIS(20,20),RHAND( 500),IFIRST,M
      COMMON/FLD/NCOLSN,NCOLSE,NCOLSS,NROWSN,X(500),NROWSS,DEBYE,DEBYE2,
     1 RHON1(50),RHOE(50),RHOS1(50),ZN(50),   ZS(50),PHI(50),NGAP,NDISK
      COMMON/A/CN,CS,CE,CW,C,V,  INDX,JSAT,RHO,Z,INDXN( 500),INDXS( 500)
     1,INDXE( 500),INDXW( 500),CONST( 500,6)
      COMMON/DEN1/IT,NAME,NEWPHI,ISAVE,NGR,NGROUP( 500),OSAVE( 500)
********** MODIFICATION FOR FINITE DISK THICKNESS
     1 ,ZFRONT
C
C ASSUME ASYMPTOTIC MONOPOLE
      ALPHAF(RR,ZZ)=-ZZ/(ZZ**2 + RR**2)
      BETAF(RR,ZZ) =-RR/(ZZ**2 + RR**2)
C
C NTOT RESTORED AFTER MODIFICATION IN DENSTY
      NTOT=JJN*NROWSN + NCOLSE + JJS*NROWSS
      IF(IFIRST.EQ.0) GO TO 45
      IF(NEWPHI.EQ.0.AND.DEBYE.GT.0.) WRITE(M,222) DEBYE,IT,NGR,
     1 (N,RHAND(N),N=1,NTOT)
      IF(NEWPHI.GT.0.AND.DEBYE.GT.0.) WRITE(M,223) DEBYE,IT,NGR,
     1 (N,RHAND(N),N=1,NTOT)
222   FORMAT(1H1/18H0FIELD CALCULATION, 10X,
     1 41HINPUT=CHARGE DENSITIES WITH DEBYE NUMBER=,F10.5,
     2 10X,4HIT =,I3,3X,5HNGR =,I3/
     3 (28X,I3,1PE15.4))
223   FORMAT(1H1/18H0FIELD CALCULATION, 10X,
     1 41HINPUT=  ION DENSITIES WITH DEBYE NUMBER=,F10.5,
     2 10X,4HIT =,I3,3X,5HNGR =,I3/
     3 (28X,I3,1PE15.4))
C     NORTH + NORTHEAST REGION
C
C     FIRST POINT,  FIRST LINE
C
45    CONTINUE
      JSAT=0
      I=1
      IF(IFIRST.EQ.0) WRITE(M,333)
  333 FORMAT(/////25H NORTH + NORTHEAST REGION///)
      IF(IFIRST.EQ.0) WRITE(M,334) I
  334 FORMAT(//5H LINE,I3,93H                    N                    W
     1          C                   E                   S)
      J=1
      INDX=J
      INDXN(INDX)=0
      INDXS(INDX)=J+JJN
      INDXE(INDX)= INDX + 1
      INDXW(INDX)=0
      Z=ZN(I)
      RHO=RNT(J)
```

```
      HN=0.
      HS= Z - ZN(I+1)
      HE=RNT(J+1)-RHO
      HW=0.
      ALPHA=ALPHAF(RHO,Z)
      CN=0.
      CS= 0.125*HE**2/HS
      CE= HS/4.
      CW=0.
      C= 0.125*HE*(HE/HS + 2.*HS/HE - ALPHA*HE)
      V= HS*HE**2/16.
      CALL PRINT
C
C      MIDDLE POINTS,  FIRST LINE
C
      JMAX=JJN-1
      DO 5 J=2,JMAX
      INDX=J
      INDXN(INDX)=0
      INDXS(INDX)=J+JJN
      INDXE(INDX)=INDX+1
      INDXW(INDX)=INDX-1
      Z=ZN(I)
      RHO=RNT(J)
      HN=0.
      HS=Z-ZN(I+1)
      HE=RNT(J+1)-RHO
      HW=RHO-RNT(J-1)
      ALPHA=ALPHAF(RHO,Z)
      CN=0.
      CS=0.5*(HE+HW)/HS*(RHO+(HE-HW)/4.)
      CE=0.5*HS/HE*(RHO+HE/2.)
      CW=0.5*HS/HW*(RHO-HW/2.)
      C=0.5*HS*(HE+HW)*(RHO/HE/HW+(1.-ALPHA*HS)/HS**2*(RHO+(HE-HW)/4.))
      V=0.25*HS*(HE+HW)*(RHO+(HE-HW)/4.)
    5 CALL PRINT
C
C      LAST POINT FIRST LINE
C
      J=JJN
      INDX=J
      INDXN(INDX)=0
      INDXS(INDX)=J+JJN
      INDXE(INDX)=0
      INDXW(INDX)=INDX-1
      Z=ZN(I)
      RHO=RNT(J)
      HN=0.
      HS=Z-ZN(I+1)
      HE=0.
      ALPHA=ALPHAF(RHO,Z)
      BETA=BETAF(RHO,Z)
      HW=RHO-RNT(J-1)
      CN=0.
      CS=0.5*HW/HS*(RHO-HW/4.)
      CE=0.
      CW=0.5*HS/HW*(RHO-HW/2.)
```

```
      C=0.5*((HW/HS+HS/HW-ALPHA*HW)* (RHO-HW/4.)-HS*(BETA*RHO+ 0.25)
      V=0.25*HS*HW*(RHO-HW/4.)
      CALL PRINT
C
C     MIDDLE LINES
C
      IMAX=NROWSN-1
      IF(IMAX.LT.2) GO TO 16
      DO 10 I=2,IMAX
      IF (IFIRST.EQ.0) WRITE (M,334) I
      DO 10 J=1,JJN
      INDX=J+(I-1)*JJN
      INDXN(INDX)=INDX-JJN
      INDXS(INDX)=INDX+JJN
      Z=ZN(I)
      HN=ZN(I-1)-Z
      HS=Z-ZN(I+1)
      RHO=RNT(J)
      JGO=2
      IF(J.EQ.1) JGO=1
      IF(J.EQ.JJN) JGO=3
      GO TO (6,7,8),JGO
    6 INDXE(INDX)=INDX+1
      INDXW(INDX)=0.
      HE=RNT(J+1)-RHO
      HW=0.
      CALL LEFT(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      GO TO 9
    7 INDXE(INDX)=INDX+1
      INDXW(INDX)=INDX-1
      HE=RNT(J+1)-RHO
      HW=RHO-RNT(J-1)
      CALL MIDDLE(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      GO TO 9
    8 INDXE(INDX)=0
      INDXW(INDX)=INDX-1
      HE=0.
      HW=RHO-RNT(J-1)
      BETA=BETAF(RHO,Z)
      CALL RIGHT (RHO,HN,HS,HE,HW,BETA,CN,CS,CE,CW,C,V)
    9 CALL PRINT
   10 CONTINUE
C
C     LAST LINE ABOVE DISK SURFACE, NORTH-NORTHEAST REGION
C
   16 I=NROWSN
      IF (IFIRST.EQ.0) WRITE (M,334) I
      DO 11 J=1,NCOLSN
      INDX=J+(I-1)*JJN
      INDXN(INDX)=INDX-JJN
      INDXS(INDX)=0
      INDXE(INDX)=INDX+1
      IF(J.GT.1)INDXW(INDX)=INDX-1
      IF(J.EQ.1)INDXW(INDX)=0
      Z=ZN(I)
      RHO=RNT(J)
      HN=ZN(I-1)-Z
```

```
      HS=Z
      HE=RNT(J+1)-RHO
      IF(J.GT.1) HW=RHO-RNT(J-1)
      IF(J.EQ.1) HW=0.
      IF(J.EQ.1) CALL LEFT(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      IF(J.GT.1)CALLMIDDLE(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      JSAT=J
      CALL PRINT
   11 CONTINUE
C
C     LAST LINE AWAY FROM DISK SURFACE, NORTH-NORTHEAST REGION
C
      JMIN=NCOLSN+1
      DO 15 J=JMIN,JJN
      INDX=J+(I-1)*JJN
      INDXN(INDX)=INDX-JJN
      INDXS(INDX)=INDX+NCOLSE
      IF(J.LT.JJN) IGO=1
      IF(J.EQ.JJN) IGO=2
      INDXW(INDX)=INDX-1
      Z=ZN(I)
      RHO=RNT(J)
      HN=ZN(I-1)-Z
      HS=Z
      HW=RHO-RNT(J-1)
      GO TO (12,13), IGO
   12 INDXE(INDX)=INDX+1
      HE=RNT(J+1)-RHO
      CALL MIDDLE(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      GO TO 14
   13 INDXE(INDX)=0
      HE=0
      BETA=BETAF(RHO,Z)
      CALL RIGHT(RHO,HN,HS,HE,HW,BETA,CN,CS,CE,CW,C,V)
   14 CALL PRINT
   15 CONTINUE
C
C
C     EAST REGION
C
C
      IF(IFIRST.EQ.0) WRITE(M,444)
  444 FORMAT(/////12H EAST REGION///)
      I=1
      IF(IFIRST.EQ.0) WRITE(M,334) I
      DO 26 J=1,NCOLSE
      INDX=J+NROWSN*JJN
      INDXN(INDX)=INDX-NCOLSE
      INDXS(INDX)=INDX+JJS
      RHO=RHOE(J)
      Z=0.
      JGO=2
      IF(J.EQ.1)JGO=1
      IF(J.EQ.NCOLSE) JGO=3
      HN=ZN(NROWSN)
      HS=-ZS(1)
      GO TO (20,21,22),JGO
```


FIELD

```
   20 INDXE(INDX)=INDX+1
      INDXW(INDX)=0
      JSAT=NGAP
      HE=RHOE(J+1) - RHO
      HW=RHO-RNT(NCOLSN)
      GO TO 23
   21 INDXE(INDX)=INDX+1
      INDXW(INDX)=INDX-1
      HE=RHOE(J+1) - RHO
      HW=RHO - RHOE(J-1)
      GO TO 23
   22 INDXE(INDX)=0
      INDXW(INDX)=INDX-1
      HE=0.
      HW=RHO - RHOE(J-1)
      BETA=BETAF(RHO,Z)
      GO TO 24
   23 CALL MIDDLE(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      CALL PRINT
      GO TO 26
   24 CALL RIGHT(RHO,HN,HS,HE,HW,BETA,CN,CS,CE,CW,C,V)
      CALL PRINT
   26 CONTINUE
C
C
C     SOUTH + SOUTHEAST REGION
C
C
      IF(IFIRST.EQ.0) WRITE(M,555)
  555 FORMAT(/////25H SOUTH + SOUTHEAST REGION///)
      DO 41 I=1,NROWSS
      IF(IFIRST.EQ.0) WRITE(M,334) I
      DO 41 J=1,JJS
      INDX=J+(I-1)*JJS+NCOLSE+NROWSN*JJN
      RHO=RST(J)
      Z=ZS(I)
      IGO=2
      JGO=2
      IF(I.EQ.1) IGO=1
      IF(I.EQ.NROWSS) IGO=3
      IF(J.EQ.1) JGO=1
      IF(J.EQ.JJS) JGO=3
      GO TO (28,29,30), IGO
C
C     FIRST LINE
C
   28 INDXN(INDX)=INDX-JJS
      IF(J.LE.NCOLSS) INDXN(INDX)=0
      IF(J.LE.NCOLSS)  JSAT=NDISK+1-J
      INDXS(INDX)=INDX+JJS
      HN=-Z
      HS=Z - ZS(I+1)
      GO TO 31
C
C     MIDDLE LINES
C
   29 INDXN(INDX)=INDX-JJS
```


FIELD

```
      INDXS(INDX)=INDX+JJS
      HN=ZS(I-1) - Z
      HS=Z - ZS(I+1)
      GO TO 31
C
C     LAST LINE
C
   30 INDXN(INDX)=INDX-JJS
      INDXS(INDX)=0
      HN=ZS(I-1) - Z
      HS=0.
      ALPHA=-ALPHAF(RHO,Z)
   31 GO TO (34,35,36), JGO
C
C     FIRST POINT
C
   34 INDXE(INDX)=INDX+1
      INDXW(INDX)=0
      HE=RST(J+1)-RHO
      HW=0.
      IF(IGO.EQ.3) GO TO 33
      CALL LEFT(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      GO TO 40
C
C     MIDDLE POINTS
C
   35 INDXE(INDX)=INDX+1
      INDXW(INDX)=INDX-1
      HE=RST(J+1)-RHO
      HW=RHO-RST(J-1)
      IF(IGO.EQ.3) GO TO 33
      CALL MIDDLE(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      GO TO 40
C
C     LAST POINT
C
   36 INDXE(INDX)=0
      INDXW(INDX)=INDX-1
      HE=0.
      HW=RHO-RST(J-1)
      BETA=BETAF(RHO,Z)
      IF (IGO.EQ.3) GO TO 33
      CALL RIGHT(RHO,HN,HS,HE,HW,BETA,CN,CS,CE,CW,C,V)
      GO TO 40
   33 GO TO (37,38,39),JGO
C
C     LAST LINE, FIRST POINT
C
   37 CN=0.125*HE**2/HN
      CS=0.
      CE=HN/4.
      CW=0.
      C=0.125*HE*(HE/HN + 2.*HN/HE - ALPHA*HE)
      V=HN*HE**2/16.
      GO TO 40
C
C     LAST LINE, MIDDLE POINTS
```

```
C
   38 CN=0.5*(HE+HW)/HN*(RHO+(HE-HW)/4.)
      CS=0.
      CE=0.5*HN/HE*(RHO+HE/2.)
      CW=0.5*HN/HW*(RHO-HW/2.)
      C=0.5*HN*(HE+HW)*(RHO/HE/HW +  (1.-ALPHA*HN)*(RHO+(HE-HW)/4.)/HN**
     12)
      V=0.250*HN*(HE+HW)*(RHO + (HE-HW)/4.)
      GO TO 48
C
C     LAST LINE, LAST POINT
C
   39 CN=0.5*HW/HN*(RHO-HW/4.)
      CS=0.
      CE=0.
      CW=0.5*HN/HW*(RHO-HW/2.)
      C=0.5*((HW/HN+HN/HW-ALPHA*HW)*(RHO-HW/4.)- HN*(BETA*RHO + 0.25))
      V=0.25*HN*HW*(RHO-HW/4.)
   40 CALL PRINT
   41 CONTINUE
      IF(IFIRST.GT.0) CALL SEIDEL
      END
```

```
      SUBROUTINE LEFT(RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      CN=0.125*HE**2/HN
      CS=CN*HN/HS
      CE=0.25*(HN+HS)
      CW=0.
      C=0.25*(HN+HS)*(1.+0.5*HE**2/HN/HS)
      V=HE**2*(HN+HS)/16.
      RETURN
      END
```

```
      SUBROUTINE MIDDLE (RHO,HN,HS,HE,HW,CN,CS,CE,CW,C,V)
      CN=0.5*(HE+HW)/HN*(RHO+(HE-HW)/4.)
      CS=CN*HN/HS
      CE=0.5*(HN+HS)/HE*(RHO+HE/2.)
      CW=0.5*(HN+HS)/HW*(RHO-HW/2.)
      C=0.5*(HN+HS)*(HE+HW)*(RHO/HE/HW+(RHO+(HE-HW)/4.)/HN/HS)
      V=0.25*(HN+HS)*(HE+HW)*(RHO+(HE-HW)/4.)
      RETURN
      END
```

```
      SUBROUTINE RIGHT (RHO,HN,HS,HE,HW,BETA,CN,CS,CE,CW,C,V)
      CN=0.5*HW/HN*(RHO-HW/4.)
      CS=CN*HN/HS
      CE=0.
      CW=0.5*(HN+HS)/HW*(RHO-HW/2.)
      C=0.5*(HN+HS)*(HW/HN/HS*(RHO-HW/4.)+(RHO-HW/2.)/HW - BETA*RHO)
      V=0.25*(HN+HS)*HW*(RHO-HW/4.)
      RETURN
      END
```

```
      SUBROUTINE PRINT
      COMMON JJN,IIN,JJS,IIS,NTOT,RNT(50),ZNT(50),RST(50),ZST(50),
     1 RZ( 500,2),PHIN(20,20),PHIS(20,20),RHAND( 500),IFIRST,M
      COMMON/FLD/NCOLSN,NCOLSE,NCOLSS,NROWSN,X(500),NROWSS,DEBYE,DEBYE2,
     1 RHON1(50),RHOE(50),RHOS1(50),ZN(50),  ZS(50),PHI(50),NGAP,NDISK
      COMMON/A/CN,CS,CE,CW,C,V,  INDX,JSAT,RHO,Z,INDXN( 500),INDXS( 500)
     1,INDXE( 500),INDXW( 500),CONST( 500,6)
      CP=0.
      CONST(INDX,1)= CN
      IF((INDXN(INDX).EQ.0 ).AND.(CN.NE.0.)) CP=CN
      CONST(INDX,2)= CS
      IF((INDXS(INDX).EQ.0 ).AND.(CS.NE.0.)) CP=CS
      CONST(INDX,3)= CE
      CONST(INDX,4)= CW
      IF((INDXW(INDX).EQ.0 ).AND.(CW.NE.0.)) CP=CW
C ********** TEMPORARY --- HELMHOLTZ EQUATION
      IF(DEBYE2.GT.0.) C=C+ V/(DEBYE2**2)
C ********** TEMPORARY --- HELMHOLTZ EQUATION
      CONST(INDX,5)= C
      CONST(INDX,6)= V
      IF (CP.GT.0..AND.DEBYE.EQ.0.) RHAND(INDX)= CP*PHI(JSAT)
      IF (CP.GT.0..AND.DEBYE.GT.0.) RHAND(INDX) = RHAND(INDX)*V/DEBYE**2
     1 + CP*PHI(JSAT)
      IF (CP.EQ.0..AND.DEBYE.GT.0.) RHAND(INDX) = RHAND(INDX)*V/DEBYE**2
      IF (IFIRST.GT.0) GO TO 3
      WRITE(M,1)   INDX, INDXN(INDX), CONST(INDX,1), INDXW(INDX), CONST(
     1INDX,4), INDX,CONST(INDX,5), INDXE(INDX), CONST(INDX,3), INDXS(IN
     2DX), CONST(INDX,2), CONST(INDX,6)
    1 FORMAT( /   6H POINT,I4,     3H/C(,I4,2H)=,1PE10.4,3H/C(,I4,2H)=,
     1E10.4,  3H/C(,I4,2H)=,E10.4,  3H/C(,I4,2H)=,E10.4,  3H/C(,I4,2H)=,
     2E10.4,  5H/VOL=,E10.4)
      IF  (CP.NE.0.) WRITE (M,2)  JSAT, CP
    2 FORMAT(31H COEFFICIENT OF POTENTIAL NO. (,I3, 4H) IS,F10.5)
    3 RZ(INDX,1)=RHO
      RZ(INDX,2)=Z
      RETURN
      END
```

```
      SUBROUTINE SEIDEL
      COMMON JJN,IIN,JJS,IIS,NTOT,RNT(50),ZNT(50),RST(50),ZST(50),
     1 RZ( 500,2),PHIN(20,20),PHIS(20,20),RHAND( 500),IFIRST,M
      COMMON/FLD/NCOLSN,NCOLSE,NCOLSS,NROWSN,X(500),NROWSS,DEBYE,DEBYE2,
     1 RHON1(50),RHOE(50),RHOS1(50),ZN(50),   ZS(50),PHI(50),NGAP,NDISK
      COMMON/A/CN,CS,CE,CW,C,V,  INDX,JSAT,RHO,Z,INDXN( 500),INDXS( 500)
     1,INDXE( 500),INDXW( 500),CONST( 500,6)
      COMMON/DEN1/IT,NAME,NEWPHI,ISAVE,NGR,NGROUP( 500),OSAVE( 500)
**********  MODIFICATION FOR FINITE DISK THICKNESS
     1 , ZFRONT
      RADIUS=RHON1(NCOLSN)
      OMEGA=1.9
      EPS= 0.00001
      ITMAX=2000
      ITCOUN =0
      IPROLD=0
      IGO=1
      IF(NEWPHI.GT.0.AND.DEBYE.GT.0.) WRITE(M,100)
100   FORMAT(/1X,44HMODIFIED POISSON PROBLEM TO INCLUDE EXP(PHI))
      IF(IT.GT.0.AND.IFIRST.GT.0) GO TO 2
      DO 1 K =1,NTOT
    1 X(K)=0.
    2 ITCOUN =ITCOUN +1
      DELTAM=0.
      DO 3 K=1,NTOT
      X1=X(K)
      SN=CONST(K,1)/CONST(K,5)
      SS=CONST(K,2)/CONST(K,5)
      SE=CONST(K,3)/CONST(K,5)
      SW=CONST(K,4)/CONST(K,5)
      SR=RHAND(K)/CONST(K,5)
      INDXNK=INDXN(K)
      INDXSK=INDXS(K)
      INDXEK=INDXE(K)
      INDXWK=INDXW(K)
      AA=SR
      IF(INDXNK.GT.0) AA=AA+SN*X(INDXNK)
      IF(INDXSK.GT.0) AA=AA+SS*X(INDXSK)
      IF(INDXEK.GT.0) AA=AA+SE*X(INDXEK)
      IF(INDXWK.GT.0) AA=AA+SW*X(INDXWK)
      IF(NEWPHI.EQ.0.OR.DEBYE.EQ.0.) GO TO 30
C MODIFICATION TO INCLUDE EXP(PHI) IN POISSON PROBLEM
      BB=CONST(K,6)/CONST(K,5)/DEBYE**2
      CALL ROOT(AA,BB,XX)
      X(K)=XX
      GO TO 35
30    X(K)=AA
35    CONTINUE
C
C
C  SET PHI=0 AT ZN BOUNDARY
      IF(K.LE.JJN.AND.(IFIRST.GT.0.OR.NEWPHI.GT.0)) X(K)=0.
**********  MODIFICATION FOR FINITE DISK THICKNESS
C SET PHI TO BODY POTENTIAL AT ADDITIONAL BODY POINTS IF ANY.
      IF(RZ(K,1).LE.RADIUS.AND.RZ(K,2).GE.ZFRONT.AND.RZ(K,2).LT.0.)
     1 X(K)=PHI(1)
C
```

```
      X(K)=OMEGA*X(K)+(1.-OMEGA)*X1
      DELTA=ABS(X(K)-X1)
      IF(X1.NE.0.) DELTA=ABS((X(K)-X1)/X1)
      IF(DELTA.GT.DELTAM) DELTAM=DELTA
    3 CONTINUE
      IF(ITCOUN.GT.ITMAX) WRITE(M,11) ITMAX
      IF(ITCOUN.GT.ITMAX) GO TO 9
   11 FORMAT(////10H MORE THAN,I4,10HITERATIONS)
      IPR=ITCOUN/500
      IF(IPR.LE.IPROLD) GO TO 15
      IPROLD=IPR
      GO TO 10
   15 IF (DELTAM.GT.EPS) GO TO 2
C
C     ITERATION FINISHED
C
    9 IGO=2
   10 NFPP=(NTOT/300) + 1
      DO 51 IP=1,NFPP
      WRITE(M,22) ITCOUN,EPS,DELTAM,OMEGA
      DO 51 I=1,60
      K1=I + 300*(IP-1)
      K2=K1 + 60
      K3=K2 + 60
      K4=K3 + 60
      K5=K4 + 60
      IF(K5.LE.NTOT)WRITE(M,3333)K1,X(K1),K2,X(K2),K3,X(K3),K4,X(K4),
     1K5,X(K5)
      IF(K5.LE.NTOT) GO TO 51
      IF(K4.LE.NTOT)WRITE(M,3333)K1,X(K1),K2,X(K2),K3,X(K3),K4,X(K4)
      IF(K4.LE.NTOT) GO TO 51
      IF(K3.LE.NTOT)WRITE(M,3333)K1,X(K1),K2,X(K2),K3,X(K3)
      IF(K3.LE.NTOT) GO TO 51
      IF(K2.LE.NTOT)WRITE(M,3333)K1,X(K1),K2,X(K2)
       IF(K2.LE.NTOT) GO TO 51
      IF(K1.LE.NTOT)WRITE(M,3333)K1,X(K1)
   51 CONTINUE
 3333 FORMAT(5(I8,F16.8))
      GO TO (15,4),IGO
   22 FORMAT(15H1SOLUTION AFTER,I6,2X,25HITERATIONS WITH TOLERANCE,F12.8
     1,8X,18HMAXIMUM DIFFERENCE,F12.8,8X,6HOMEGA=,F8.5)
    4 RETURN
      END
```

```
      OVERLAY (DISKUS,2,0)
      PROGRAM DENSTY
      COMMON JJN,IIN,JJS,IIS,ND,RN(50),ZN(50),RS(50),ZS(50),RSV( 500),
     1 ZSV( 500),PHIN(20,20),PHIS(20,20),COSV( 500),IFIRST,M
      COMMON/DEN/NPRINT,MD,MC,MA,MB,ME,STEP,RSAVE,ZSAVE,ALPHA,BETA,EE,
     1 XMSAVE,RADIUS
      COMMON/DEN1/IT,MAME,NEWPHI,ISAVE,NGR,NGROUP( 500),DSAVE( 500)
*********** MODIFICATION FOR FINITE DISK THICKNESS
     1 , ZFRONT
      COMMON/P/INT,IGN,JGN,IGS,JGS
      COMMON/T/X,Y,Z,R,XDOT,YDOT,ZDOT,PHI,PHIR,PHIZ,ERGO,DERG,C1,C2,E
      DIMENSION A(2),DNORTH(20,20),DSOUTH(20,20)
      DIMENSION PARTCL(2),PART1(2),PART2(2),FATE(2),END1(2),END2(2)
      DATA PART1/6H ION  ,6H      /,PART2/6H ELECT,6HRON   /
      DATA END1/6H ABSOR,6HBED   /,END2/6H ESCAP,6HES   /
      PI = 3.1415926536
      A(1)=-1./SQRT(3.)
      A(2)=-A(1)
      MSTEP=10000
      MSTEP=20000
      IPRINT=1
      IF(MC.GT.0.OR.NGR.GT.1    ) IPRINT=0
      NTOT=ND
      NPTS=ND
      IINM=IIN-1
      IISM=IIS-1
      N1=IINM*JJN
      N2=NTOT-IISM*JJS
      DO 5 I=1,IIN
      DO 5 J=1,JJN
5     DNORTH(I,J)=0.
      DO 6 I=1,IIS
      DO 6 J=1,JJS
6     DSOUTH(I,J)=0.
50    CONTINUE
C DO ONE CHARGE DENSITY OR CURRENT DENSITY, OR DO ALL
      IF(MD.EQ.0) NPTS=1
      IF(MD.EQ.0)  RSV(1)=RSAVE
      IF(MD.EQ.0)  ZSV(1)=ZSAVE
      IF(MC.EQ.0) WRITE(M,666) IT,NGR
      IF(MC.GT.0) WRITE(M,667) IT
      WRITE(M,664)NPRINT,MD,MC,MA,MB,ME,STEP,RSAVE,ZSAVE,
     1 ALPHA,BETA,EE,XMSAVE
666   FORMAT(1H1/10HDENSITIES,5X,4HIT =,I3,3X,5HNGR =,I3)
667   FORMAT(1H1/9HCURRENTS,5X,4HIT =,I3)
664   FORMAT( X,22HNPRINT,MD,MC,MA,MB,ME=,6I4/
     1 1X,37HSTEP,RSAVE,ZSAVE,ALPHA,BETA,EE,XMACH=,7F10.5/)
C
      IF(NPRINT.EQ.0) WRITE(M,660) NPRINT
      IF(NPRINT.EQ.1) WRITE(M,661) NPRINT
      IF(NPRINT.EQ.2) WRITE(M,662) NPRINT
      IF(NPRINT.EQ.3) WRITE(M,663) NPRINT
660   FORMAT(1X,8HNPRINT =,I2,38H MEANS NO TRAJECTORY PRINTING          )
661   FORMAT(1X,8HNPRINT =,I2,38H INDICES OF ESCAPING TRAJECTORIES ONLY)
662   FORMAT(1X,8HNPRINT =,I2,38H=FIRST + LAST STEPS OF EACH TRAJECTORY)
663   FORMAT(1X,8HNPRINT =,I2,38H EVERY STEP OF ALL TRAJECTORIES        )
C
```

```
      MOSTPS=0
C DO 95 LOOP ENDS AT END OF PROGRAM
      IF(MD.GT.C.AND.MC.EQ.0) NPTS=ND+1
60    DO 95 N=1,NPTS
      RSAVE=RSV(N)
      ZSAVE=ZSV(N)
      IF(N.LE.N1) GO TO 7
      IF(N.GT.N2) GO TO 8
7     DO 1 I=1,IIN
      IF(ZSAVE.EQ.ZN(I)) IN=I
1     CONTINUE
      DO 2 J=1,JJN
      IF(RSAVE.EQ.RN(J)) JN=J
2     CONTINUE
      IF(N.LE.N1) GO TO 9
8     DO 3 I=1,IIS
      IF(ZSAVE.EQ.ZS(I)) IS=I
3     CONTINUE
      DO 4 J=1,JJS
      IF(RSAVE.EQ.RS(J)) JS=J
4     CONTINUE
9     CONTINUE
      IF(N.LE.N2.AND.IN.GT.0.AND.JN.GT.0) DNORTH(IN,JN)=DSAVE(N)
      IF(N.GT.N1.AND.IS.GT.0.AND.JS.GT.0) DSOUTH(IS,JS)=DSAVE(N)
      IF(NGR.EQ.0) GO TO 15
      IF(IT.GT.0.AND.N.LT.NPTS.AND.NGROUP(N).NE.NGR) CDSV(N)=DSAVE(N)
      IF(IT.GT.0.AND.N.LT.NPTS.AND.NGROUP(N).NE.NGR) GO TO 95
15    CONTINUE
********** MODIFICATION FOR FINITE DISK THICKNESS
      IF(RSAVE.LE.RADIUS.AND.ZSAVE.GE.ZFRONT.AND.ZSAVE.LT.0.) CDSV(N)=0.
      IF(RSAVE.LE.RADIUS.AND.ZSAVE.GE.ZFRONT.AND.ZSAVE.LT.0.) GO TO 95
      IF(MC.EQ.0.AND.ISAVE.EQ.0) DSAVE(N)=0.
      MASAVE=MA
      MBSAVE=MB
      MESAVE=ME
      STEPSV=STEP
      INCREA=0
      IF(MC.GT.0.OR.MAME.EQ.0) GO TO 20
***** INCREASE ACCURACY NEAR AXIS
      IF(RSAVE.LE.RN(2).AND.ZSAVE.GT.0.) MA=ME=16
      IF(RSAVE.LE.RN(2).AND.ZSAVE.GT.0.) STEP=.05
      IF(RSAVE.LE.RN(2).AND.ZSAVE.GT.0.) INCREA=1
20    CONTINUE
C FIRST WE DO THE IONS
      SCALE=1.
      PARTCL(1)=PART1(1)
      PARTCL(2)=PART1(2)
C RETURN FROM END OF MAIN FOR ELECTRONS
  237 IF(SCALE.GT.0..AND.N.EQ.1.AND.IPRINT.EQ.1) WRITE(M,3000)
     1 (RN(J),J=1,JJN)
3000  FORMAT(///,1X,24H POTENTIAL ARRAY - NORTH//1X,2HR=,16F8.4/
     1 (/3X,16F8.4))
C
      IF(SCALE.GT.0.) XMACH=XMSAVE
      IF(SCALE.LT.0.) XMACH=0.
      PHIMAX = 0.
      DO 11 I=1,IIN
```

```
      DO 10 J=1,JJN
   10 PHIN(I,J)=SCALE*PHIN(I,J)
      IF (PHIMAX.LT.PHIN(I,1)) PHIMAX = PHIN(I,1)
      IF(SCALE.EQ.-1..OR.N.GT.1.OR.IPRINT.EQ.0) GO TO 11
      WRITE(M,333) I,ZN(I),(PHIN(I,J),J=1,JJN)
   11 CONTINUE
  333 FORMAT(/5H LINE,I4,5X,2HZ=,F8.4//(/7F16.8))
C
      IF(SCALE.GT.0..AND.N.EQ.1.AND.IPRINT.EQ.1) WRITE(M,4000)
     1 (RS(J),J=1,JJS)
 4000 FORMAT(////24H POTENTIAL ARRAY - SOUTH//1X,2HR=,16F8.4//
     1 (/3X,16F8.4))
C
      DO 13 I=1,IIS
      DO 12 J=1,JJS
   12 PHIS(I,J)=SCALE*PHIS(I,J)
      IF(SCALE.EQ.-1..OR.N.GT.1.OR.IPRINT.EQ.0) GO TO 13
      WRITE(M,333) I,ZS(I),(PHIS(I,J),J=1,JJS)
   13 CONTINUE
C
C SET UP SUMS OVER TRAJECTORIES
C
      IF (MA.EQ.0) GO TO 32
      JAMAX=2
      JBMAX=2
      KAMAX=MA
      KBMAX=MB
      NUMBER=MA*MB*4
C DO ONLY ONE BETA ON AXIS (SYMMETRY)
      IF(RSAVE.EQ.0.) KBMAX=1
      IF(RSAVE.EQ.0.) NUMBER=MA*2
      IF(RSAVE.EQ.0.) JBMAX=1
      IF(SCALE.GT.0..AND.N.EQ.1) WRITE(M,668) MA,MB,NUMBER
  668 FORMAT(/1X,I6,16H ALPHA-INTERVALS,1X,I6, 15H BETA-INTERVALS,1X,I6,
     1 24H TRAJECTORIES PER ENERGY)
C
      IF(ME.EQ.0) GO TO 31
      JEMAX=2
      KEMAX=ME
      IF(SCALE.GT.0..AND.N.EQ.1) WRITE(M,670) ME
  670 FORMAT(1X,I6,47H ENERGY INTERVALS, WITH 2 ENERGIES PER INTERVAL//)
C
      GO TO 33
C
C SINGLE ENERGY
C
   31 JEMAX=1
      KEMAX=1
      IF(SCALE.GT.0..AND.N.EQ.1) WRITE(M,673) EE
  673 FORMAT(1X,31H MONOENERGETIC CASE WITH ENERGY,F10.5//)
C
      GO TO 33
C
C SINGLE TRAJECTORY ONLY
C
   32 JAMAX=1
      JBMAX=1
```

```
      JEMAX=1
      KAMAX=1
      KBMAX=1
      KEMAX=1
      NUMBER=1
      WRITE (M,669) ALPHA, BETA,EE
  669 FORMAT( 18H SINGLE TRAJECTORY/ 7H ALPHA=,F20.18, 8H DEGREES/
     1  6H BETA=,F20.18, 8H DEGREES/  8H ENERGY=,F10.6)
C
      ALPHA=ALPHA*PI/180.
      BETA=BETA*PI/180.
      WRITE( M,665) ALPHA,BETA
  665 FORMAT( 1X, 3HOR,/ 1X, 6HALPHA=,F11.8, 8H RADIANS/ 1X, 5HBETA=,F10
     1.8, 8H RADIANS)
C
      SINA=SIN(ALPHA)
      COSA=COS(ALPHA)
C
C SUM OVER ALPHA, BETA, AND ENERGY
C
C
   33 DENST=0.
      DO 1001 KE=1,KEMAX
      DO 1001 JE=1,JEMAX
      DENS=0.
      NOESC=0
      DO 1000 KB=1, KBMAX
      DO 1000 JB=1, JBMAX
      DO 1000 KA=1, KAMAX
      DO 1000 JA=1, JAMAX
C
C INITIAL POSITION
C
      R=RSAVE
      Z=ZSAVE
      X=R
      Y=0.
      INT=0
      CALL INTERP
      INT=1
      PHISAV=PHI
***** ASSUME BOLTZMANN FACTOR FOR ELECTRONS (OVERRIDE)
      IF(ABS(PHI).GT.500.) GO TO 96
      IF(MC.EQ.0.AND.SCALE.LT.0.) DENST=EXP(-PHI)
      IF(MC.EQ.0.AND.SCALE.LT.0.) GO TO 96
      IF(MC.EQ.0.AND.ISAVE.GT.0) DENST=DSAVE(N)
      IF(MC.EQ.0.AND.ISAVE.GT.0) GO TO 96
C
      IF(STEP.LE.0.) WRITE(M,111)
  111 FORMAT(/////1X,43HSTOP DUE TO STEP LE. ZERO ***** ***** *****)
      IF(STEP.LE.0.) STOP
C
C INITIAL VELOCITY
C
      SPEED=0.
      IF(ME.NE.0) GO TO 41
      E=EE
```

```
      IF(E.LT.PHI) WRITE(M,674) KE,JE,KB,JB,KA,JA,BETA1,ALPHA1,E,PHISAV
      IF(E.LT.PHI) GO TO 1001
      GO TO 40
   41 CE=(A(JE) + FLOAT(2*KE-1-ME))/FLOAT(ME)
      E= (1.+CE)/(1.-CE) + AMAX1(PHI    ,0.)
      IF(XMACH.GT.1.) E=XMACH**2*(1.+CE)/(1.-CE) + AMAX1(PHI,0.)
      IF(E.LT.0. ) WRITE(M,674) KE,JE,KB,JB,KA,JA,BETA1,ALPHA1,E,PHISAV
      IF(E.LT.0. ) STOP
C
   40 SPEED=SQRT(E - PHI)
      IF(MA.EQ.0) GO TO 39
      CA  =(A(JA) + FLOAT(2*KA - 1 - MA))/FLOAT(MA)
      IF(MC.EQ.0) COSA=CA
      POWER=XMACH
      IF(MC.EQ.0.AND.POWER.GT.1.)COSA=-1.+2.*((1.+CA)/2.)**POWER
      IF(MC.EQ.0) SINA=SQRT(1. - COSA**2)
      IF(MC.GT.0) SINA=SQRT(.5*(1. + CA))
      IF(MC.GT.0) COSA=SQRT(1. - SINA**2)
      CBETA=(A(JB)+ FLOAT(2*KB - 1 - MB))/FLOAT(MB)
      BETA=PI*(1.+ CBETA)/2.
C
   39 XDOT=SPEED*SINA*COS(BETA)
      YDOT=SPEED*SINA*SIN(BETA)
      ZDOT=SPEED*COSA
      C1=STEP*E
      C2=SQRT(C1)
      KSTEP=0
      ERGMAX=0.
      ERGO=0.
      DERGMX=0.
      DERG=0.
      ALPHA1 = ACOS(COSA)*180./PI
      BETA1 = BETA*180./PI
      ALPHA =ALPHA1
      BETA=BETA1
      ZOLD=Z
      IF(NPRINT.NE.2.AND.NPRINT.NE.3) GO TO 34
C PRINT INITIAL CONDITIONS OF TRAJECTORY
      WRITE(M,674) KE,JE,KB,JB,KA,JA,BETA1,ALPHA1,E,PHISAV
  674 FORMAT(1X,3(I3,I2),F17.8,F14.8,2X,1P2E11.3,2X,46H =KE,JE,  KB,JB,
     1 KA,JA,  BETA1,ALPHA1,  E,PHI)
C
      WRITE(M,659)
  659 FORMAT(13X,115HSTEPS      X           Y           Z         XDOT
     1 YDOT        ZDOT       ERGMAX      DERGMX                        )
C
      WRITE(M,888) KSTEP,X,Y,Z,XDOT,YDOT,ZDOT
  888 FORMAT(13X,I5,1P6E11.3)
C
C TAKE A STEP
C
C
   34 CALL TRACK
      KSTEP=KSTEP+1
      IF(NPRINT.EQ.3)WRITE(M,888) KSTEP,X,Y,Z,XDOT,YDOT,ZDOT
      IF(ABS(ERGMAX).LT.ABS(ERGO))ERGMAX=ERGO
      IF(ABS(DERGMX).LT.ABS(DERG)) DERGMX=DERG
```

```
      IF(KSTEP.LT.MSTEP) GO TO 35
      WRITE (M,999) MSTEP
      WRITE(M,97) KSTEP,N,KE,JE,KB,JB,KA,JA
  999 FORMAT(10H MORE THAN,I10,5HSTEPS)
      STOP
   35 R=SQRT(X*X+Y*Y)
      IF (R.LT.RADIUS.AND.SIGN(1.,Z).NE.SIGN(1.,ZOLD)) GO TO 36
**********  MODIFICATION FOR FINITE DISK THICKNESS
      IF(R.LT.RADIUS.AND.Z.GT.ZFRONT.AND.Z.LT.0.) GO TO 36
      IF(R.GT.RN(JJN).OR.Z.GT.ZN(1).OR.Z.LT.ZS(IIS)) GO TO 37
      ZOLD=Z
      CALL INTERP
      GO TO 34
C
C PARTICLE IS ABSORBED
C
   36 CONTINUE
      IF(NPRINT.NE.2.AND.NPRINT.NE.3) GO TO 1002
      FATE(1)=END1(1)
      FATE(2)=END1(2)
      GO TO 374
C
C PARTICLE ESCAPES
C
C
37    IF(NPRINT.EQ.1) GO TO 372
      IF(NPRINT.NE.2.AND.NPRINT.NE.3) GO TO 373
      FATE(1)=END2(1)
      FATE(2)=END2(2)
      GO TO 373
372   WRITE(M,674) KE,JE,KB,JB,KA,JA,BETA1,ALPHA1,E,PHISAV
373   NOESC=NOESC+1
      IF(ME.EQ.0) GO TO 374
C
      CSANGL=ZDOT/SQRT(XDOT**2 + YDOT**2 + ZDOT**2)
      XPON=-2.*XMACH*SQRT(E)*CSANGL - E - XMACH**2
      IF(MC.EQ.0) COEFF1=SPEED/FLOAT(NUMBER)
      IF(MC.EQ.0.AND.POWER.GT.1.)
     1 COEFF1=COEFF1*POWER*((1.+CA)/2.)**(POWER-1.)
      IF(ABS(XPON).GT.500.) GO TO 374
      IF(MC.GT.0) COEFF1=SPEED**2/FLOAT(NUMBER)
      DADD=COEFF1*EXP(XPON)
      DENS=DENS + DADD
C
  374 IF(NPRINT.NE.2.AND.NPRINT.NE.3) GO TO 1002
      WRITE(M,889) FATE,KSTEP,X,Y,Z,XDOT,YDOT,ZDOT,ERGMAX,DERGMX
  889 FORMAT(1X,2A6,I5,1P8E11.3)
C
 1002 CONTINUE
      IF(MOSTPS.GE.KSTEP) GO TO 1000
      KES=KE
      JES=JE
      KBS=KB
      JBS=JB
      KAS=KA
      JAS=JA
      NSAVE=N
      MOSTPS=KSTEP
```

```
 1000 CONTINUE
C
C END OF ANGLE SUM
C
      FRACT=FLOAT(NOESC)/FLOAT(NUMBER)
C
      IF(NPRINT.GT.0.OR.(MD.EQ.0.AND.MC.GT.0))
     1                              WRITE(M,671)NOESC,NUMBER,FRACT,E,DENS
  671 FORMAT(18H0 RATIO ESCAPING =, I5, 8H OUT OF ,I5,
     1  15H OR A FRACTION ,F13.8,    13H AT ENERGY E=,F13.8,4X,
     2 6H(DENS=,1PE13.4,1H))
C
      IF(NPRINT.EQ.0) GO TO 5555
      IF(ME.NE.0.AND.MC.EQ.0) WRITE(M,675)
      IF(ME.NE.0.AND.MC.GT.0) WRITE(M,676)
  675 FORMAT(1X, 66HDENS IS THE SUM OF DADO=SPEED*EXP(XPON)/NUMBER OVER
     1ALL DIRECTIONS//)
  676 FORMAT(1X, 67HDENS IS THE SUM OF DADO=SPEED**2*EXP(XPON)/NUMBER OV-
     1ER A HEMISPHERE//)
C
 5555 IF(ME.EQ.0) GO TO 1001
      IF(MC.EQ.0) COEFF2=4./SQRT(PI)/(1. - CE)**2/FLOAT(ME)
      IF(MC.GT.0) COEFF2=2./(1. - CE)**2/FLOAT(ME)
      IF(XMACH.GT.1.) COEFF2=COEFF2*XMACH**2
      DENST=DENST + COEFF2*DENS
C
 1001 CONTINUE
C
      IF(ME.EQ.0.AND.MC.EQ.0) DENST=SPEED*FRACT
      IF(ME.EQ.0.AND.MC.GT.0) DENST=SPEED**2*FRACT
   96 IF(MC.GT.0) WRITE(M,677) RSAVE,ZSAVE,PHISAV,PARTCL,DENST
  677 FORMAT(/6H AT R=,F13.8, 7H AND Z=,F13.8, 13H, THE POTENTIAL IS=,
     1F13.8/1X,20H AND THE NORMALIZED ,2A6,20H CURRENT DENSITY IS=,1PE13
     2.4//)
C
      IF(SCALE.LT.1..AND.MC.EQ.0) GO TO 91
      IF(SCALE.LT.1..AND.MC.GT.0) GO TO 90
      SCALE=-1.
      PARTCL(1)=PART2(1)
      PARTCL(2)=PART2(2)
      DENSA=DENST
      IF(MC.EQ.0.AND.ISAVE.EQ.0) OSAVE(N)=DENSA
      IF(N.LE.N2.AND.IN.GT.0.AND.JN.GT.0) ONORTH(IN,JN)=DENSA
      IF(N.GT.N1.AND.IS.GT.0.AND.JS.GT.0) OSOUTH(IS,JS)=DENSA
      IF(MD.GT.0.AND.MC.EQ.0.AND.N.EQ.NPTS) ONORTH(IIN,1)=DENSA
      GO TO 237
C RETURN TO BEGINNING OF TRAJECTORIES FOR ELECTRONS
C
C CONTINUE IF IONS AND ELECTRONS COMPLETED
   91 CD=DENSA-DENST
      CDSV(N)=CD
      IF(MC.EQ.0.AND.NEWPHI.EQ.0) OSAVE(N)=CD
C SAVE ION DENSITY ONLY IF EXP(PHI) IS TO BE INCLUDED IN POISSON SOLUTION
      IF(MC.EQ.0.AND.NEWPHI.GT.0) CDSV(N)=DENSA
      PHISAV=-PHISAV
      IF(MC.EQ.0) WRITE(M,672) N,RSAVE,ZSAVE,PHISAV,DENSA,DENST,CD
  672 FORMAT(1X, 5HAT N=,I4,9H R,Z,PHI=,1P2E10.2,E12.4,
```

```
     1 43H, THE ION/ELECTRON/CHARGE DENSITIES ARE=, 3E13.4)
       IF(INCREA.GT.0) WRITE(M,6721) N,MA,MB,ME,STEP
6721   FORMAT(1X,12HFOR POINT N=, I3,4X,
     1 33HMA,MB,ME, AND STEP ARE CHANGED TO,3I5,4H AND,F13.5,
     2 23H FOR INCREASED ACCURACY/)
C
   90  DO 92 I=1,IIN
       DO 92 J=1,JJN
   92  PHIN(I,J)=SCALE*PHIN(I,J)
       DO 93 I=1,IIS
       DO 93 J=1,JJS
   93  PHIS(I,J)=SCALE*(PHIS(I,J))
***** RESTORE MA,MB,ME,STEP
       MA=MASAVE
       MB=MBSAVE
       ME=MESAVE
       STEP=STEPSV
   95  CONTINUE
       IF(MD.GT.0) WRITE(M,666) IT,NGR
       IF(MD.GT.0) WRITE(M,94) (N,RSV(N),ZSV(N),COSV(N), N=1,NPTS)
94     FORMAT(/4X,1HN,8X,1HR,12X,1HZ,12X,4HD/CD/(1X,I4,3F13.5))
C TRAJECTORY WITH MOST STEPS. PRINT INDICES(N,AND K AND J INDICES)
                        WRITE(M,97) MOSTPS,NSAVE,KES,JES,KBS,JBS,KAS,JAS
   97  FORMAT(1X,I5,I4, 3(I3,I2),34H =MOSTSTEPS,N, KE,JE, KB,JB, KA,JA )
       IF(IPRINT.EQ.0) GO TO 99
       WRITE(M,2001)
       IF(ISAVE.GT.0) WRITE(M,2003)
2001   FORMAT(1H1/1X,21HDENSITY ARRAY - NORTH  )
       WRITE(M,2004) (RN(J),J=1,JJN)
2004   FORMAT(/1X,2HR=,16F8.4/(/3X,16F8.4))
2003   FORMAT(1X,40HDENSITIES READ IN RATHER THAN CALCULATED //)
       DO 100 I=1,IIN
       WRITE(M,333) I,ZN(I),(DNORTH(I,J),J=1,JJN)
100    CONTINUE
       WRITE(M,2002)
2002   FORMAT(////1X,21HDENSITY ARRAY - SOUTH//)
       WRITE(M,2004) (RS(J),J=1,JJS)
       DO 101 I=1,IIS
       WRITE(M,333) I,ZS(I),(DSOUTH(I,J),J=1,JJS)
101    CONTINUE
99     CONTINUE
       END
```

```
      SUBROUTINE INTERP
      COMMON JJN,IIN,JJS,IIS,ND,RN(50),ZN(50),RS(50),ZS(50),RSV( 500),
     1 ZSV( 500),PHIN(20,20),PHIS(20,20),COSV( 500),IFIRST,M
      COMMON/P/INT,IGN,JGN,IGS,JGS
      COMMON/T/X,Y,Z,R,XDOT,YDOT,ZDOT,PHI,PHIR,PHIZ,ERSO,DERG,C1,C2,E
      IGN=JGN=IGS=JGS=1
      NCH=0
      IGO2 = 0
      IF(INT.NE.0) IGO2 = IGO
      IGO=2
      IF (Z.GE.0.) IGO=1
      IF(IGO.NE.IGO2) INT = 0
      GO TO (1,2),IGO
C
C     NORTH Z
C
C     ASSUMING ZN(IIN) LESS THAN OR EQUAL TO Z LESS THAN OR EQUAL TO ZERO
C
    1 IF(Z.EQ.ZN(1)) IG=2
      IF(Z.EQ.ZN(1)) GO TO 103
      IF(INT.NE.0) GO TO 100
      DO 10 I=2,IIN
      IG=IIN-I+2
      IF(Z.LT.ZN(IG-1)) GO TO 103
   10 CONTINUE
C
C     ACCEPT IF ZN(IG) LESS THAN OR EQUAL TO Z LESS THAN ZN(IG-1).
C
  100 IF(Z.GE.ZN(IG-1)) GO TO 102
      IF(Z.GE.ZN(IG)) GO TO 104
  101 IG=IG+1
      IF(Z.LT.ZN(IG)) GO TO 101
      GO TO 103
  102 IG=IG-1
      IF(Z.GE.ZN(IG-1)) GO TO 102
  103 NCH=1
  104 CONTINUE
C
C     NORTH R
C
C     ASSUMING RN(1) LESS THAN OR EQUAL TO R LESS THAN OR EQUAL TO RN(JJ)
C
      IF(R.EQ.RN(JJN))JG=JJN-1
      IF(R.EQ.RN(JJN))GO TO 153
      IF(INT.NE.0) GO TO 150
      DO 15 J=2,JJN
      JG=J-1
      IF(R.LT.RN(J)) GO TO 153
   15 CONTINUE
C
C     ACCEPT IF RN(JG) LESS THAN OR EQUAL TO R LESS THAN RN(JG+1).
C
  150 IF(R.GE.RN(JG+1)) GO TO 152
      IF(R.GE.RN(JG)) GO TO 154
  151 JG=JG-1
      IF(R.LT.RN(JG))GO TO 151
      GO TO 153
```

```
  152 JG=JG+1
      IF(R.GE.RN(JG+1))GO TO 152
  153 NCH=1
  154 CONTINUE
C
C     SET UP FRACTIONS
C
      DELZ=ZN(IG-1)-ZN(IG)
      DELR=RN(JG+1)-RN(JG)
      FZ=(Z-ZN(IG))/DELZ
      FR=(R-RN(JG))/DELR
        P22=PHIN(IG-1,JG+1)
        P21=PHIN(IG-1,JG  )
        P12=PHIN(IG  ,JG+1)
        P11=PHIN(IG  ,JG  )
      IGN=IG
      JGN=JG
      GO TO 500
C
C     SOUTH Z
C
C     ASSUMING ZS(IIS) LESS THAN OR EQUAL TO Z LESS THAN ZERO.
C
    2 IF(INT.NE.0) GO TO 200
      DO 20 I=2,IIS
      IG=IIS-I+2
      IF(Z.LT.ZS(IG-1)) GO TO 203
   20 CONTINUE
C
C     ACCEPT IF ZS(IG) LESS THAN OR EQUAL TO Z LESS THAN ZS(IG-1).
C
      IF(Z.GE.ZS(IG)) GO TO 204
  201 IG=IG+1
  200 IF(Z.GE.ZS(IG-1))GO TO 202
      IF(Z.LT.ZS(IG))GO TO 201
      GO TO 203
  202 IG=IG-1
      IF(Z.GE.ZS(IG-1))GO TO 202
  203 NCH=1
  204 CONTINUE
C
C     SOUTH R
C
C     ASSUMING RS(1) LESS THAN OR EQUAL TO R LESS THAN OR EQUAL TO RS(JJ
C
      IF(R.EQ.RS(JJS)) JG=JJS-1
      IF(R.EQ.RS(JJS)) GO TO 253
      IF(INT.NE.0) GO TO 250
      DO 25 J=2,JJS
      JG=J-1
      IF(R.LT.RS(J)) GO TO 253
   25 CONTINUE
C
C     ACCEPT IF RS(JG) LESS THAN OR EQUAL TO R LESS THAN RS(JG+1).
C
  250 IF(R.GE.RS(JG+1)) GO TO 252
      IF(R.GE.RS(JG))GO TO 254
```

```
  251 JG=JG-1
      IF(R.LT.RS(JG))GO TO 251
      GO TO 253
  252 JG=JG+1
      IF(R.GE.RS(JG+1))GO TO 252
  253 NCH=1
  254 CONTINUE
C
C     SET UP FRACTIONS
C
      DELZ=ZS(IG-1)-ZS(IG)
      DELR=RS(JG+1)-RS(JG)
      FZ=(Z-ZS(IG))/DELZ
      FR=(R-RS(JG))/DELR
      P22=PHIS(IG-1,JG+1)
      P21=PHIS(IG-1,JG)
      P12=PHIS(IG,JG+1)
      P11=PHIS(IG,JG)
      IGS=IG
      JGS=JG
C
C     INTERPOLATE
C
  500 IF(NCH.EQ.0) GO TO 501
C
C     SKIP IF NO CHANGE IN PHI-BOX
C
      D1=(P22-P12) /DELZ
      D2=(P21-P11) /DELZ
      D3=(P22-P21) /DELR
      D4=(P12-P11) /DELR
  501 PHIZ=D2 + FR*(D1-D2)
      PHIR=D4 + FZ*(D3-D4)
      PHI=P11 + FR*(P12-P11) + FZ*(P21-P11) + FR*FZ*(P22-P21-P12+P11)
      RETURN
      END
```

```
      SUBROUTINE TRACK
      COMMON/T/X,Y,Z,R,XDOT,YDOT,ZDOT,PHI,PHIR,PHIZ,ERGO,DERG,C1,C2,E
      DERG=(PHI + XDOT**2 + YDOT**2 + ZDOT**2)/E-1. - ERGO
      ERGO=ERGO + DERG
      VMAX=ABS(XDOT) + ABS(YDOT) + ABS(ZDOT)
C
C STEP CONTROL
C
      IF(R.EQ.0.)  PHIX=0.
      IF(R.EQ.0.)  PHIY=0.
      IF(R.EQ.0.) GO TO 1
      PHIX=PHIR*X/R
      PHIY=PHIR*Y/R
    1 SS=AMIN1(C2, C1/VMAX)
      DT=SS/(ABS(PHIX) + ABS(PHIY) + ABS(PHIZ) + 1.E-6)
      DT=AMAX1(DT, .01/VMAX)
C THE FOLLOWING CARD IS FOR ZERO-POTENTIAL TESTS
      IF(PHIR.EQ.0..AND.PHIZ.EQ.0.) DT=C1/E/VMAX
      X=X+DT*(XDOT-.25*DT*PHIX)
      Y=Y+DT*(YDOT-.25*DT*PHIY)
      Z=Z+DT*(ZDOT-.25*DT*PHIZ)
      XDOT=XDOT-.5*DT*PHIX
      YDOT=YDOT-.5*DT*PHIY
      ZDOT=ZDOT-.5*DT*PHIZ
      RETURN
      END
```

REFERENCES

Call, S. M., The interaction of a satellite with the ionosphere, Rep. 46, Columbia Univ., New York, 1969.

Davis, A. H., and I. Harris, Interaction of a charged satellite with the ionosphere, in Rarefied Gas Dynamics, edited by L. Talbot, pp. 691-699, Academic, New York, 1961.

Fournier, G., Collisionless plasma flow around a cylinder considering applications to ionospheric probes (in French), Publication No. 137, Office Nat. d' Etudes et de Recherches Aérospatiales, Paris, 1971.

Fournier, G., and D. Pigache, Wakes in collisionless plasma, Phys. Fluids, 18, 1443-53, 1975.

Grabowski, R., and T. Fischer, Theoretical density distribution of plasma streaming around a cylinder, Planet. Space Sci., 23, 287-304, 1975.

Gurevich, A. V., L. P. Pitaevskii, and V. V. Smirnova, Ionospheric aerodynamics, Space Sci. Rev., 9, 805-871, 1969.

Henderson, C. L., and U. Samir, Observations of the disturbed region around an ionospheric spacecraft, Planet. Space Sci., 15, 1499-1513, 1967.

Jew, H., Numerical studies of rarefied-plasma interactions at mesothermal speeds, Ph.D. Thesis, Univ. Michigan, 1968.

Kiel, R. E., F. C. Gey, and W. A. Gustafson, Electrostatic potential fields of an ionospheric satellite, AIAA J., 6, 690-694, 1968.

Liu, V-C., Ionospheric gas dynamics of satellite and diagnostic probes, Space Sci. Rev., 9, 483-490, 1969.

Martin, A. R., Numerical solutions to the problem of charged particle flow around an ionospheric spacecraft, Planet. Space Sci., 22, 121-141, 1974.

Maslennikov, M. V., and Yu. S. Sigov, A discrete model for the study of the flow of a rarefied plasma about a body, Soviet Phys. - Doklady, 9, 1063-1065, 1965.

McDonald, P. W., and F. O. Smetana, Results of a numerical experiment to determine the current collected by a charged cylinder in a collisionless plasma stream, in Rarefied Gas Dynamics, edited by L. Trilling and H. Y. Wachman, pp. 1627-1636, Academic, New York, 1969.

Oran, W. A., N. H. Stone, and U. Samir, The effects of body geometry on the structure in the near wake zone of bodies in a flowing plasma, J. Geophys. Res., 80, 207-9, 1975.

Parker, L. W., Numerical methods for computing the density of a rarefied gas about a moving abject, Rep. AFCRL-64-193 (for Air Force Cambridge Research Laboratories), Allied Research Associates, Inc., Concord, MA, 1964.

Parker, L.W., Theory of the circular planar (guardring) Langmuir probe, Rep. ERL 100-AL2 (for Environmental Science Services Administration), Mt. Auburn Research Associates, Inc., Cambridge, MA, 1968.

Parker, L. W., Theory of the external sheath structure and ion collection characteristics of a rocket-borne mass spectrometer, Rep. AFCRL-71-0105 (for Air Force Cambridge Research Laboratories), Mt. Auburn Research Associates, Inc., Cambridge, MA, 1970.

Parker, L. W., Computer solutions in electrostatic probe theory, Rep. AFAL-TR-72-222, Part II (for Air Force Avionics Laboratory), Mt. Auburn Research Associates, Inc., Newton, MA, 1973.

Parker, L. W., Computer method for satellite plasma sheath in steady-state spherical symmetry, Rep. AFCRL-TR-75-0410 (for Air Force Cambridge Research Laboratories), Lee W. Parker, Inc., Concord, MA, 1975.

Parker, L. W., and E. C. Sullivan, Iterative methods for plasma-sheath calculations - Application to spherical probe, Rep. NASA TN D-7409, 1974.

Parker, L. W., and E. C. Whipple, Jr., Theory of a satellite electrostatic probe, Ann. Phys. (USA), 44, 126-161, 1967.

Parker, L. W. and E. C. Whipple, Jr., Theory of spacecraft sheath structure, potential, and velocity effects or ion measurements by traps and mass spectrometers, J. Geophys. Res., 75, 4720-33, 1970.

Samir, U., E. J. Maier, and B. E. Troy, Jr., The angular distribution of ion flux around an ionospheric satellite, J. Atmos. Terr. Phys., 35, 513-519, 1973.

Taylor, J. C., Disturbance of a rarefied plasma by a supersonic body on the basis of the Poisson-Vlasov equations - I, Planet. Space Sci., 15, 155-187, 1967.

☆ U.S GOVERNMENT PRINTING OFFICE 1976-641-255 284 REGION NO 4